# Image Analysis by Moments

by
Simon Xinmeng Liao

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

The Department of
Electrical and Computer Engineering
The University of Manitoba
Winnipeg, Manitoba, Canada

**献给“六·四”殉难未能完成学业的学生们。**

To the students who were massacred in Beijing, June 4th, 1989, and could not finish their education.

# Abstract

To select a set of appropriate numerical attributes of features from the interested objects for the purpose of classification has been among the fundamental problems in the design of an imagery pattern recognition system. One of the solutions, the utilization of moments for object characterization has received considerable attentions in recent years. In this research, the new techniques derived to increase the accuracy and the efficiency in moment computing are addressed. Based on these developments, the significant improvement on image reconstructions via **Legendre** moments and **Zernike** moments has been achieved. The effect of image noise on image reconstruction, the automatic selection of the optimal order of moments for image reconstruction from noisy image, and the usage of moments as image features for character recognition are analyzed as well.

# Acknowledgements

Many people have provided advice, support, and encouragement to the author, during the research which led to this thesis. I would like to express my heartfelt appreciation to:

My supervisor, Prof. Dr. Miroslaw Pawlak, for his generous support and intellectual guidance throughout my years as a graduate student; his insightful advice, clear vision, many suggestions, and endless efforts to be available for many educational discussions, were invaluable;

Prof. Dr. David Erbach, whose friendship and encouragement were invaluable and kept me thinking that there really was a light at the end of the tunnel, and who provided many valuable comments on drafts of this thesis;

my committee members, Prof. Dr. Richard Gordon and Prof. Dr. Waldemar Lehn, for valuable insights and suggestions which have significantly improved this thesis in both structure and contents;

my External Examiner, Prof. Dr. Adam Krzyzak, for his critical comments and constructive suggestions on this thesis;

my wife, Dr. Ming Yang, who shared with the pains and happiness during the course of this work; her endless support, sacrifice, and understanding kept me going through it all;

and finally, my parents, Li Bofan and Liao Cuichuan, who first taught me the importance of education.

# Contents

# List of Figures

# List of Tables

# List of Symbols

Some of the most frequently occurring abbreviations and symbols used in the text are tabulated here. Other symbols are explained where used.

| | |
|---|---|
| $A_{nm}$ | = **Zernike** moments of order n with repetition m |
| $\widehat{A}_{nm}$ | = Digital version of $A_{nm}$ |
| $C_n f$ | = cubature formula |
| $C_{pq}$ | = complex moments |
| $E_A$ | $= \sum \sum \lvert\widehat{A}_{nm}\rvert^2 \qquad m = n \neq 0$ |
| $E_\lambda$ | $= \sum_{m=0}^{M_{max}} \sum_{n=0}^{m} \widehat{\lambda}_{mn}^2 \qquad m = n \neq 0$ |
| $\widehat{f}(x,y)$ | = reconstructed image from $f(x,y)$ |
| $F(u,v)$ | = characteristic function of the image function $f(x,y)$ |
| $g(x,y)$ | = noisy degraded version of $f(x,y)$ |
| $M_{pq}$ | = geometric moments of order (p+q) |
| $N$ | = the number of points which are spaced apart by a constant step $h$ inside a single interval |
| $P_n(x)$ | = **Legendre** polynomials |
| $R_{nm}$ | = Radial polynomials |
| $V_{nm}$ | = **Zernike** polynomials |
| $\delta_{mn}$ | = **Kronecker** symbol |
| $\lambda_{mn}$ | = **Legendre** moments |
| $\widehat{\lambda}_{mn}$ | = Digital version of $\lambda_{mn}$ |

|   |   |
|---|---|
| $\widetilde{\lambda}_{mn}$ | = **Legendre** moments from $g(x, y)$ |
| $\mu_{pq}$ | = central moments |

# Chapter 1

# Introduction

One of the basic problems in the design of an imagery pattern recognition system relates to the selection of a set of appropriate numerical attributes of features to be extracted from the object of interest for the purpose of classification. The recognition of objects from imagery may be achieved with many methods by identifying an unknown object as a member of a set of known objects. Efficient object recognition techniques abstracting characterizations uniquely from objects for representation and comparison are crucially important for a given pattern recognition system.

Research on the utilization of moments for object characterization in both invariant and noninvariant tasks has received considerable attention in recent years. The principal techniques explored include *Moment Invariants*, *Geometric Moments*, *Rotational Moments*, *Orthogonal Moments*, and *Complex Moments*. Various forms of moment descriptors have been extensively employed as pattern features in scene recognition, registration, object matching as well as data compression.

The mathematical concept of moments has been around for many years and has been used in many diverse fields ranging from mechanics and statistics to pattern recognition and image understanding. Describing images with moments instead of other more commonly used image features means that global properties of the image are used rather than local properties.

Historically, Hu[36][37] published the first significant paper on the utilization of moment invariants for image analysis and object representation in 1961. Hu's approach was based on the work of the nineteenth century mathematicians Boole, Cayley, and Sylvester, on the theory of algebraic forms. Hu's *Uniqueness Theorem* states that if $f(x, y)$ is piecewise continuous and has nonzero values only in the finite part of the $(x, y)$ plane, then geometric moments of all orders exist. It can then be shown that the moment set $\{m_{pq}\}$ is uniquely determined by $f(x, y)$ and conversely, $f(x, y)$ is uniquely determined by $\{m_{pq}\}$. Since an image segment has finite area and, in the worst case, is piecewise continuous, a moment set can be computed and used to uniquely describe the information contained in the image segment. Using nonlinear combinations of geometric moments, Hu derived a set of invariant moments which has the desirable properties of being invariant under image translation, scaling, and rotation. However, the reconstruction of the image from these moments is deemed to be quite difficult.

The *Rotational* moment is an alternative to the regular geometric moment. The *Rotational* moments are based on a polar coordinate representation of the image and can be used to extend the definition of moment invariants to arbitrary order in a manner which ensures that their magnitudes do not diminish significantly with increasing order. Smith and Wright[69] used a simplified *Rotational* moment technique to derive invariant features from noisy low resolution images of ships. Boyce and Hossack[15] derived the *Rotational* moments of arbitrary order that are invariant to rotation, radial scaling, and intensity change.

In 1980, Teague[73] presented two inverse moment transform techniques to determine how well an image could be reconstructed from a set of moments. The first method, called *moment matching*, derives a continuous function

$$g(x,y) = g_{00} + g_{10}x + g_{01}y + g_{20}x^2 + g_{11}xy + g_{02}y^2 +$$

$$g_{30}x^3 + g_{21}x^2y + g_{12}xy^2 + g_{03}y^3 + \ldots,$$

whose moments exactly match the geometric moments $\{m_{pq}\}$ of $f(x, y)$ through order $n$. However, this technique is impractical for calculation as it requires one to solve an increasing number of coupled equations when higher order moments are considered. Then, Teague suggested the notion of orthogonal moments to recover the image from moments based on the theory of orthogonal polynomials. Teague introduced the rotationally invariant **Zernike** moment, which employs the complex **Zernike** polynomials as the moment basis set, and the **Legendre** moment, using **Legendre** polynomials as its basis set. Significant efforts have been made in various experimental image reconstruction tasks performed by Teague, then Boyce and Hossack[15], Teh and Chin[75], Taylor and Reeves[72], and more recently, Khotanzad and Hong[45][46] with both **Zernike** and **Legendre** methods. However, no high quality multi-graylevel image has ever been successfully reconstructed from its original version.

Later, the notion of *Complex* moments was introduced by Abu-Mostafa and Psaltis[1] as a simple and straightforward way to derive a set of invariant moments. Abu-Mostafa and Psaltis used *Complex* moments to investigate the informational properties of moment invariants. However, comparing *Complex* moments with the **Zernike** moments, they concluded that the *Complex* moment invariants are not good image features. In other work, Abu-Mostafa and Psaltis[2] examined the utilization of moments in a generalized image normalization scheme for invariant pattern recognition. They redefined the classic image normalizations of size, position, rotation, and contrast, in terms of *Complex* moments. Moment invariants were shown to be derivable from *Complex* moments of the normalized image as well.

Teh and Chin[75] performed an extensive analysis and comparison of the most common moment definitions. They examined the noise sensitivity and information redundancy of **Legendre** moments along with five other types of moments. Teh and Chin concluded that higher order moments are more sensitive to noise. Among the explored techniques, *Complex* moments are least sensitive to noise while **Legendre** moments are most severely affected by noise. In terms of information redundancy, **Legendre**, **Zernike**, and pseudo-**Zernike** moments are uncorrelated and have the least redundancy. In terms of overall performance, **Zernike** and pseudo-**Zernike** moments are the best. In general, orthogonal moments are better than other types of moments in terms of information redundancy and image representation.

More recently, Prokop and Reeves[62] reviewed the basic geometric moment theory and its application to object recognition and image analysis. The geometric properties of low-order moments were discussed along with the definition of several moment-space linear geometric transforms. Prokop and Reeves also presented an extensive review summarizing most of research advancements related to the moment-based object representation and recognition techniques over the past 30 years.

The speed of computing image moments is extraordinarily important when higher order moments are involved. Several schemes of hardware architectures have been performed to speed up the computation of image moments. Reeves[64] proposed a parallel, mesh-connected SIMD computer architecture for rapidly manipulating moment sets. The architecture offered a reasonable speeding up over a single processor for high speed image analysis operations and was expected to be implemented in VLSI technology. Andersson[6] developed a VLSI moment-generating chip and presented a real-time system by implementing the processor. Hatamian[33] proposed an algorithm and single chip VLSI implementation to generate raw moments. It is claimed that 16 geometric moments, $m_{pq}(p = 0, 1, 2, 3, q = 0, 1, 2, 3)$, of a $512 \times 512 \times 8$ bit image can be computed at 30 frames/sec. The moment algorithm is based on using the one-dimensional discrete moment-generating function as a digital filter.

The organization of this thesis is as follows. Chapter 2 will review the general characteristics of various types of moments and their properties. In Chapter 3, the new techniques derived to increase the accuracy and the efficiency in moment computing, will be addressed. Chapter 4 will discuss the reconstruction algorithms of the **Legendre** moments and the **Zernike** moments, and provide significantly improved reconstructed images from these orthogonal moments. Then, the effect of image noise on image reconstruction and the automatic selection of the optimal order of moments for image reconstruction will be analyzed in Chapter 5. Several specific recognition aspects of proposed moment techniques for character recognition are studied in Chapter 6. Finally, Chapter 7 will summarize the important results and conclusions of the entire study.

# Chapter 2

# Theory of Moments

## 2.1 Introduction

Numerous problems in mechanics, physics, and engineering lead to the problem of characterization of a function in terms of some functionals. In particular, moment functionals have attracted great attention[78] due to their mathematical simplicity and numerous physical interpretations.

A complete characterization of moment functionals over a class of univariate functions was given by Hausdorff[26] in 1921.

Let $\{\mu_n\}$ be a real sequence of numbers and let us define

$$\Delta^m \mu_n = \sum_{i=0}^{m} (-1)^i \left( \begin{array}{c} m \\ i \end{array} \right) \mu_{n+i}. \tag{2.1}$$

Note that $\Delta^m \mu_n$ can be viewed as the $m$th order derivative of $\mu_n$.

By Hausdorff's theorem, a necessary and sufficient condition that there exists a monotonic function $F(x)$ satisfying the system

$$\mu_n = \int_0^1 x^n dF(x), \qquad n = 0, 1, 2, ... \tag{2.2}$$

is that the system of linear inequalities

$$\Delta^k \mu_n \geq 0 \qquad k = 0, 1, 2, ... \tag{2.3}$$

should be satisfied. I.e., if $f(x)$ is a positive function (which is the case in image processing), then the set of functionals

$$\int_0^1 x^n f(x)dx, \qquad n = 0, 1, ...$$

completely characterizes the function.

A necessary and sufficient condition that there exists a function $F(x)$ of bounded variation satisfying (2.2) is that the sequence

$$\sum_{m=0}^{p} \begin{pmatrix} p \\ m \end{pmatrix} |\Delta^{p-m}\mu_m| \qquad p = 0, 1, 2, ...$$

should be bounded.

These results were extended to the two-dimensional case by Hildebrandt and Schoenberg[34] in 1933.

Since then, moments and functions of moments have been utilized in a number of applications to achieve both invariant and noninvariant recognitions of two-dimensional and three-dimensional image patterns[62].

In this chapter, the various types of moments are defined and their properties are summarized. It is assumed that an image can be represented by a real valued measurable function $f(x, y)$.

## 2.2 Geometric Moments in Image Processing

### 2.2.1 Preliminaries

The two-dimensional geometric moment of order $(p + q)$ of a function $f(x, y)$ is defined as

$$M_{pq} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} x^p \, y^q \, f(x, y) \, dxdy, \tag{2.4}$$

where $p, q = 0, 1, 2, ..., \infty$. Note that the monomial product $x^p y^q$ is the basis function for this moment definition.

A set of $n$ moments consists of all $M_{pq}$'s for $p + q \leq n$, i.e., the set contains $\frac{1}{2}(n+1)(n+2)$ elements.

The use of moments for image analysis and pattern recognition was inspired by Hu[37] and Alt[5]. Hu stated that if $f(x, y)$ is piecewise continuous and has nonzero values only in a finite region of the $(x, y)$ plane, then the moment sequence $\{M_{pq}\}$ is uniquely determined by $f(x, y)$, and conversely, $f(x, y)$ is uniquely determined by $\{M_{pq}\}$. Considering the fact that an image segment has finite area, or in the worst case is piecewise continuous, moments of all orders exist and a complete moment set can be computed and used uniquely to describe the information contained in the image. However, to obtain all of the information contained in an image requires an infinite number of moment values. Therefore, to select a meaningful subset of the moment values that contain sufficient information to characterize the image uniquely for a specific application becomes very important.

## 2.2.2 Properties of Geometric Moments

The lower order moments represent some well known fundamental geometric properties of the underlying image functions.

### Central Moments

The central moments of $f(x, y)$ are defined as

$$\mu_{pq} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} (x - \bar{x})^p \, (y - \bar{y})^q \, f(x, y) \, dx dy, \tag{2.5}$$

where $\bar{x}$ and $\bar{y}$ are defined in (2.10).

The central moments $\mu_{pq}$ defined in Eq. (2.5) are invariant under the translation of coordinates[37]:

$$\begin{aligned} x' &= x + \alpha, \\ y' &= y + \beta, \end{aligned} \tag{2.6}$$

where $\alpha$ and $\beta$ are constants.

### Mass and Area

The definition of the zeroth order moment, $\{M_{00}\}$, of the function $f(x, y)$

$$M_{00} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} f(x, y)\, dxdy \tag{2.7}$$

represents the total mass of the given function or image $f(x, y)$. When computed for a binary image, the zeroth moment (2.7) represents the total area of the image.

### Centre of Mass

The two first order moments,

$$M_{10} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} x\, f(x, y)\, dxdy \tag{2.8}$$

and

$$M_{01} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} y\, f(x, y)\, dxdy \tag{2.9}$$

represent the centre of mass of the image $f(x, y)$. The centre of mass is the point where all the mass of the image could be concentrated without changing the first moment of the image about any axis. In the two-dimensional case, in terms of moment values, the coordinates of the centre of mass are

$$\bar{x} = \frac{M_{10}}{M_{00}}, \qquad \bar{y} = \frac{M_{01}}{M_{00}}. \tag{2.10}$$

As a usual practice, the centre of mass is chosen to represent the position of an image in the field of view. The equations in (2.10) define a unique location of the image $f(x, y)$ that can be used as a reference point to describe the position of the image.

**Orientations**

The second order moments, $\{M_{02}, M_{11}, M_{20}\}$, known as the moments of intertia, may be used to determine an important image feature, orientation. In general, the orientation of an image describes how the image lies in the field of view, or the directions of the principal axes.

In terms of moments, the orientations of the principal axes, $\theta$, are given by[35]

$$\theta = \frac{1}{2} tan^{-1}(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}). \tag{2.11}$$

In (2.11), $\theta$ is the angle of the principal axis nearest to the $x$ axis and is in the range $-\pi/4 \leq \theta \leq \pi/4$.

**Projections**

An alternative means of describing image properties represented by moments is to consider the relationship between the moments of an image and those of the projections of that image. The moments in the sets $\{M_{p0}\}$ and $\{M_{0p}\}$ are equivalent to the moments of the image projection onto the $x$ axis and $y$ axis respectively.

Consider the horizontal projection, $h(y)$, of an image $f(x, y)$ onto the $y$ axis given by

$$h(y) = \int_{a_1}^{a_2} f(x, y)\, dx. \tag{2.12}$$

Then, the one-dimensional moments, $M_q$, of $h(y)$ are obtained by

$$M_q = \int_{b_1}^{b_2} y^q\, h(y)\, dy. \tag{2.13}$$

Substituting (2.12) into (2.13) gives

$$M_q = \int_{a_1}^{a_2} \int_{b_1}^{b_2} y^q\, f(x, y)\, dxdy = M_{0q}. \tag{2.14}$$

Figure 2.1 illustrates the projections of an object onto the $x$ axis and $y$ axis and the moment subsets corresponding to the projections.

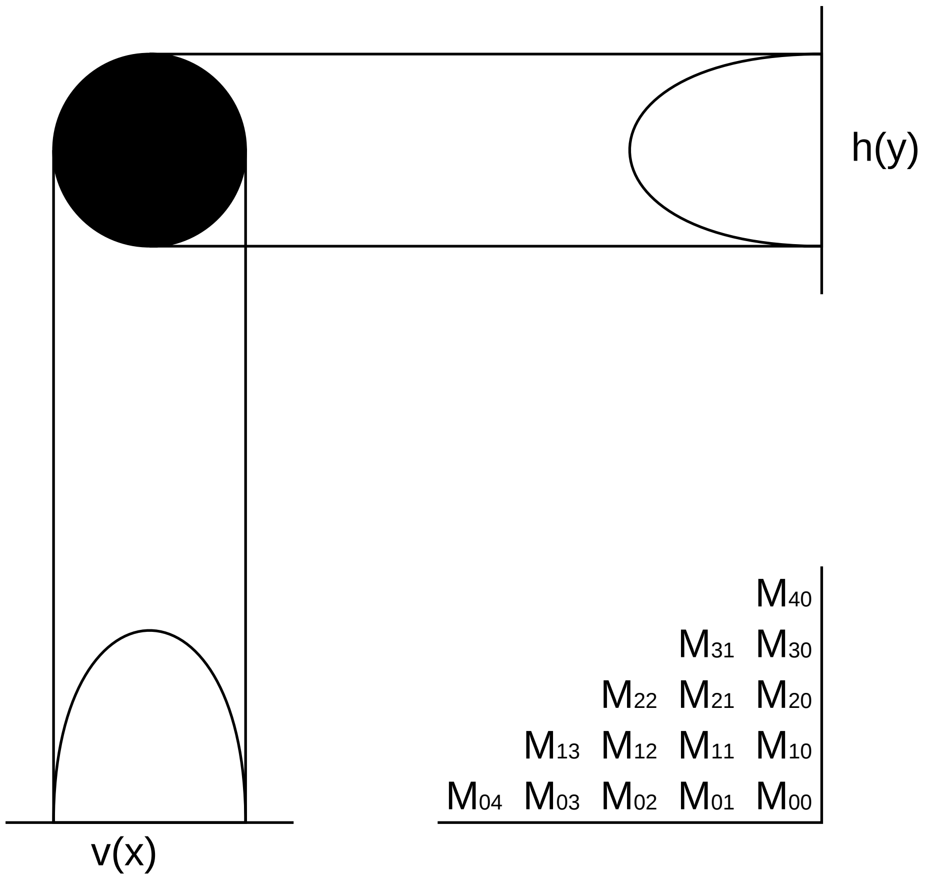


Figure 2.1: Moments projections onto $x$ and $y$ axes.

### 2.2.3 Moment Invariants

The earliest significat work employing moments for image processing and pattern recognition was performed by Hu[37] and Alt[5]. Based on the theory of algebraic invariants, Hu[36][37] derived relative and absolute combinations of moments that are invariant with respect to scale, position, and orientation.

The method of moment invariants is derived from algebraic invariants applied to the moment generating function under a rotation transformation. The set of absolute moment invariants consists of a set of nonlinear combinations of central moments that remain invariant under rotation. Hu defines the following seven functions, computed from central moments through order three, that are invariant with respect to object scale, translation and rotation:

$$\phi_1 = \mu_{20} + \mu_{02} \tag{2.15}$$

$$\phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \tag{2.16}$$

$$\phi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \tag{2.17}$$

$$\phi_4 = (\mu_{30} - \mu_{12})^2 + (\mu_{21} - \mu_{03})^2 \tag{2.18}$$

$$\begin{aligned}\phi_5 = {} & (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ & + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]\end{aligned} \tag{2.19}$$

$$\begin{aligned}\phi_6 = {} & (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \\ & + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})\end{aligned} \tag{2.20}$$

$$\begin{aligned}\phi_7 = {} & (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ & - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2].\end{aligned} \tag{2.21}$$

The functions $\phi_1$ through $\phi_6$ are invariant with respect to rotation and reflection while $\phi_7$ changes sign under reflection.

Hu's original result has been slightly modified by Reiss[66] in 1991. Reiss revised the fundamental theorem of moment invariants with four new invariants. The correction presented by Reiss affects neither similitude (scale) nor rotation invariants derived using the original theorem, but it does affect features invariant to general linear transformations.

The definition of the geometric moments (2.4) has the form of the projection of the image function $f(x, y)$ onto the monomial $x^p y^q$. However, with the Weierstrass approximation theorem[17], the basis set $\{x^p y^q\}$, while complete, is not orthogonal.

## 2.3 Complex Moments

The notion of complex moments was introduced in [1] as a simple and straightforward technique to derive a set of invariant moments. The two-dimensional complex moments of order $(p, q)$ for the image function $f(x, y)$ are defined by:

$$C_{pq} = \int_{a_1}^{a_2} \int_{b_1}^{b_2} (x + jy)^p \, (x - jy)^q \, f(x, y) \, dxdy, \tag{2.22}$$

where $p$ and $q$ are nonnegative integers and $j = \sqrt{-1}$.

The complex moments of order $(p, q)$ are a linear combination with complex coefficients of all of the geometric moments $\{M_{nm}\}$ satisfying $p + q = n + m$.

In polar coordinates, the complex moments of order $(p + q)$ can be written as

$$C_{pq} = \int_0^{2\pi} \int_0^{+\infty} \rho^{p+q} \, e^{j(p-q)\theta} \, f(\rho \, cos\theta, \rho \, sin\theta) \, \rho \, d\rho d\theta. \tag{2.23}$$

If the complex moment of the original image and that of the rotated image in the same polar coordinates are denoted by $C_{pq}$ and $C^r_{pq}$, the relationship between them is

$$C^r_{pq} = C_{pq} e^{-j(p-q)\theta}, \tag{2.24}$$

where $\theta$ is the angle that the original image has been rotated.

The complex moment invariants can be written in the form of

$$C_{rs} C^k_{tu} + C_{sr} C^k_{ut}, \tag{2.25}$$

where

$$(r-s)+k(t-u)=0. \tag{2.26}$$

This combination of complex moments cancels both the imaginary moment and the rotational phase factor, and thus provides real-valued rotation invariants.

However, these complex moment invariants are not, in general, good features[1]. They suffer from information loss, suppression, and redundancy which limit their discrimination power.

## 2.4 Orthogonal Moments

### 2.4.1 Legendre Moments

#### Legendre Polynomials

The $n$th - order **Legendre** polynomial is defined by

$$P_n(x)=\sum_{j=0}^{n} a_{nj}\, x^j = \frac{1}{2^n n!}\,\frac{d^n}{dx^n}(x^2-1)^n. \tag{2.27}$$

The **Legendre** polynomials have the generating function

$$\frac{1}{\sqrt{1-2rx+r^2}}=\sum_{s=0}^{\infty} r^s\, P_s(x), \qquad r<1. \tag{2.28}$$

From the generating function, the recurrent formula of the **Legendre** polynomials can be acquired straightforwardly:

$$\begin{aligned}
\frac{d}{dr}\left(\frac{1}{\sqrt{1-2rx+r^2}}\right) &= \frac{d}{dr}\left(\sum_{s=0}^{\infty} r^s\, P_s(x)\right) \\
\frac{x-r}{(1-2rx+r^2)^{3/2}} &= \sum_{s=0}^{\infty} s\, r^{s-1}\, P_s(x) \\
(x-r)\sum_{s=0}^{\infty} r^s\, P_s(x) &= (1-2rx+r^2)\sum_{s=0}^{\infty} s\, r^{s-1}\, P_s(x).
\end{aligned}$$

Then we have

$$xP_k(x)-P_{k-1}(x)=(k+1)P_{k+1}(x)-2xkP_k(x)+(k-1)P_{k-1}(x),$$

or, the recurrent formula of the **Legendre** polynomials:

$$P_{n+1}(x) = \frac{2n+1}{n+1}\, x\, P_n(x) - \frac{n}{n+1}\, P_{n-1}(x). \tag{2.29}$$

The **Legendre** polynomials $\{P_m(x)\}$ [17] are a complete orthogonal basis set on the interval [-1, 1]:

$$\int_{-1}^{+1} P_m(x)\, P_n(x) dx = \frac{2}{2m+1}\, \delta_{mn}, \tag{2.30}$$

where $\delta_{mn}$ is the **Kronecker** symbol.

Figure 2.2 and Figure 2.3 show some of the two-dimensional **Legendre** polynomials in the image space.

### Legendre Moments

The $(m+n)$th order of **Legendre** moment of $f(x,y)$ defined on the square $[-1,1]\times[-1,1]$ is

$$\lambda_{mn} \equiv \frac{(2m+1)(2n+1)}{4} \int_{-1}^{+1}\int_{-1}^{+1} P_m(x)\, P_n(y)\, f(x,y)\, dxdy, \tag{2.31}$$

where $m, n = 0, 1, 2, ....$

Using (2.4), (2.27), and (2.31), we have

$$\begin{aligned} \lambda_{mn} &= \frac{(2m+1)(2n+1)}{4} \int_{-1}^{+1}\int_{-1}^{1} P_m(x)\, P_n(y)\, f(x,y)\, dxdy \\ &= \frac{(2m+1)(2n+1)}{4} \int_{-1}^{1}\int_{-1}^{1} \sum_{j=0}^{m} a_{mj}\, x^j \sum_{k=0}^{n} a_{nk}\, y^k\, f(x,y)\, dxdy \\ &= \frac{(2m+1)(2n+1)}{4} \sum_{j=0}^{m}\sum_{k=0}^{n} a_{mj}\, a_{nk} \int_{-1}^{1}\int_{-1}^{1} x^j\, y^k\, f(x,y)\, dxdy. \end{aligned}$$

Therefore, the **Legendre** moments and geometric moments are related by

$$\lambda_{mn} = \frac{(2m+1)(2n+1)}{4} \sum_{j=0}^{m}\sum_{k=0}^{n} a_{mj}\, a_{nk}\, M_{jk}. \tag{2.32}$$

The above relationship indicates that a given **Legendre** moment depends only on geometric moments of the same order and lower, and conversely.

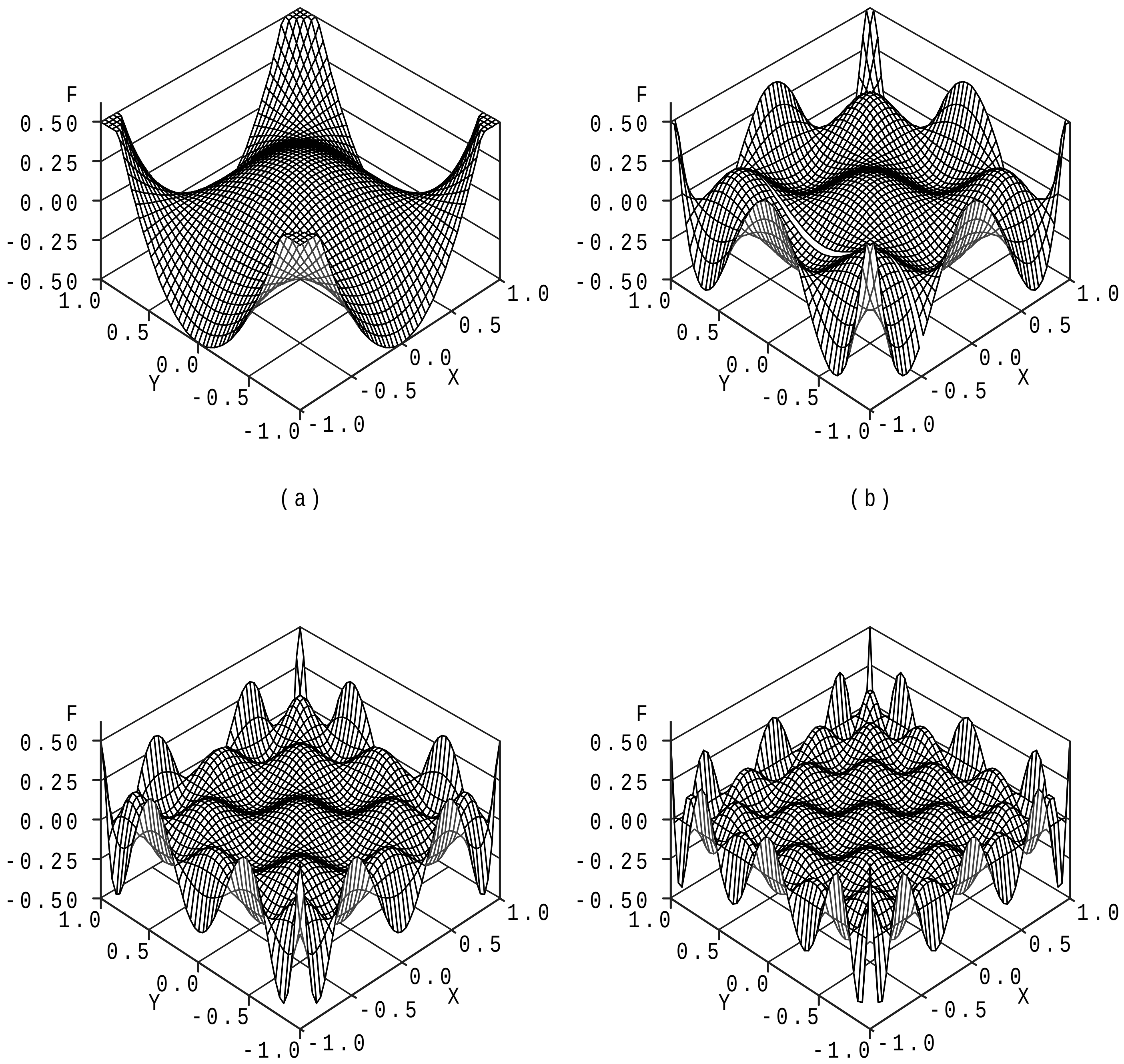


Figure 2.2: The plots of some two-dimensional $P_m(x)P_n(y)$ **Legendre** polynomials. (a) $P_2(x)P_2(y)$, (b) $P_4(x)P_4(y)$, (c) $P_6(x)P_6(y)$, and (d) $P_8(x)P_8(y)$.

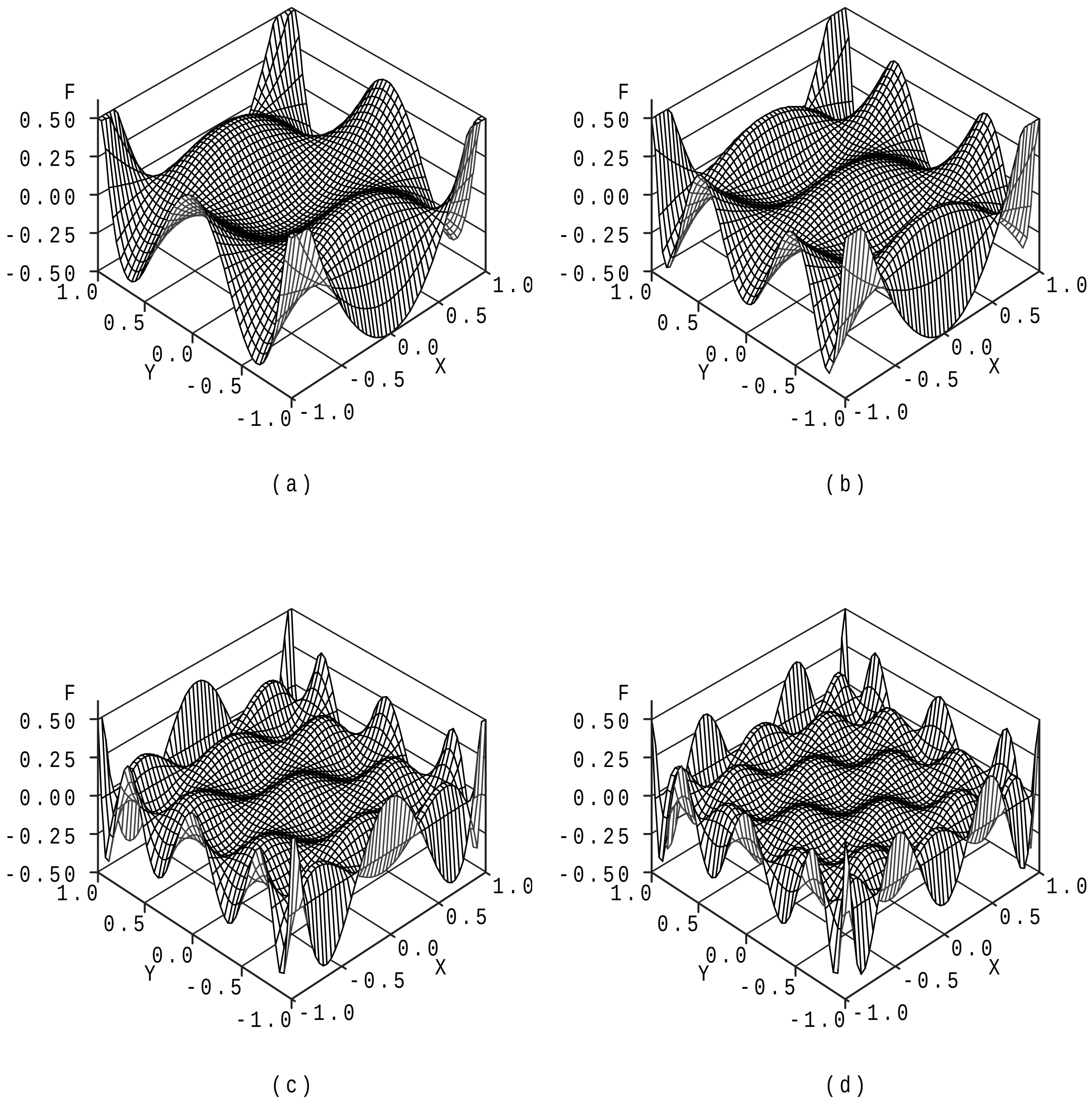


Figure 2.3: The plots of some two-dimensional $P_m(x)P_n(y)$ **Legendre** polynomials. (a) $P_2(x)P_4(y)$, (b) $P_2(x)P_6(y)$, (c) $P_4(x)P_8(y)$, and (d) $P_6(x)P_8(y)$.

### 2.4.2 Zernike Moments

The usage of **Zernike** polynomials in optics dates back to the early 20th century, while the applications of orthogonal moments based on **Zernike** polynomials for image processing were pioneered by Teague[73] in 1980.

#### Zernike Polynomials

A set of orthogonal functions with simple rotation properties which forms a complete orthogonal set over the interior of the unit circle was introduced by **Zernike**[79]. The form of these polynomials is

$$V_{nm}(x,y) = V_{nm}(\rho sin\theta, \rho cos\theta) = R_{nm}(\rho)\, exp(jm\theta) \tag{2.33}$$

where $n$ is either a positive integer or zero, and $m$ takes positive and negative integers subject to constraints $n - |m| =$ even, $|m| \leq n$, $\rho$ is the length of the vector from the origin to the pixel at $(x, y)$, and $\theta$ is the angle between vector $\rho$ and the $x$ axis in the counterclockwise direction.

The Radial polynomial $R_{nm}(\rho)$ is defined as

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s!\,(\frac{n+|m|}{2}-s)!\,(\frac{n-|m|}{2}-s)!}\,\rho^{n-2s}, \tag{2.34}$$

with $R_{n,-m}(\rho) = R_{n,m}(\rho)$.

Figure 2.4 and Figure 2.5 show some of the $V_{nm}(x, y)$ polynomials.

The **Zernike** polynomials (2.33) are a complete set of complex-valued functions orthogonal on the unit disk $x^2 + y^2 \leq 1$:

$$\iint_{x^2+y^2\leq 1} [V_{nm}(x,y)]^*\, V_{pq}(x,y)\, dxdy = \frac{\pi}{n+1}\,\delta_{np}\,\delta_{mq}, \tag{2.35}$$

or, in polar coordinates

$$\int_0^{2\pi}\int_0^1 [V_{nm}(r,\theta)]^*\, V_{pq}(r,\theta)\, r\, drd\theta = \frac{\pi}{n+1}\,\delta_{np}\,\delta_{mq}, \tag{2.36}$$

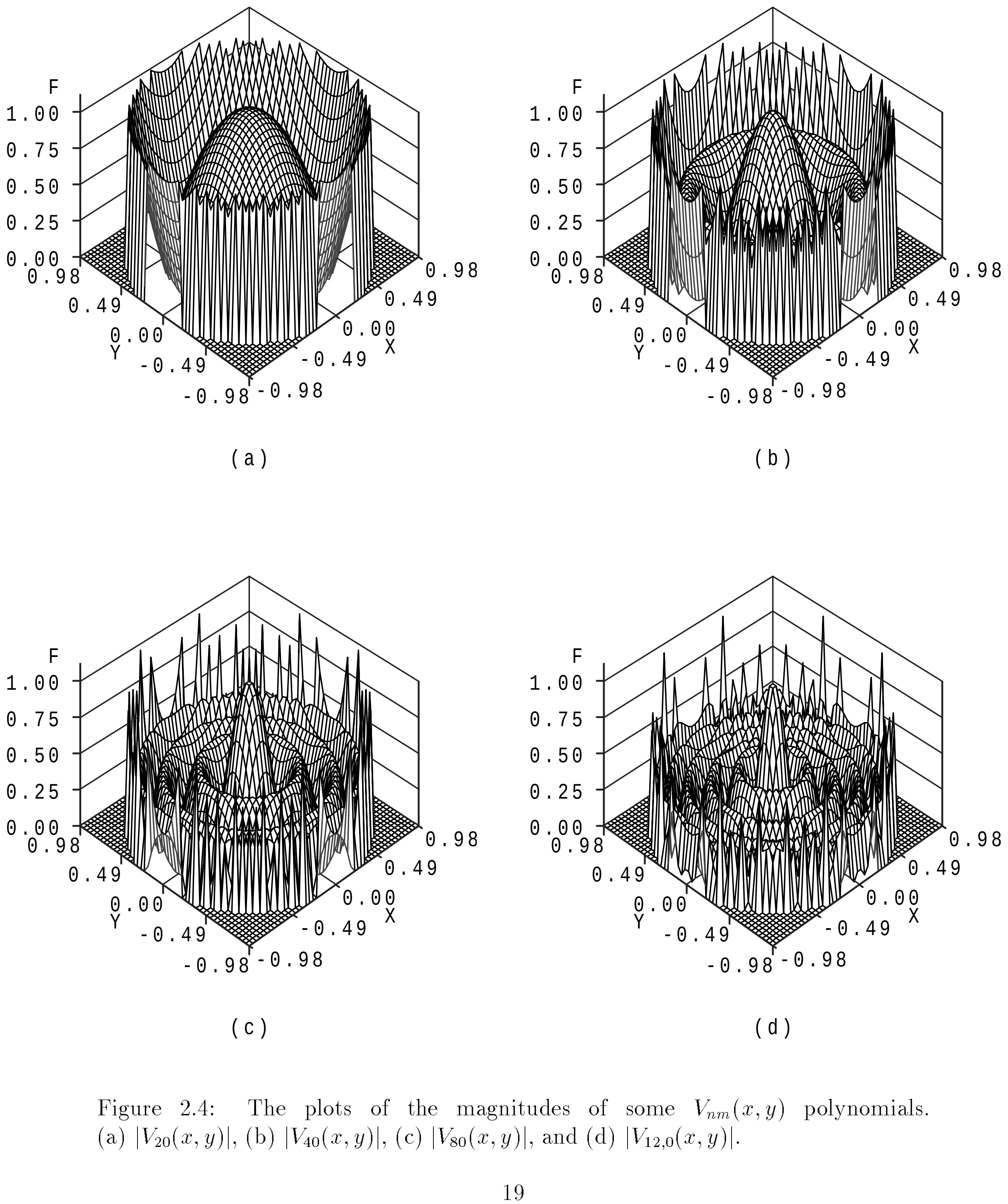
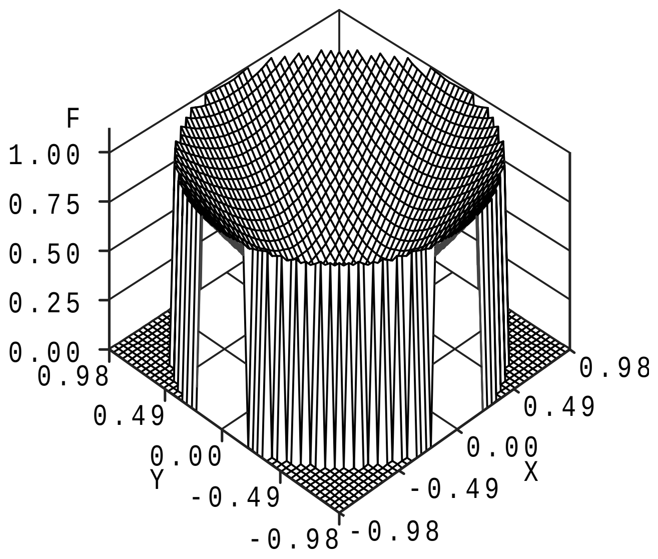
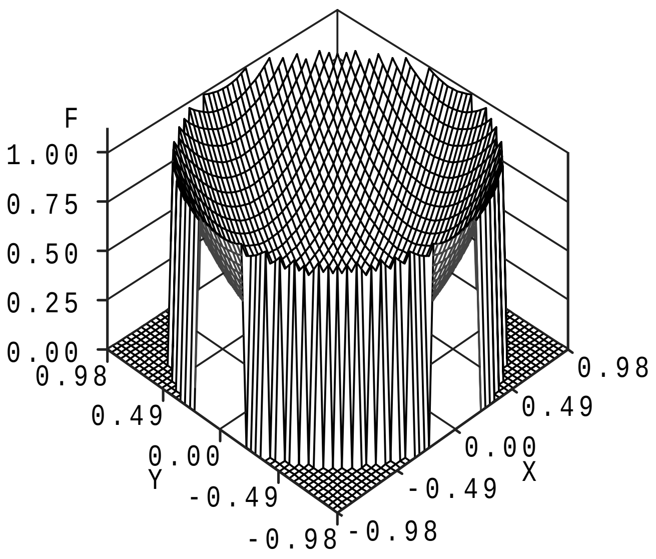


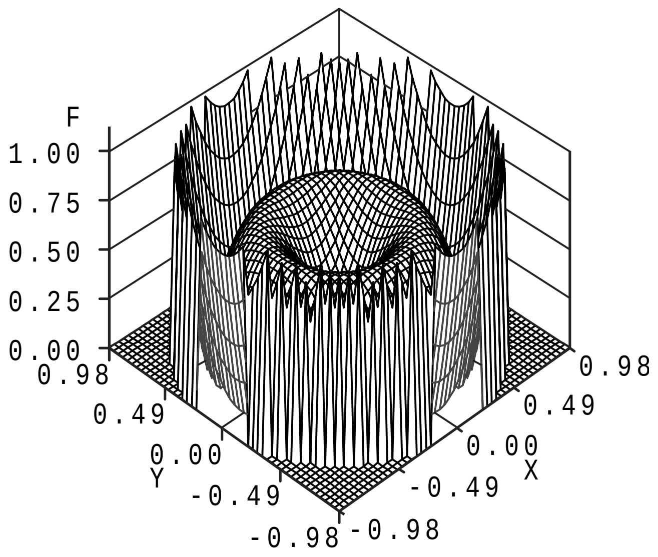
Figure 2.4: The plots of the magnitudes of some $V_{nm}(x, y)$ polynomials. (a) $|V_{20}(x, y)|$, (b) $|V_{40}(x, y)|$, (c) $|V_{80}(x, y)|$, and (d) $|V_{12,0}(x, y)|$.



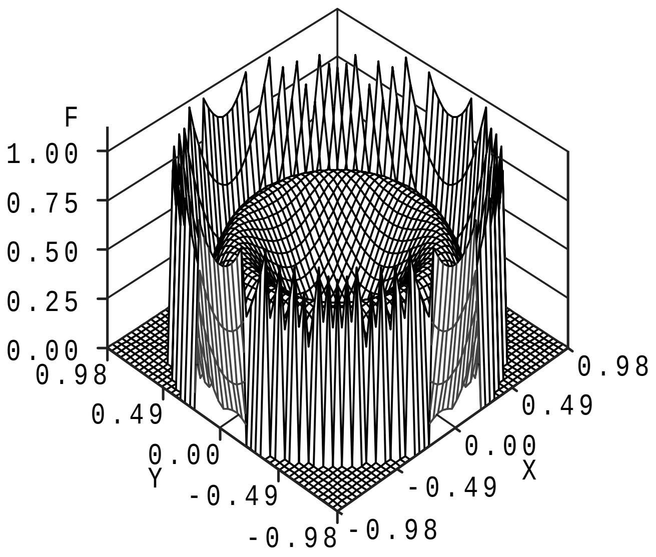
(a)



(b)



(c)



(d)

Figure 2.5: The plots of the magnitudes of some $V_{nm}(x,y)$ polynomials. (a) $|V_{11}(x,y)|$, (b) $|V_{22}(x,y)|$, (c) $|V_{31}(x,y)|$, and (d) $|V_{42}(x,y)|$.

where the asterisk denotes the complex conjugate.

As is seen from (2.33) and (2.36), the real-valued radial polynomials $\{R_{nm}(r)\}$ satisfy the relation

$$\int_0^1 R_{nl}(r)\, R_{ml}(r)\, r\, dr = \frac{1}{2(n+1)}\, \delta_{mn}. \tag{2.37}$$

The radial polynomials $R_{nm}(\rho)$ have the generating function

$$\frac{[1+t-\sqrt{1-2t(1-2\rho^2)+t^2}\,]^m}{(2t\rho)^m\sqrt{1-2t(1-2\rho^2)+t^2}} = \sum_{s=0}^{\infty} t^s R_{m+2s,m}(\rho). \tag{2.38}$$

When $m = 0$, it is interesting to see that the equation (2.38) reduces to

$$\frac{1}{\sqrt{1-2t(1-2\rho^2)+t^2}} = \sum_{s=0}^{\infty} t^s P_s(1-2\rho^2), \tag{2.39}$$

and becomes the generating function for the **Legendre** polynomials of argument $2\rho^2 - 1$, so that

$$R_{2n,0}(\rho) = P_n(2\rho^2 - 1). \tag{2.40}$$

## Zernike Moments

The complex **Zernike** moments of order $n$ with repetition $m$ for an image function $f(x, y)$ are defined as

$$A_{nm} = \frac{n+1}{\pi} \int\int_{x^2+y^2\leq 1} f(x,y)\, V_{nm}^*(\rho,\theta)\, dxdy, \tag{2.41}$$

or, in polar coordinates

$$A_{nm} = \frac{n+1}{\pi} \int_0^{2\pi}\int_0^1 f(\rho,\theta)\, R_{nm}(\rho)\, exp(-jm\theta)\, \rho\, d\rho d\theta. \tag{2.42}$$

where the real-valued radial polynomial $R_{nm}(\rho)$ is defined in (2.34).

Due to the conditions $n - |m| =$ even and $|m| \leq n$ for the **Zernike** polynomials (2.33), the set of **Zernike** polynomials contains $\frac{1}{2}(n+1)(n+2)$ linearly independent polynomials if the given maximum degree is $n$.

Since $A_{nm}^* = A_{n,-m}$, then $|A_{nm}| = |A_{n,-m}|$, therefore, one only needs to consider $|A_{nm}|$ with $m \geq 0$.

**Rotational Properties of Zernike Moments**

Under a rotation transformation, the angle of rotation of the **Zernike** moments is simply a phase factor. Therefore, the **Zernike** moments are invariant under image rotation.

If the original image and the rotated image in the same polar coordinates are denoted by $f(\rho, \theta)$ and $f^r(\rho, \theta)$ respectively, the relationship between them is

$$f^r(\rho, \theta) = f(\rho, \theta - \alpha), \tag{2.43}$$

where $\alpha$ is the angle that the original image has been rotated. Using (2.42), the **Zernike** moment of the rotated image is

$$\begin{aligned} A^r_{nm} &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho, \theta - \alpha)\, R_{nm}(\rho)\, exp(-jm\theta)\, \rho\, d\rho d\theta \\ &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho, \theta - \alpha)\, R_{nm}(\rho)\, exp(-jm(\theta - \alpha + \alpha))\, \rho\, d\rho d\theta \\ &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho, \theta - \alpha)\, R_{nm}(\rho)\, exp(-jm(\theta - \alpha))\, exp(-jm\alpha)\, \rho\, d\rho d\theta, \end{aligned}$$

therefore, the relationship between $A^r_{nm}$ and $A_{nm}$ is

$$A^r_{nm} = A_{nm}\, exp(-jm\alpha). \tag{2.44}$$

Equation (2.44) indicates that the **Zernike** moments have simple rotational transformation properties. The magnitudes of the **Zernike** moments of a rotated image function remain identical to the original image function. Thus the magnitude of the **Zernike** moment, $|A_{nm}|$, can be employed as a rotation invariant feature of the fundamental image function.

## 2.4.3 Pseudo-Zernike Moments

If we eliminate the condition $n - |m| =$ even from the the **Zernike** polynomials defined in (2.33), $\{V_{nm}\}$ becomes the set of pseudo-**Zernike** polynomials. The set of pseudo-**Zernike** polynomials was derived by Bhatia and Wolf[12] and has properties analogous to those of **Zernike** polynomials.

For the pseudo-**Zernike** polynomials, the real-valued radial polynomial $R_{nm}(\rho)$ is defined as

$$R_{nm}(\rho) = \sum_{s=0}^{n-|m|} (-1)^s \frac{(2n+1-s)!}{s!\,(n-|m|-s)!\,(n+|m|+1-s)!} \rho^{n-s}, \tag{2.45}$$

where $n = 0, 1, 2, ..., \infty$ and $m$ takes on positive and negative integers subject to $|m| \leq n$ only. Unlike the set of **Zernike** polynomials, this set of pseudo-Zernike polynomials contains $(n+1)^2$ linearly independent polynomials instead of $\frac{1}{2}(n+1)(n+2)$ if the given maximum order is $n$.

The definition of the pseudo-**Zernike** moments is the same as that of the **Zernike** moments in (2.41) and (2.42) except that the radial polynomials $\{R_{nm}(\rho)\}$ in (2.45) are used.

Since the set of pseudo-**Zernike** orthogonal polynomials is analogous to that of **Zernike** polynomials, most of the previous discussion for the **Zernike** moments can be adapted to the case of pseudo-**Zernike** moments.

# Chapter 3

# Accuracy and Efficiency of Moment Computing

## 3.1 Introduction

An essential issue in the field of pattern analysis is the recognition of patterns and characters regardless of their positions, sizes, and orientations. As discussed in the previous chapter, moments and functions of moments can be employed as the invariant global features of an image in pattern recognition, image classification, target identification, and scene analysis.

Generally, these features are invariant under image translation, scale change, and rotation only when they are computed from the original two-dimensional images. In practice, one observes the digitized, quantized, and often noisy version of the image and all these properties are satisfied only approximately. The problem of the discretization error for moment computing has been barely investigated, though some initial studies into this direction for the case of geometric moments were performed by Teh and Chin[74].

In this chapter, the detailed analysis of the discretization error for moment computing is addressed. Several new techniques developed to increase the accuracy in moment computing are provided.

## 3.2 Geometric Moments Computing

Geometric moments are the most popular type of moments. The definition of geometric moments (2.4) is rewritten here for convenience:

$$M_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p \, y^q \, f(x, y) \, dx dy. \tag{3.1}$$

If an analog original image function $f(x, y)$ is digitized into its discrete version $f(x_i, y_j)$ with an $M \times N$ array of pixels, the double integration of (3.1) must be approximated by double summations. In fact, in digital image processing, one can observe $f(x, y)$ only at discrete pixels, i.e., instead of $\{f(x, y), (x, y) \in \mathcal{R}\}$, $\{f(x_i, y_j); 1 \le i \le M, 1 \le j \le N\}$ is used. It has been a common prescription to replace $M_{pq}$ in (3.1) with its digital version

$$\widehat{M}_{pq} = \sum_{i=1}^{M} \sum_{j=1}^{N} x_i^p \, y_j^q \, f(x_i, y_j) \, \Delta x \Delta y, \tag{3.2}$$

where $\Delta x$ and $\Delta y$ are sampling intervals in the $x$ and $y$ directions. However, when the moment order increases, (3.2) cannot produce accurate results.

By (3.1), one obtains

$$\begin{aligned} M_{pq} &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p \, y^q \, f(x, y) \, dx dy \\ &= \int \int_A x^p \, y^q \, f(x, y) dx dy \\ &= \int_{a_1}^{a_2} \int_{b_1}^{b_2} x^p \, y^q \, f(x, y) dx dy, \end{aligned} \tag{3.3}$$

where

$$A = [a_1, a_2] \times [b_1, b_2]$$

is the area covered by the image.

Then, it is clear that

$$M_{pq} = \sum_{i=1}^{M} \sum_{j=1}^{N} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} \int_{y_j - \frac{\Delta y}{2}}^{y_j + \frac{\Delta y}{2}} x^p \, y^q \, f(x, y) \, dx dy, \tag{3.4}$$

where $\Delta x = x_i - x_{i-1}$ and $\Delta y = y_j - y_{j-1}$ are the sampling intervals in the $x$ and $y$ directions, and

$$x_M + \frac{\Delta x}{2} = a_2; \tag{3.5}$$

$$x_1 - \frac{\Delta x}{2} = a_1; \tag{3.6}$$

$$y_N + \frac{\Delta y}{2} = b_2; \tag{3.7}$$

$$y_1 - \frac{\Delta y}{2} = b_1. \tag{3.8}$$

By the second mean value theorem for integration, if $f$ and $g$ are integrable functions on the set $A$, and if $f$ is also continuous, then

$$\int_A f(z)g(z)dz = f(\alpha)\int_A g(z)dz \tag{3.9}$$

for some $\alpha \in A$.

Applying this result to (3.4) yields

$$M_{pq} = \sum_{i=1}^{M}\sum_{j=1}^{N} f(\alpha_i, \beta_j) \int_{x_i-\frac{\Delta x}{2}}^{x_i+\frac{\Delta x}{2}} \int_{y_j-\frac{\Delta y}{2}}^{y_j+\frac{\Delta y}{2}} x^p\, y^q\, dxdy, \tag{3.10}$$

where $(\alpha_i, \beta_j)$ belongs to the $(i, j)$ pixel.

Let us assume without loss of generality that each pixel is quantized to one value, it is normal to replace $f(\alpha_i, \beta_j)$ by $f(x_i, y_j)$. This gives the following approximation of $M_{pq}$:

$$\widehat{M}_{pq} = \sum_{i=1}^{M}\sum_{j=1}^{N} h_{pq}(x_i, y_j)\, f(x_i, y_j), \tag{3.11}$$

where

$$h_{pq}(x_i, y_j) = \int_{x_i-\frac{\Delta x}{2}}^{x_i+\frac{\Delta x}{2}} \int_{y_j-\frac{\Delta y}{2}}^{y_j+\frac{\Delta y}{2}} x^p\, y^q\, dxdy \tag{3.12}$$

represents the double integration of $x^p y^q$ over the pixel $[x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2}]\times$ $[y_j - \frac{\Delta y}{2}, y_j + \frac{\Delta y}{2}]$.

Note that

$$h_{pq}(x_i, y_j) = \frac{[(x_i + \frac{\Delta x}{2})^{p+1} - [(x_i - \frac{\Delta x}{2})^{p+1}]}{(p+1)} \frac{[(y_j + \frac{\Delta y}{2})^{q+1} - [(y_j - \frac{\Delta y}{2})^{q+1}]}{(q+1)}. \tag{3.13}$$

Then the question is how to acquire the double integration (3.12)? The simplest method to carry out the computation of $h_{pq}(x_i, y_j)$ is to use the following formula:

$$h'_{pq}(x_i, y_j) = x_i^p\, y_j^q\, \Delta x\, \Delta y \tag{3.14}$$

to replace (3.12). However, the above approximation will result in a substantial error when the order $p + q$ increases.

Since the double integration in (3.12) can be separated as

$$h_{pq}(x_i, y_j) = \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} x^p\, dx \int_{y_j - \frac{\Delta y}{2}}^{y_j + \frac{\Delta y}{2}} y^q\, dy, \tag{3.15}$$

for simplicity, we consider the single integration

$$h_p(x_i) = \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} x^p\, dx, \tag{3.16}$$

and replace $h_p(x_i)$ with

$$h'_p(x_i) = x_i^p\, \Delta x. \tag{3.17}$$

When $p = 1$, $h'_p(x_i)$ holds.

When the order $p$ increases to 2, from (3.16), we have

$$\begin{aligned}
h_2(x_i) &= \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} x^2\, dx \\
&= \frac{1}{3}[(x_i + \frac{\Delta x}{2})^3 - (x_i - \frac{\Delta x}{2})^3] \\
&= x_i^2\, \Delta x + \frac{(\Delta x)^3}{12} \\
h_2(x_i) &= h'_2(x_i) + \frac{(\Delta x)^3}{12}.
\end{aligned}$$

The approximation error for the single integration is $\frac{\Delta x^3}{12}$.

In the case of $p = 3$,

$$\begin{aligned}
h_3(x_i) &= \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} x^3\, dx \\
&= \frac{1}{4}[(x_i + \frac{\Delta x}{2})^4 - (x_i - \frac{\Delta x}{2})^4]
\end{aligned}$$

$$\begin{aligned} &= x_i^3\,\Delta x + \frac{x_i\,(\Delta x)^3}{4} \\ h_3(x_i) &= h_3'(x_i) + \frac{x_i\,(\Delta x)^3}{4}. \end{aligned}$$

The error is increased.

The approximation error will quickly get out of control when the order $p$ increases. Obviously, when the higher order moments are involved, the problem of numerical approximation error in the moment computing must be solved before any implementation.

By the well-known techniques of numerical integration[18], the integration of (3.16) can be approximated with various accuracies. For example, applying **Simpson**'s rule in the case of moment order $p = 3$, we get

$$\begin{aligned} \int_{x_i-\frac{\Delta x}{2}}^{x_i+\frac{\Delta x}{2}} x^3\,dx &= \frac{\Delta x}{2}[\frac{1}{3}\,(x_i - \frac{\Delta x}{2})^3 + \frac{4}{3}\,x_i^3 + \frac{1}{3}\,(x_i + \frac{\Delta x}{2})^3] \\ &= x_i^3\,\Delta x + \frac{x_i\,(\Delta x)^3}{4}. \end{aligned}$$

This is the same result as that of the integration.

Evidently, when the order $p$ goes higher, more accurate rules are required to limit the approximate error to a tolerable level.

As the solution, the *alternative extended Simpson's rule*

$$\begin{aligned} \int_{x_1}^{x_N} f(x)dx &= \frac{h}{48}[17f_1 + 59f_2 + 43f_3 + 49f_4 + f_5 + f_6 + ... \\ &\quad + f_{N-4} + 49f_{N-3} + 43f_{N-2} + 59f_{N-1} + 17f_N] \\ &\quad + O(\frac{1}{N^4}) \qquad (3.18) \end{aligned}$$

is employed in this research to compute the moments numerically[60]. In (3.18), $N$ is the number of points which are equally spaced apart by constant $h$ inside a single interval.

The above discussion about the approximation errors in geometric moment calculations certainly can be extended to the acquisition of the **Legendre** moments.

## 3.3 Legendre Moments Computing

### 3.3.1 Approximation Error

The $(m+n)$th order **Legendre** moment is defined in (2.31) as

$$\lambda_{mn} = \frac{(2m+1)(2n+1)}{4} \int_{-1}^{+1} \int_{-1}^{+1} P_m(x)\, P_n(y)\, f(x,y) dx dy,$$

where the $m$th order **Legendre** polynomial is

$$P_m(x) = \frac{1}{2^m m!} \frac{d^m}{dx^m} (x^2-1)^m.$$

Similar to the case of geometric moments, we can approximate $\lambda_{mn}$ by

$$\widehat{\lambda}_{mn} = \frac{(2m+1)(2n+1)}{4} \sum_{i=1}^{m} \sum_{j=1}^{n} h_{\lambda_{mn}}(x_i, y_j)\, f(x_i, y_j), \tag{3.19}$$

where

$$h_{\lambda_{mn}}(x_i, y_j) = \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} \int_{y_j - \frac{\Delta y}{2}}^{y_j + \frac{\Delta y}{2}} P_m(x)\, P_n(y)\, dx dy. \tag{3.20}$$

Since the **Legendre** polynomials $P_m(x)$ and $P_n(y)$ are independent, the double integration in (3.20) can be written as

$$\int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} \int_{y_j - \frac{\Delta y}{2}}^{y_j + \frac{\Delta y}{2}} P_m(x)\, P_n(y)\, dx dy = \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} P_m(x) dx \int_{y_j - \frac{\Delta y}{2}}^{y_j + \frac{\Delta y}{2}} P_n(y) dy. \tag{3.21}$$

Therefore, similar to the case of geometric moments, the *alternative extended Simpson's rule* can be applied in **Legendre** moment calculations to limit the approximate error to a certain level.

By using the *alternative extended Simpson's rule*, the approximation errors are reduced dramatically. It makes further use of the **Legendre** moments possible as well.

To show the improvement of accuracy in **Legendre** moment computing by adopting the *alternative extended Simpson's rule*, an experiment was designed.

If we assume that the image function $f(x,y)$ is a constant image with graylevel $a$, i.e., $f(x,y) = a$, it is easily seen that all **Legendre** moments should equal zero except $\lambda_{00} = a$. We use the sum of all **Legendre** moment squares except for the case of $m = n = 0$ as the measure to evaluate the approximation error, which has the form of

$$E_\lambda = \sum_{m=0}^{M_{max}} \sum_{n=0}^{m} \widehat{\lambda}_{mn}^2 \qquad m = n \neq 0. \tag{3.22}$$

Clearly, the smaller the $E_\lambda$ value in (3.22), the better the performance of the approximation. Five different numerical integration rules, $N = 3, N = 8, N = 13, N = 18$, and $N = 23$ are employed and all normalized $E_\lambda$'s are illustrated in Figure 3.1. The highest **Legendre** moment order used in this experiment is 56.

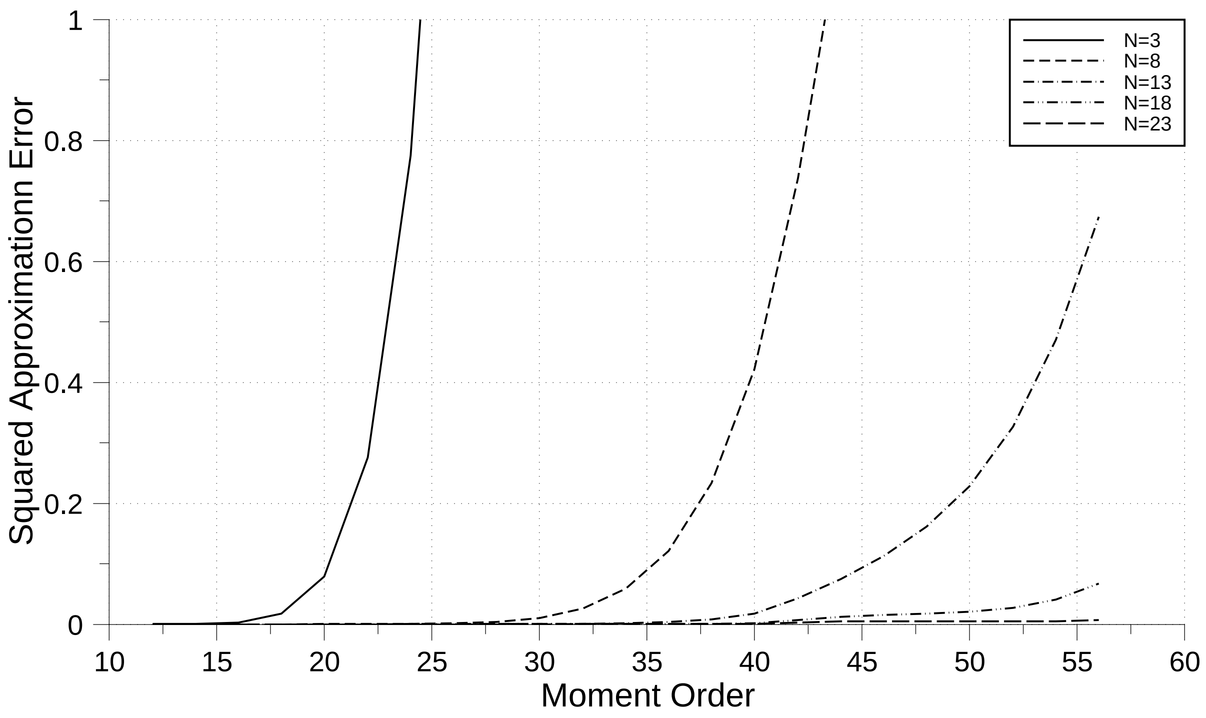


Figure 3.1: Normalized $E_\lambda$'s obtained by applying five different numerical integration rules to a constant image.

Only the $E_\lambda$'s which are less than 1.0 are presented in Figure 3.1. Each $E_\lambda$ increases very sharply after the moment order is over a certain number. As expected, the higher accuracy approximation rules perform better than the lower ones do.

| Order | $N = 3$ | $N = 8$ | $N = 13$ | $N = 18$ | $N = 23$ |
|---|---|---|---|---|---|
| 12 | 0.00003 | | | | |
| 14 | 0.00039 | | | | |
| 16 | 0.00308 | | | | |
| 18 | 0.01780 | | | | |
| 20 | 0.07873 | 0.00002 | | | |
| 22 | 0.27537 | 0.00009 | | | |
| 24 | 0.77483 | 0.00036 | 0.00001 | | |
| 26 | 1.77074 | 0.00122 | 0.00002 | | |
| 28 | | 0.00373 | 0.00008 | 0.00001 | |
| 30 | | 0.01027 | 0.00024 | 0.00002 | |
| 32 | | 0.02562 | 0.00064 | 0.00005 | 0.00001 |
| 34 | | 0.05822 | 0.00160 | 0.00012 | 0.00002 |
| 36 | | 0.12126 | 0.00372 | 0.00029 | 0.00004 |
| 38 | | 0.23284 | 0.00807 | 0.00068 | 0.00010 |
| 40 | | 0.42134 | 0.01779 | 0.00190 | 0.00043 |
| 42 | | 0.73673 | 0.04279 | 0.00728 | 0.00294 |
| 44 | | 1.15393 | 0.07487 | 0.01208 | 0.00436 |
| 46 | | | 0.11278 | 0.01524 | 0.00458 |
| 48 | | | 0.16089 | 0.01773 | 0.00459 |
| 50 | | | 0.22783 | 0.02100 | 0.00469 |
| 52 | | | 0.32621 | 0.02745 | 0.00469 |
| 54 | | | 0.47053 | 0.04105 | 0.00499 |
| 56 | | | 0.67317 | 0.06711 | 0.00695 |

Table 3.1: N is the number of points which are equally spaced apart by constant $h$ inside a single interval.

### 3.3.2 Efficiency

With the appearance of more powerful computers, it becomes practical to compute and use the higher order moments. However, the computation of moments, specifically, if the higher order moments are involved, is still a time consuming procedure. Since most of computing work in this thesis was achieved with a 25MHz 386 personal micro-computer, reducing the computing time became even more critical.

From the discussion in the previous section, the **Legendre** moments of an image function $f(x, y)$ can be obtained numerically by the formula

$$\widehat{\lambda}_{mn} = \frac{(2m+1)(2n+1)}{4} \sum_{i=1}^{m} \sum_{j=1}^{n} h_{\lambda_{mn}}(x_i, y_j)\, f(x_i, y_j) \tag{3.23}$$

where

$$h_{\lambda_{mn}}(x_i, y_j) = \int_{x_i-\frac{\Delta x}{2}}^{x_i+\frac{\Delta x}{2}} \int_{y_j-\frac{\Delta y}{2}}^{y_j+\frac{\Delta y}{2}} P_m(x)\, P_n(y)\, dx dy. \tag{3.24}$$

As we have discussed, when the higher order **Legendre** polynomials $P_m(x)$ and $P_n(y)$ are involved, to keep the approximation error under a certain level, the multi-interval step *alternative extended Simpson's rule* can be employed. However, if the well accepted recurrent formula (2.29) of the **Legendre** polynomials is applied to compute the **Legendre** polynomials $P_m(x)$ and $P_n(y)$, under the situation that $N$ takes a moderate value 10, even when the image consists of a small number of pixels, for example, 24 by 24, the computing time could be too long to be tolerated.

To speed up the computation, the most important measure is to avoid using the recurrent formula (2.29) of the **Legendre** polynomials. The fastest, the most efficient measure, of course, is to use the **Legendre** polynomials themselves. Based on this requirement, the **Legendre** polynomials up to order 55 are worked out.

Some of the higher order **Legendre** polynomials are included in Appendix A.

To speed up the computation of **Legendre** polynomials further, the well known **Horner's Rule** has been applied. For instance, a real polynomial $f(x)$ of degree $n$

or less is given by

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_2 x^2 + a_1 x + a_0 \tag{3.25}$$

with the coefficients $a_0$, $a_1$, $a_2$, ..., $a_{n-1}$, and $a_n$ representing real numbers. In programming practice, assuming that all coefficients are nonzero, a straightforward naive approach to compute this polynomial will cost $\frac{n(n+1)}{2}$ multiplications and $n$ addition operations. However, with **Horner's Rule**, the polynomial $f(x)$ can be expressed by writing

$$f(x) = ((...((a_n x + a_{n-1})x + a_{n-2})x + ...)x + a_1)x + a_0. \tag{3.26}$$

With this new formula, it requires only $n$ multiplications and $n$ additions to compute the polynomial. Since the operation of multiplication takes much longer than that of addition, in terms of the computation time, the new formula is about $\frac{n+1}{2}$ times faster than the straightforward naive approach.

Adopting the high order **Legendre** polynomials listed in Appendix A and **Horner's Rule** has dramatically reduced the computing time required in **Legendre** moments computation, and more importantly, made this research possible.

## 3.4 Zernike Moments

### 3.4.1 Introduction

As mentioned in the previous chapter, the complex **Zernike** moments of order $n$ with repetition $m$ for a continuous image function $f(x, y)$ are defined as

$$A_{nm} = \frac{n+1}{\pi} \int\int_{x^2+y^2 \leq 1} f(x, y)\, V_{nm}^*(\rho, \theta) dx dy \tag{3.27}$$

in the $xy$ image plane, and

$$A_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho, \theta)\, R_{nm}(\rho)\, exp(-jm\theta)\, \rho d\rho d\theta \tag{3.28}$$

in polar coordinates. The real-valued radial polynomial $R_{nm}(\rho)$ is defined as

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s!\,(\frac{n+|m|}{2}-s)!\,(\frac{n-|m|}{2}-s)!} \rho^{n-2s}, \tag{3.29}$$

where $n - |m| =$ even and $|m| \leq n$.

The feature of invariance under image rotation makes the **Zernike** function one of the most important moments. However, the nature of **Zernike** moment computing, using the summation of square pixels to achieve the computation defined on a unit disk, makes it more difficult to solve the accuracy problems.

For a digitized image function $f(x,y)$, as discussed in the previous chapter, the double integration of (2.41) can be approximated by double summation:

$$\widehat{A}_{nm} = \frac{n+1}{\pi} \sum_{x_i} \sum_{y_j} h_{A_{nm}}(x_i, y_j)\, f(x_i, y_j), \qquad x_i^2 + y_j^2 \leq 1, \tag{3.30}$$

where

$$h_{A_{nm}}(x_i, y_j) = \int_{x_i-\frac{\Delta x}{2}}^{x_i+\frac{\Delta x}{2}} \int_{y_j-\frac{\Delta y}{2}}^{y_j+\frac{\Delta y}{2}} V_{nm}^*(\rho, \theta) dx dy. \tag{3.31}$$

From the definitions of $\widehat{A}_{nm}$ and $h_{A_{nm}}(x_i, y_j)$, we can find that there are two kinds of major errors in the computation of the **Zernike** moments $\widehat{A}_{nm}$, geometric and approximate.

## 3.4.2 Geometric Error

When computing the **Zernike** moments, if the centre of a pixel falls inside the border of unit disk $x^2 + y^2 \leq 1$, this pixel will be used in the computation; if the centre of the pixel falls outside the unit disk, the pixel will be discarded. Therefore, the area covered by the moment computation is not exactly the area of the unit disk.

Figure 3.2 shows the different areas covered by a unit disk and all pixels whose centres fall inside the unit disk.

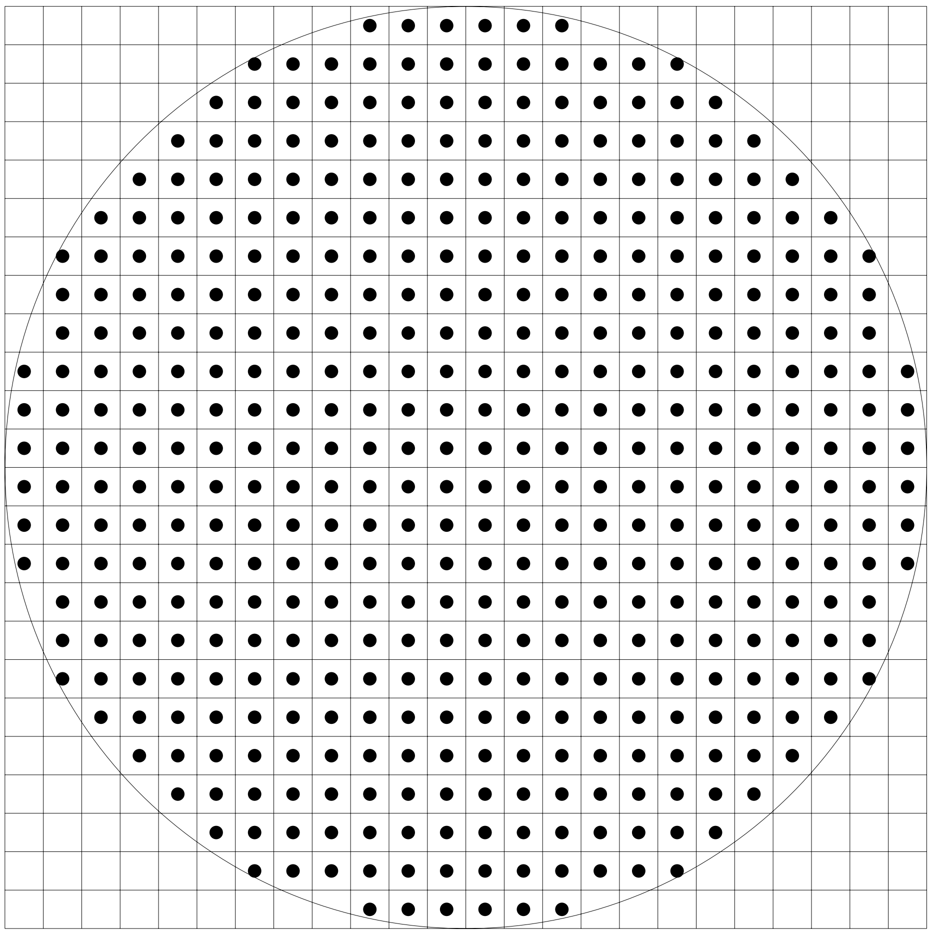


Figure 3.2: Different areas covered by a disk and all pixels whose centres fall inside the disk.

In the case of **Zernike** moment, the unit disk is located in a 2 units $\times$ 2 units square which is composed of $n \times n$ pixels. Therefore, the area of the unit disk is $\pi$. If $A(n)$ represents the number of pixels whose centres fall inside the unit disk, the summation of the areas of all these pixels is

$$A_{pixels} = A(n)\,\frac{4}{n^2}. \tag{3.32}$$

Now, the geometric error between the unit disk and the summation of all the pixels used in the **Zernike** moment computation is

$$R(n) = A(n)\,\frac{4}{n^2} - \pi. \tag{3.33}$$

For the **Zernike** moment computing, it is crucial to know, when $n$ tends to infinity, i.e., if the number of pixels is increasing, how fast the geometric error $R(n)$ converges to zero.

In fact, this issue is closely related to a famous problem in analytic number theory, due originally to **Gauss** and referred as *The Lattice Points of a Circle Problem* [47].

## 3.4.3 The Lattice Points of a Circle Problem

Let $A(x)$ be the number of lattice points $(u, v)$ inside or on the circle $u^2 + v^2 = x$, so that $A(x) \sim \pi x$ as $x$ tends to infinity. Let

$$R(x) = A(x) - \pi x, \tag{3.34}$$

and let $\vartheta$ be the lower bound on the number $\theta$ such that

$$R(x) = O(x^{\theta}). \tag{3.35}$$

We list some significant steps in the history of the estimation of $R(x)$ here:

| | | |
|---|---|---|
| Gauss | (1834), | $\theta = \frac{1}{2} = 0.5000000000$; |
| Sierpinski | (1906), | $\theta = \frac{1}{3} = 0.3333333333...$; |

| | | |
|---|---|---|
| Walfisz | (1927), | $\theta = \frac{163}{494} = 0.329959514...;$ |
| Titchmarsh[76] | (1935), | $\theta = \frac{15}{46} = 0.326086957...;$ |
| Hua[38] | (1942), | $\theta = \frac{13}{40} = 0.325000000;$ |

and more recently

Iwaniec and Mozzochi[40] (1988), $\theta = \frac{7}{22} = 0.318181818...$ .

In the other direction, it has long been known that $\vartheta \geq 0$, and this result also has been improved by Hardy[31], Landau[47], and Ingham[39] to:

$$\lim_{x\to\infty} \frac{R(x)}{x^{\frac{1}{4}}(logx)^{\frac{1}{4}}} = 0, \tag{3.36}$$

and

$$\overline{\lim_{x\to\infty}} \frac{R(x)}{x^{\frac{1}{4}}} = +\infty. \tag{3.37}$$

This result shows that the smallest possible $\theta$ cannot reach $\theta = \frac{1}{4}$. This still remains an open problem in the number theory.

Comparing our problem with *The Lattice Points of a Circle Problem*, we find that the $x$ in (3.34) is equivalent to $n^2$ in (3.33) when both $x$ and $n$ tend to infinity. On the other hand, the number of lattice points in *The Lattice Points of a Circle Problem* and the number of pixels within the unit disk in our problem are identical when the area of each lattice is 1, and the area of each pixel is $\frac{4}{n^2}$. Then, it follows that (3.33) can be

$$\begin{aligned} R(n) &= A(n)\frac{4}{n^2} - \pi \\ &= A(x)\frac{4}{x} - \pi \\ &= R(x)\frac{4}{x} \\ R(n) &= O(x^{\theta-1}). \end{aligned} \tag{3.38}$$

Therefore, we obtain

$$R(n) = O(n^{2(\theta-1)}). \tag{3.39}$$

With the latest result from Iwaniec and Mozzochi, the geometric error in the **Zernike** moment computing is

$$R(n) = O(n^{-\frac{15}{11}}). \tag{3.40}$$

| | n=24 | n=48 | n=64 | n=128 |
|---|---|---|---|---|
| $n^{-1}$ | 0.0416666 | 0.0208333 | 0.0156250 | 0.0078125 |
| $n^{-\frac{15}{11}}$ | 0.0131188 | 0.0050979 | 0.0034437 | 0.0013382 |
| $n^{-\frac{3}{2}}$ | 0.0085051 | 0.0030070 | 0.0019531 | 0.0006905 |

Table 3.2: Range of geometric errors for several commonly used image sizes.

Several commonly used image sizes are employed here to show the range of geometric errors in cases of $n^{-1}, n^{-\frac{15}{11}}$, and $n^{-\frac{3}{2}}$, respectively. The results are displayed in Table 3.2.

Like the case in our experiment, when $n$ is 24, with the best result from Iwaniec and Mozzochi[40], the geometric error is at the range of

$$n^{-\frac{15}{11}} = 0.0131188....$$

Obviously, this is not a very encouraging result. Since the higher order **Zernike** moments are the accumulations of the lower order computed **Zernike** moments, if the geometric error is around $O(n^{-\frac{15}{11}})$, when the order of **Zernike** moments goes higher, the accumulated geometric errors would quickly get out of control and the use of higher order **Zernike** moments would be severely handicaped.

### 3.4.4 Approximation Error

As discussed previously, in the $xy$ image plane with a digitized image function $f(x, y)$, the **Zernike** moments of order $n$ with $m$ repetitions are

$$\widehat{A}_{nm} = \frac{n+1}{\pi} \sum_{x_i} \sum_{y_j} h_{A_{nm}}(x_i, y_j)\, f(x_i, y_j), \qquad x_i^2 + y_j^2 \leq 1, \tag{3.41}$$

where

$$h_{A_{nm}}(x_i, y_j) = \int_{x_i-\frac{\Delta x}{2}}^{x_i+\frac{\Delta x}{2}} \int_{y_j-\frac{\Delta y}{2}}^{y_j+\frac{\Delta y}{2}} V_{nm}^*(\rho, \theta) dxdy. \tag{3.42}$$

The **Zernike** polynomials $V_{nm}^*(\rho, \theta)$ are defined as

$$V_{nm}(\rho, \theta) = R_{nm}(\rho, \theta)\, exp(jm\theta), \tag{3.43}$$

where the Radial polynomial $R_{nm}(\rho)$ is

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s!\,(\frac{n+|m|}{2}-s)!\,(\frac{n-|m|}{2}-s)!} \rho^{n-2s}. \tag{3.44}$$

Unlike the **Legendre** polynomials $P_m(x)$ and $P_n(y)$, which are independent, the **Zernike** polynomials $V_{nm}(\rho, \theta)$ are two-dimensional functions of $\rho$ and $\theta$. Therefore, to reduce the approximation error in the **Zernike** moment computation is more complex than that of **Legendre** moments.

Under this particular situation, naturally, the way to reduce the approximation error is to compute the double integrations and $h_{A_{nm}}(x_i, y_j)$ by using some well known cubature formulas[23].

Suppose we have a two-dimensional domain $\Omega$ and wish to approximate $\int_\Omega f(\mathbf{x})d\Omega$. Let $f \in \Omega$, and $\mathbf{a}^T = (a, b) \in \Omega$. We have the **Taylor** expansion of the integrand function $f(\mathbf{x})$:

$$\begin{aligned} f(\mathbf{x}) &= f(x,y) \\ &= f(\mathbf{a}) + (x-a)f_x(\mathbf{x}) + (y-b)f_y(\mathbf{a}) \\ &\quad + \frac{1}{2}[(x-a)^2 f_{xx}(\mathbf{a}) + 2(x-a)(y-b)f_{xy}(\mathbf{a}) + (y-b)^2 f_{yy}(\mathbf{a})] \\ &\quad + ... + \frac{1}{(n-1)!}[\sum_{i=0}^{n-1}\binom{n-1}{i}(x-a)^{n-i-1}(y-b)^i \frac{\partial^{n-1} f(\mathbf{a})}{\partial x^{n-i-1}\partial y^i}] \\ &\quad + error. \end{aligned} \tag{3.45}$$

Let

$$I_\Omega f = \int_\Omega f(\mathbf{x})d\Omega,$$

then it follows[23] that:

$$\begin{aligned} I_\Omega f &= |\Omega| f(\mathbf{a}) + f_x(\mathbf{a}) I_\Omega(x-a) + f_y(\mathbf{a}) I_\Omega(y-b) + ... \\ &+ \frac{1}{(n-1)!} \sum_{i=0}^{n-1} \binom{n-1}{i} \frac{\partial^{n-1} f(\mathbf{a})}{\partial x^{n-i-1} \partial y^i} I_\Omega[(x-a)^{n-i-1}(y-b)^i] \\ &+ error, \end{aligned} \tag{3.46}$$

and

$$\begin{aligned} C_n f &= f(\mathbf{a}) \sum_{i=1}^{n} A_i + f_x(\mathbf{a}) \sum_{i=1}^{n} A_i(x_i - a) + f_y(\mathbf{a}) \sum_{i=1}^{n} A_i(y_i - b) + ... \\ &+ \frac{1}{(n-1)!} [\sum_{j=0}^{n-1} \binom{n-1}{j} \frac{\partial^{n-1} f(\mathbf{a})}{\partial x^{n-j-1} \partial y^j} \sum_{i=1}^{n} A_i (x_i - a)^{n-j-1} (y-b)^j]. \end{aligned} \tag{3.47}$$

Taking $\mathbf{a} = 0$, and comparing $I_\Omega f$ with $C_n f$, we obtain the following equations which can determine the weights of a cubature formula:

$$\sum_{i=1}^{n} A_i x_i^{k-j} y_i^j = I_\Omega x^{k-j} y_i^j \qquad j = 0, 1, ..., k \leq n, \tag{3.48}$$

where $n$ is the number of nodes inside $\Omega$, and

$$I_\Omega = \int_\Omega f(x,y) d\Omega. \tag{3.49}$$

This is a linear system of equations for the weights $A_i$. For example, taking $j \leq 2$, we obtain the equations

$$\begin{array}{lll} \sum_{i=1}^{n} A_i = |\Omega| & & \\ \sum_{i=1}^{n} A_i x_i = I_\Omega x, & \sum_{i=1}^{n} A_i y_i = I_\Omega y & \\ \sum_{i=1}^{n} A_i x_i^2 = I_\Omega x^2, & \sum_{i=1}^{n} A_i x_i y_i = I_\Omega xy, & \sum_{i=1}^{n} A_i y_i^2 = I_\Omega y^2. \end{array}$$

To achieve sufficient accuracy, traditionally, we can increase the number of nodes in each pixel. Solving the linear system equations obtained from the (3.48), we can find the weights for all nodes inside each pixel.

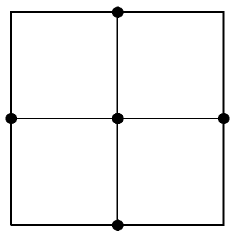


Figure 3.3: 5-dimensional formula I.

One simple formula which can be adopted to increase the approximation accuracy is shown in Figure 3.3. Using the unit height, we can determine the weights of the cubature formula

$$C_5 f = A_1 f(0,0) + A_2 f(0,1) + A_3 f(1,0) + A_4 f(0,-1) + A_5 f(-1,0). \tag{3.50}$$

Employing (3.48), we obtain five linear system equations:

$$\left\{\begin{array}{l} \sum_{i=1}^{5} = |\Omega| \\ \sum_{i=1}^{5} A_i x_i = I_\Omega x \\ \sum_{i=1}^{5} A_i y_i = I_\Omega y \\ \sum_{i=1}^{5} A_i x_i^2 = I_\Omega x^2 \\ \sum_{i=1}^{5} A_i y_i^2 = I_\Omega y^2 \end{array}\right. \tag{3.51}$$

where

$$I_\Omega = \int_\Omega f(x,y) d\Omega.$$

From (3.51), straightforwardly, we can obtain the 5-dimensional cubature formula by solving the following five linear system equations:

$$\left\{\begin{array}{cccccccc} A_1 & +A_2 & +A_3 & +A_4 & +A_5 & = & 4 \\ & & +A_3 & & -A_5 & = & 0 \\ & +A_2 & & -A_4 & & = & 0 \\ & & +A_3 & & +A_5 & = & 2/3 \\ & +A_2 & & +A_4 & & = & 2/3. \end{array}\right. \tag{3.52}$$

The solutions of the above five linear system equations lead to

$$C_5 f = \frac{1}{3}\{8f(0,0) + f(0,1) + f(1,0) + f(0,-1) + f(-1,0)\}, \tag{3.53}$$

where $f(x_1, x_2)$ is a two dimensional function.

We can use another type of the 5-dimensional formula, which is shown in Figure 3.4, as well.

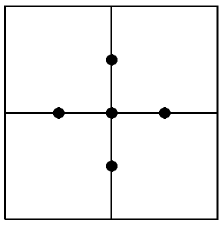


Figure 3.4: 5-dimensional formula II.

With the same five equations in (3.51) but different $x_i$ and $y_i$ values, we have a set of five linear system equations

$$\left\{\begin{array}{llllllll} A_1 & +A_2 & +A_3 & +A_4 & +A_5 & = & 4 \\ & & +0.5A_3 & & -0.5A_5 & = & 0 \\ & +0.5A_2 & & -0.5A_4 & & = & 0 \\ & & +0.25A_3 & & +0.25A_5 & = & 2/3 \\ & +0.25A_2 & & +0.25A_4 & & = & 2/3. \end{array}\right. \tag{3.54}$$

which produce the 5-dimensional cubature formula:

$$C_5 f = \frac{4}{3}\{-f(0,0) + f(0,1) + f(1,0) + f(0,-1) + f(-1,0)\}. \tag{3.55}$$

The number of nodes in each pixel can be increased further to achieve higher accuracy. An example is to use the 13-dimensional cubature formula, which is illustrated in Figure 3.5, to reduce the approximation error.

We can determine the weights of the cubature formula

$$\begin{aligned} C_{13} f \;=\; & A_1 f(0,0) + A_2 f(0.5,0.5) + A_3 f(0.5,-0.5) + A_4 f(-0.5,-0.5) \\ & + A_5 f(-0.5,0.5) + A_6 f(0,1) + A_7 f(1,1) + A_8 f(1,0) + A_9 f(1,-1) \\ & + A_{10} f(0,-1) + A_{11} f(-1,-1) + A_{12} f(-1,0) + A_{13} f(-1,1) \end{aligned} \tag{3.56}$$

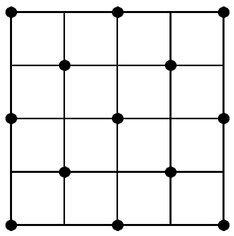


Figure 3.5: 13-dimensional formula

.

the same way we did for the 5-dimensional formula by employing (3.48) and solving thirteen linear system equations. The equations are listed in the Appendix of this chapter, and the solutions of these equations lead to

$$\begin{aligned} C_{13}f &= \frac{1}{45}\{120f(0,0) \\ &\quad +16[f(0.5,0.5)+f(0.5,-0.5)+f(-0.5,-0.5)+f(-0.5,0.5)] \\ &\quad -9[f(0,1)+f(1,0)+f(0,-1)+f(-1,0)] \\ &\quad +8[f(1,1)+f(1,-1)+f(-1,-1)+f(-1,1)]\} \end{aligned} \tag{3.57}$$

It is possible to impose additions on the weights $A_i$ to reduce the number of linear system equations and make the determination of the $A_i$ easier. For example, we can assume that the thirteen dimensional cubature formula is

$$\begin{aligned} C_{13}f &= A_1(0,0) \\ &\quad + A_2[f(0.5,0.5)+f(0.5,-0.5)+f(-0.5,-0.5)+f(-0.5,0.5)] \\ &\quad + A_3[f(0,1)+f(1,0)+f(0,-1)+f(-1,0)] \\ &\quad + A_4[f(1,1)+f(1,-1)+f(-1,-1)+f(-1,1)], \end{aligned} \tag{3.58}$$

which leads to

$$\begin{cases} \sum_{i=1}^{13} = |\Omega| \\ \sum_{i=1}^{13} A_i x_i^2 = I_\Omega x^2 \\ \sum_{i=1}^{13} A_i x_i^2 y_i^2 = I_\Omega x^2 y^2 \\ \sum_{i=1}^{13} A_i x_i^4 = I_\Omega x^4, \end{cases} \tag{3.59}$$

and

$$\begin{cases} A_1 & +4A_2 & +4A_3 & +4A_4 & = & 4 \\ & +A_2 & +2A_3 & +4A_4 & = & 2/3 \\ & +.25A_2 & & +4A_4 & = & 4/9 \\ & +.25A_2 & +2A_3 & +4A_4 & = & 2/3. \end{cases} \tag{3.60}$$

Therefore, the 13-dimensional cubature formula (3.58) can be written as:

$$\begin{aligned} C_{13}f &= \frac{1}{45}\{104f(0,0) \\ &+ 16[f(0.5,0.5) + f(0.5,-0.5) + f(-0.5,-0.5) + f(-0.5,0.5)] \\ &- [f(0,1) + f(1,0) + f(0,-1) + f(-1,0)] \\ &+ 4[f(1,1) + f(1,-1) + f(-1,-1) + f(-1,1)]\}. \end{aligned} \tag{3.61}$$

If we ignore the geometric error and let the image function $f(x,y)$ be a constant image with graylevel $a$, like the case of **Legendre** moments, all **Zernike** moments should equate to zero except $A_{00} = a$. Therefore, we can use the following measure to evaluate the approximation errors of the **Zernike** moments

$$E_A = \sum\sum |\widehat{A}_{nm}|^2 \qquad m = n \neq 0. \tag{3.62}$$

The two different types of 5-dimensional cubature formulas, the 13-dimensional formula with different sets of weights, and the simplest 1-dimensional formula are employed to evaluate the approximation errors in the computation of the **Zernike** moment. All normalized $E_A$'s which are less that 1.0 are illustrated in Figure 3.6, and their values are listed in Table 3.3.

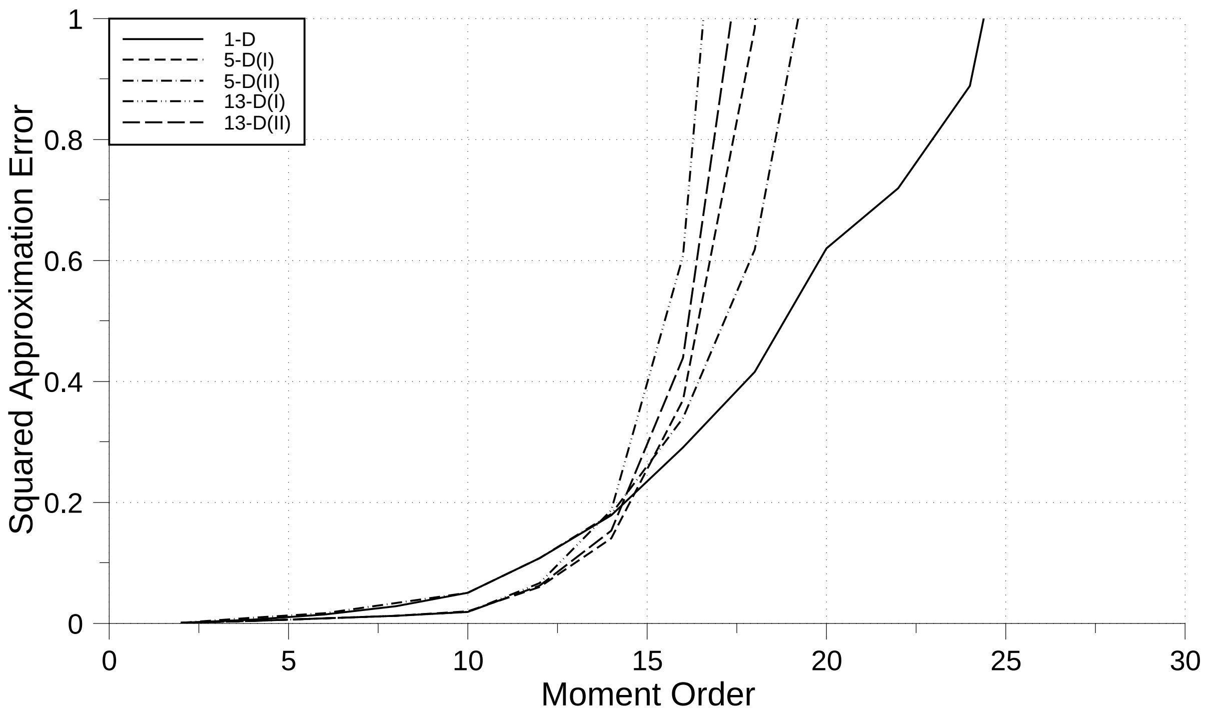


Figure 3.6: Normalized approximation errors obtained by applying five different types of multi-dimensional cubature formulas on a constant image.

| Order | 1-D | 5-D(I) | 5-D(II) | 13-D(I) | 13-D(II) |
|---|---|---|---|---|---|
| 2 | 0.0009 | 0.0007 | 0.0009 | 0.0007 | 0.0007 |
| 4 | 0.0056 | 0.0040 | 0.0088 | 0.0040 | 0.0040 |
| 6 | 0.0141 | 0.0080 | 0.0167 | 0.0079 | 0.0079 |
| 8 | 0.0281 | 0.0122 | 0.0334 | 0.0120 | 0.0121 |
| 10 | 0.0502 | 0.0191 | 0.0499 | 0.0189 | 0.0188 |
| 12 | 0.1077 | 0.0603 | 0.1071 | 0.0661 | 0.0615 |
| 14 | 0.1785 | 0.1401 | 0.1803 | 0.1863 | 0.1531 |
| 16 | 0.2907 | 0.3693 | 0.3395 | 0.6067 | 0.4388 |
| 18 | 0.4149 | 0.9837 | 0.6178 | 1.9872 | 1.2722 |
| 20 | 0.6193 | 2.7800 | 1.2479 | | |
| 22 | 0.7192 | | | | |
| 24 | 0.8879 | | | | |
| 26 | 1.4654 | | | | |

Table 3.3: Values of the normalized approximation errors from appling five different types of multi-dimensional cubature formulas on a constant image.

Table 3.3 and Figure 3.6 show that the multi-dimensional formulas could not produce better results than the simplest 1-point formula did. In other words, the traditional method to reduce the approximation errors could not improve the accuracy in this particular situation.

The reason that these multi-dimensional cubature formulas do not work is that the **Zernike** moments are defined within the unit disk $x^2 + y^2 \leq 1$. Since we use all pixels whose centres fall into the unit disk for the **Zernike** moment computing, the one-dimensional formula will not produce extra errors because all $f(x_i, y_j)$ used in the computing are covered by the definition. However, when multi-dimensional cubature formulas are adopted, on the boundary of the unit disk, some $f(x_i, y_j)$ used to compute the pixels on the boundary will not fit the condition $x^2 + y^2 \leq 1$. For example, with the condition $x^2 + y^2 \leq 1$, there will be respectively 40, 16, and 140 nodes used in the **Zernike** moment computation which fall outside the unit disk for 5-dimensional formula I, 5-dimensional formula II, and 13-dimensional formulas. This certainly brings extra errors to the **Zernike** moments and makes the approximation errors go up quickly.

### 3.4.5 A New Proposed Solution to Reduce Approximation Error

We redefine the digitized version **Zernike** moments as

$$\widehat{A}_{nm} = \frac{n+1}{\pi} \sum_{x_i} \sum_{y_j} h_{A_{nm}}(x_i, y_j)\, f(x_i, y_j), \qquad x_i^2 + y_j^2 \leq 1 - \gamma, \tag{3.63}$$

where

$$h_{A_{nm}}(x_i, y_j) = \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} \int_{y_j - \frac{\Delta y}{2}}^{y_j + \frac{\Delta y}{2}} V_{nm}^*(\rho, \theta)\, dx dy, \tag{3.64}$$

and $\gamma$ is an adjustable factor. For example, in our case, we let

$$\gamma = \frac{\Delta x + \Delta y}{4} + \epsilon, \tag{3.65}$$

where $\epsilon$ is an arbitrary small number. Then, with this new condition

$$x_i^2 + y_j^2 \leq 1 - \frac{\Delta x + \Delta y}{4} - \epsilon,$$

the number of nodes that fall outside the unit disk will be reduced to 16, 0, and 68 for the 5-dimensional formula I, 5-dimensional formula II, and 13-dimensional formulas, respectively. Obviously, under this condition, the geometric errors will be higher.

Employing 0.0001 as the $\epsilon$ value, we re-evaluate all five different formulas discussed above. Table 3.4 and Figure 3.7 show the results.

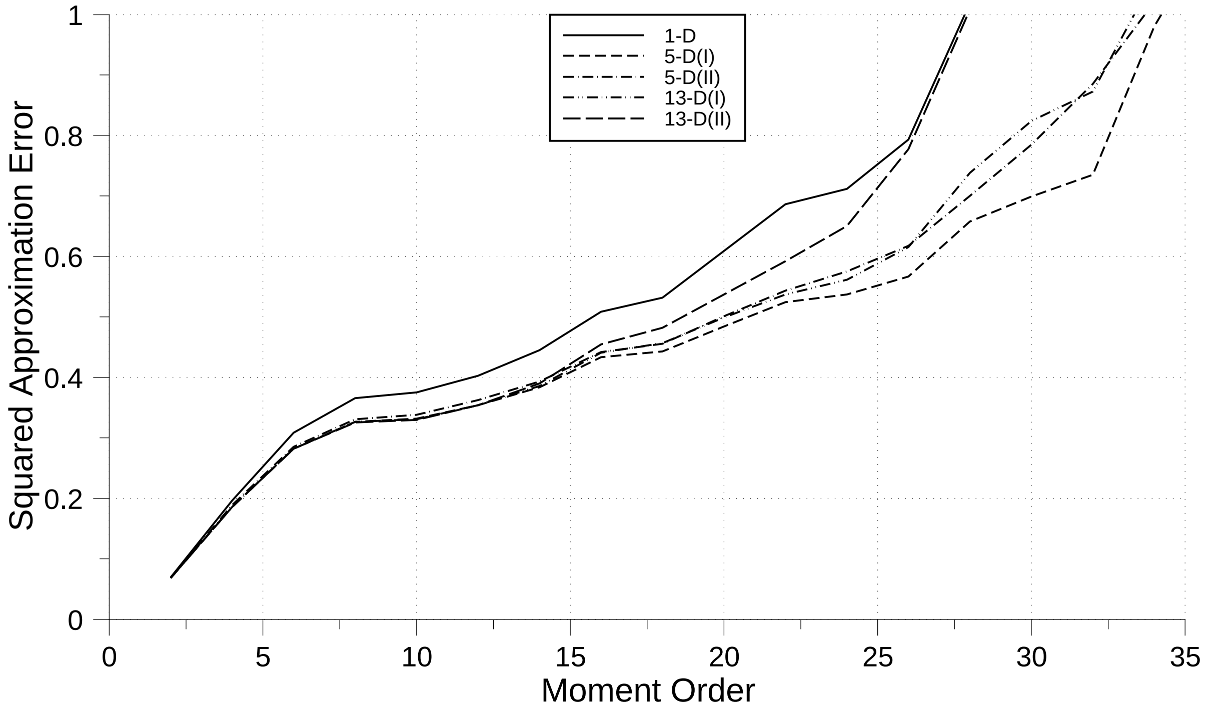


Figure 3.7: Normalized $E_A$'s obtained by applying five different types of multi-dimensional cubature formulas on a constant image with the new proposed technique.

Compared with Figure 3.6, Figure 3.7 shows that the error $E_A$ goes up quickly to the level of 30% for all five different formulas, then the ratios of increase slow

| Order | 1-D | 5-D(I) | 5-D(II) | 13-D(I) | 13-D(II) |
|---|---|---|---|---|---|
| 2 | 0.0698 | 0.0681 | 0.0683 | 0.0681 | 0.0681 |
| 4 | 0.1961 | 0.1858 | 0.1886 | 0.1858 | 0.1858 |
| 6 | 0.3082 | 0.2820 | 0.2852 | 0.2819 | 0.2818 |
| 8 | 0.3661 | 0.3266 | 0.3307 | 0.3264 | 0.3259 |
| 10 | 0.3752 | 0.3314 | 0.3385 | 0.3307 | 0.3298 |
| 12 | 0.4026 | 0.3544 | 0.3621 | 0.3538 | 0.3537 |
| 14 | 0.4445 | 0.3840 | 0.3928 | 0.3857 | 0.3897 |
| 16 | 0.5083 | 0.4333 | 0.4418 | 0.4410 | 0.4540 |
| 18 | 0.5313 | 0.4429 | 0.4560 | 0.4569 | 0.4820 |
| 20 | 0.6089 | 0.4839 | 0.5008 | 0.4989 | 0.5373 |
| 22 | 0.6863 | 0.5242 | 0.5434 | 0.5374 | 0.5918 |
| 24 | 0.7116 | 0.5375 | 0.5750 | 0.5610 | 0.6505 |
| 26 | 0.7928 | 0.5667 | 0.6177 | 0.6151 | 0.7776 |
| 28 | 1.0185 | 0.6572 | 0.7004 | 0.7384 | 1.0084 |
| 30 |  | 0.6993 | 0.7843 | 0.8240 |  |
| 32 |  | 0.7353 | 0.8848 | 0.8721 |  |
| 34 |  | 0.9804 | 1.0203 | 1.0620 |  |
| 36 |  | 1.1503 |  |  |  |

Table 3.4: Values of the normalized $E_A$'s from appling five different types of multi-dimensional cubature formulas on a constant image with the new proposed technique.

down. As expected, all four multi-dimensional formulas produce better results than the simplest one-dimensional formula does.

The results shown in Figure 3.7 and Table 3.4 are better than those of Figure 3.6 and Table 3.3. However, one is reluctant to call the digitized version of **Zernike** moments under the new condition a flawless solution to the approximation error problem of the **Zernike** moment computing. Though it indeed controls the increase ratio of the error to a lower level under certain circumstances, we would like to use it as an alternative rather than call it a complete solution to compute the **Zernike** moments.

## 3.5 Conclusions

In this chapter, the problems of accuracy and efficiency in moment computing are discussed.

### 3.5.1 Legendre Moment Computing

It has been shown that most problems concerning accuracy and efficiency in the **Legendre** moment computing have been solved. Therefore, we are able to use the higher order of the **Legendre** moments in further research confidently.

### 3.5.2 Zernike Moment Computing

Because of the nature of the **Zernike** moment calculations, the two major problems in the **Zernike** moment computing, geometric and approximation errors, are more difficult.

#### Geometric Error

We adopted the latest results from a classical problem in Number Theory, *The Lattice Points of a Circle*, in our study on the geometric error of **Zernike** moment computing. It shows that the geometric error is

$$R(n) = O(n^{-\frac{15}{11}}). \tag{3.66}$$

For example, in the case of $n = 24$, $n^{-\frac{15}{11}} = 0.0131188....$

We have to admit that the errors in the range of $O(n^{-\frac{15}{11}})$ are too large to be ignored. More seriously, since the higher order **Zernike** moments are the accumulations of the lower order computed **Zernike** moments, when the order of **Zernike** moments goes higher, the accumulated geometric errors will be quickly out of control.

Increasing the size of an image, or $n$, will indeed make the geometric errors $R(n)$ for the individual moments smaller. However, in many cases, to increase $n$ will result in higher order moments being required to provide the needed image features. Therefore, to increase the size of an image in order to reduce the geometric errors is not recommended.

Certainly, the existence of the geometric errors severely handicaps the usage of the **Zernike** moments.

### Approximation Error

The approximation error in the **Zernike** moment computing is discussed in this chapter as well. To reduce the approximation error, some well known cubature formulas are applied. However, to implement the multi-dimensional cubature formulas cannot improve the accuracy significantly.

The digitized **Zernike** moments are achieved from the summation of square pixels, whose centres fall inside the unit disk. However, on the unit disk boundary, a pixel whose centre falls inside the boundary does not mean that the entire pixel falls into the unit disk. Therefore, the multi-dimensional cubature formulas, which use a number of nodes inside a pixel to achieve sufficient accuracy, will no longer be valid.

To make the the multi-dimensional cubature formulas valid in the **Zernike** moment computation, we proposed a new condition in the **Zernike** moment computing.

The new condition

$$x_i^2 + y_j^2 \leq 1 - \gamma,$$

where

$$\gamma = \frac{\Delta x + \Delta y}{4} + \epsilon$$

in our case, was employed as an alternative solution. The results show that, for all four multi-dimensional formulas, the approximation errors go up quickly in the early stage, then the ratios slow down. From the approximation error point of view, the multi-dimensional formulas under the new condition provide better results than the simplest one-dimensional formula does.

Though it is premature to say that changing the condition is a perfect solution to reduce approximation errors in the **Zernike** moment computing, careful selection of the multi-dimensional formula and modification on the condition on the choice of better points, i.e. $x_i^2 + y_j^2 \leq 1 - \gamma$, will indeed improve the performance of the **Zernike** moment computation significantly.

## Appendix

Following is a set of thirteen linear system equations

$$
\left\{
\begin{array}{l}
\sum_{i=1}^{13} = |\Omega| \\
\sum_{i=1}^{13} A_i x_i = I_\Omega x \\
\sum_{i=1}^{13} A_i y_i = I_\Omega y \\
\sum_{i=1}^{13} A_i x_i^2 = I_\Omega x^2 \\
\sum_{i=1}^{13} A_i x_i y_i = I_\Omega xy \\
\sum_{i=1}^{13} A_i y_i^2 = I_\Omega y^2 \\
\sum_{i=1}^{13} A_i x_i^3 = I_\Omega x^3 \\
\sum_{i=1}^{13} A_i x_i^2 y_i = I_\Omega x^2 y \\
\sum_{i=1}^{13} A_i x_i y_i^2 = I_\Omega x y^2 \\
\sum_{i=1}^{13} A_i y_i^3 = I_\Omega y^3 \\
\sum_{i=1}^{13} A_i x_i^4 = I_\Omega x^4 \\
\sum_{i=1}^{13} A_i x_i^3 y = I_\Omega x^3 y \\
\sum_{i=1}^{13} A_i x_i^2 y^2 = I_\Omega x^2 y^2 ,
\end{array}
\right.
\tag{3.67}
$$

which leads to

$$
\left\{
\begin{array}{lllllllllllllll}
A_1 & +A_2 & +A_3 & +A_4 & +A_5 & +A_6 & +A_7 & +A_8 & +A_9 & +A_{10} & +A_{11} & +A_{12} & +A_{13} & = & 4 \\
 & +.5A_2 & +.5A_3 & -.5A_4 & -.5A_5 & & +A_7 & +A_8 & +A_9 & & -A_{11} & -A_{12} & -A_{13} & = & 0 \\
 & +.5A_2 & -.5A_3 & -.5A_4 & +.5A_5 & +A_6 & +A_7 & & -A_9 & -A_{10} & -A_{11} & & +A_{13} & = & 0 \\
 & +.25A_2 & +.25A_3 & +.25A_4 & +.25A_5 & & +A_7 & +A_8 & +A_9 & & +A_{11} & +A_{12} & +A_{13} & = & 2/3 \\
 & +.25A_2 & -.25A_3 & +.25A_4 & -.25A_5 & & +A_7 & & -A_9 & & +A_{11} & & -A_{13} & = & 0 \\
 & +.25A_2 & +.25A_3 & +.25A_4 & +.25A_5 & +A_6 & +A_7 & & +A_9 & +A_{10} & +A_{11} & & +A_{13} & = & 2/3 \\
 & +.125A_2 & +.125A_3 & -.125A_4 & -.125A_5 & & +A_7 & +A_8 & +A_9 & & -A_{11} & -A_{12} & -A_{13} & = & 0 \\
 & +.125A_2 & -.125A_3 & -.125A_4 & +.125A_5 & & +A_7 & & -A_9 & & -A_{11} & & +A_{13} & = & 0 \\
 & +.125A_2 & +.125A_3 & -.125A_4 & -.125A_5 & & +A_7 & & +A_9 & & -A_{11} & & -A_{13} & = & 0 \\
 & +.125A_2 & -.125A_3 & -.125A_4 & +.125A_5 & +A_6 & +A_7 & & -A_9 & -A_{10} & -A_{11} & & +A_{13} & = & 0 \\
 & +.0625A_2 & +.0625A_3 & +.0625A_4 & +.0625A_5 & & +A_7 & +A_8 & +A_9 & & +A_{11} & +A_{12} & +A_{13} & = & 2/5 \\
 & +.0625A_2 & -.0625A_3 & +.0625A_4 & -.0625A_5 & & +A_7 & & -A_9 & & +A_{11} & & -A_{13} & = & 0 \\
 & +.0625A_2 & +.0625A_3 & +.0625A_4 & +.0625A_5 & & +A_7 & & +A_9 & & +A_{11} & & +A_{13} & = & 4/9.
\end{array}
\right.
\tag{3.68}
$$

Solving these equations gives us the formula in (3.57) used in Chapter 3.

# Chapter 4

# Image Reconstruction from Moments

In Chapter 3, the problems of accuracy and efficiency in moment computing have been studied. In this chapter, we want to verify how much information is contained in moments. This issue can be addressed by analyzing the reconstruction power of the moments.

A problem which is raised here can be stated as follows: if only a finite set of moments of an image are given, how well can we reconstruct the image? We start the investigation by discussing the inverse moment problem.

## 4.1 Inverse Moment Problem

Consider the characteristic function[55] for the image function $f(x,y)$:

$$F(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\, e^{j(ux+vy)}\, dxdy. \tag{4.1}$$

Provided that $f(x,y)$ is piecewise continuous and the integration limits are finite, $F(u,v)$ is a continuous function and may be expanded as a power series in $u$ and $v$. Therefore,

$$F(u,v) \;\;=\;\; \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y) \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} \frac{(jux)^k}{k!}\,\frac{(jvy)^l}{l!}\, dxdy$$

$$\begin{aligned}
&= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\sum_{k=0}^{\infty}\frac{(ju)^k}{k!}\sum_{l=0}^{\infty}\frac{(jv)^l}{l!}\,x^k\,y^l\,f(x,y)\,dxdy \\
&= \sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\frac{j^{k+l}}{k!\,l!}\,u^k\,v^l\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\,x^k\,y^l\,dxdy \\
F(u,v) &= \sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\frac{j^{k+l}}{k!\,l!}\,u^k\,v^l\,M_{kl}, \qquad (4.2)
\end{aligned}$$

where the interchange of order of summation and integration is permissible, and the moment $M_{kl}$ is the geometric moment of order $(k+l)$ of the image function $f(x,y)$

$$M_{kl} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\,x^k\,y^l\,dxdy.$$

We see from (4.2) that the moment $M_{kl}$ is the expansion coefficient to the $u^k v^l$ term in the power series expansion of the characteristic function of the image function $f(x,y)$.

Then, we consider the inverse form of the characteristic function $F(u,v)$. From (4.2) and the two-dimensional inversion formula for **Fourier** transforms, it follows that

$$\begin{aligned}
f(x,y) &= \frac{1}{4\pi^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)\,e^{-j(ux+vy)}\,dudv \\
f(x,y) &= \frac{1}{4\pi^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\frac{j^{k+l}}{k!\,l!}\,u^k\,v^l\,M_{kl}\,e^{-j(ux+vy)}\,dudv. \qquad (4.3)
\end{aligned}$$

However, the order of summation and the integration in (4.3) cannot be interchanged. Thus we conclude that the power series expansion for $F(u,v)$ cannot be integrated term by term. Particularly, if only a finite set of moments is given, we cannot use a truncated series in (4.3) to learn about the original image function $f(x,y)$.

The difficulty encountered in (4.3) could have been solved if the basis set $\{u^k v^l\}$ were orthogonal. Unfortunately, with the **Weierstrass** approximation theorem[17], the basis set $\{u^k v^l\}$, while complete, is not orthogonal.

To solve this problem, we need a set of basis functions which are orthogonal over a finite interval. Based on this requirement, the **Legendre** polynomials would be the appropriate set.

## 4.2 Method of Legendre Moments

### 4.2.1 Theory of Image Reconstruction from Legendre Moments

As mentioned in Chapter 2, the **Legendre** polynomials $\{P_m(x)\}$[17] are a complete orthogonal basis set on the interval [-1, 1]:

$$\int_{-1}^{+1} P_m(x)\, P_n(x)\, dx = \frac{2}{2m+1}\, \delta_{mn}. \tag{4.4}$$

By the orthogonality principle, and considering that $f(x,y)$ is piecewise continuous over the image plane, we can write the image function $f(x,y)$ as an infinite series expansion:

$$f(x,y) = \sum_{m=0}^{\infty} \sum_{n=0}^{m} \lambda_{m-n,n}\, P_{m-n}(x)\, P_n(y), \tag{4.5}$$

where the **Legendre** moment of $f(x,y)$ with order $(m+n)$ is defined by

$$\lambda_{mn} = \frac{(2m+1)(2n+1)}{4} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} P_m(x)\, P_n(y)\, f(x,y)\, dxdy. \tag{4.6}$$

However, in practice, one has to truncate infinite series in (4.5). If only **Legendre** moments of order $\leq M_{max}$ are given, the function $f(x,y)$ can be approximated by a truncated series:

$$f(x,y) \simeq f_{M_{max}}(x,y) = \sum_{m=0}^{M_{max}} \sum_{n=0}^{m} \lambda_{m-n,n}\, P_{m-n}(x)\, P_n(y). \tag{4.7}$$

Furthermore, $\lambda_{m-n,n}$'s must be replaced by their approximations given by (3.19), yielding the following reconstruction scheme

$$\widehat{f}_{M_{max}}(x,y) = \sum_{m=0}^{M_{max}} \sum_{n=0}^{m} \widehat{\lambda}_{m-n,n}\, P_{m-n}(x)\, P_n(y). \tag{4.8}$$

This is actually the basic equation used in the image reconstruction via the **Legendre** moments. It is important to note that when the given order $M_{max}$ is increased, the previously determined $\widehat{\lambda}_{m-n,n}$'s do not change.

### 4.2.2 Reconstruction Error Analysis

To measure the error between the original image and its reconstructed version, the following formula is employed

$$Error(\widehat{f}_{M_{max}}) = \int_{-1}^{1}\int_{-1}^{1}[\widehat{f}_{M_{max}}(x,y) - f(x,y)]^2 dxdy, \tag{4.9}$$

where $M_{max}$ is the highest moment order involved in reconstruction, and $\widehat{f}(x,y)$ represents the reconstructed image from $f(x,y)$.

Since

$$\widehat{f}_{M_{max}}(x,y) = \sum_{m=0}^{M_{max}}\sum_{n=0}^{m} P_{m-n,n}(x)\,P_n(y)\,\widehat{\lambda}_{m-n,n}$$

and

$$f(x,y) = \sum_{m=0}^{\infty}\sum_{n=0}^{m} P_{m-n,n}(x)\,P_n(y)\,\lambda_{m-n,n},$$

therefore

$$\begin{aligned}\widehat{f}_{M_{max}}(x,y) - f(x,y) &= \sum_{m=0}^{M_{max}}\sum_{n=0}^{m} P_{m-n,n}(x)\,P_n(y)\,\widehat{\lambda}_{m-n,n} \\ &\quad - \sum_{m=0}^{\infty}\sum_{n=0}^{m} P_{m-n,n}(x)\,P_n(y)\,\lambda_{m-n,n} \\ &= \sum_{m=0}^{M_{max}}\sum_{n=0}^{m} P_{m-n,n}(x)\,P_n(y)\,[\widehat{\lambda}_{m-n,n} - \lambda_{m-n,n}] \\ &\quad - \sum_{m=M_{max}+1}^{\infty}\sum_{n=0}^{m} P_{m-n,n}(x)\,P_n(y)\,\lambda_{m-n,n}. \end{aligned} \tag{4.10}$$

Then, we have

$$\begin{aligned}Error(\widehat{f}_{M_{max}}) &= \int_{-1}^{1}\int_{-1}^{1}[\widehat{f}_{M_{max}}(x,y) - f(x,y)]^2 dxdy \\ &= \int_{-1}^{1}\int_{-1}^{1}[\sum_{m=0}^{M_{max}}\sum_{n=0}^{m} P_{m-n,n}(x)\,P_n(y)\,(\widehat{\lambda}_{m-n,n} - \lambda_{m-n,n})]^2\, dxdy\end{aligned}$$

$$-2\int_{-1}^{1}\int_{-1}^{1}[\sum_{m=0}^{M_{max}}\sum_{n=0}^{m} P_{m-n,n}(x)\,P_n(y)\,(\widehat{\lambda}_{m-n,n}-\lambda_{m-n,n})]$$
$$[\sum_{m=M_{max}+1}^{\infty}\sum_{n=0}^{m} P_{m-n,n}(x)\,P_n(y)\,\lambda_{m-n,n}]\,dxdy$$
$$+\int_{-1}^{1}\int_{-1}^{1}[\sum_{m=M_{max}+1}^{\infty}\sum_{n=0}^{m} P_{m-n,n}(x)\,P_n(y)\,\lambda_{m-n,n}]^2\,dxdy. \quad (4.11)$$

Since the second term in (4.11) is zero and **Legendre** polynomials $P_m(x)$ and $P_n(y)$ are orthogonal, applying (2.30) to (4.11), we have

$$\begin{aligned} Error(\widehat{f}_{M_{max}}) &= 4\sum_{m=0}^{M_{max}}\sum_{n=0}^{m}\frac{1}{2(m-n)+1}\,\frac{1}{2n+1}\,[\widehat{\lambda}_{m-n,n}-\lambda_{m-n,n}]^2 \\ &\quad +4\sum_{m=M_{max}+1}^{\infty}\sum_{n=0}^{m}\frac{1}{2(m-n)+1}\,\frac{1}{2n+1}\,\lambda^2_{m-n,n}. \end{aligned} \quad (4.12)$$

As shown in (4.12), the reconstruction error $Error(\widehat{f}_{M_{max}})$ consists of two parts. The first part comes from the discrete approximation of the true moment $(\lambda_{m,n})$ while the second part is a result of using a finite number of moments.

With the new techniques introduced in the previous chapter, we can reduce the discrete approximation error to a tolerable low level. Based on these new techniques, the experimental results of image reconstruction via **Legendre** moments, which will be presented in the following section, indicate that when the maximum given order $M_{max}$ reaches a certain value, $\widehat{f}_{M_{max}}(x,y)$ can be very close to the original image function $f(x,y)$.

### 4.2.3 Experimental Results

The proposed approach was implemented in the C language and tested on a 25MHz 386 computer. In the experiments, a set of five Chinese characters, shown in Figure 4.1, is used as the test images. Each image consists of $24 \times 24$ pixels and the range of graylevels for each pixel is 32 . All characters have the gray level 11 and the background has the value 21.

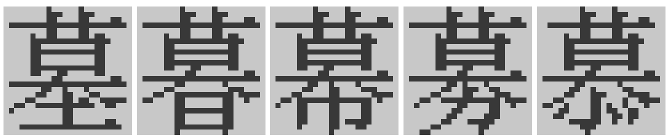


Figure 4.1: Five original Chinese characters used in image reconstruction via **Legendre** moments. From left to right are $C_1$, $C_2$, $C_3$, $C_4$, and $C_5$.

The reason we use these five Chinese characters is that they are very similar to each other. Actually, among more than 50,000 Chinese characters, one cannot find another set of five(or even set of three or four) in which the individual characters are so similar to each other. Therefore, it seems that if these five characters can be recognized successfully, the method can be applied to all the Chinese characters with confidence.

The normalized mean square error between the original image $f(x,y)$ and the reconstructed image $\widehat{f}_{M_{max}}(x,y)$ is defined by

$$\overline{e^2_{M_{max}}} = \frac{Error(\widehat{f}_{M_{max}})}{\int\int[f(x,y)]^2\,dxdy} = \frac{\int\int[\widehat{f}_{M_{max}}(x,y) - f(x,y)]^2\,dxdy}{\int\int[f(x,y)]^2\,dxdy}, \qquad -1 \le x,y \le 1, \tag{4.13}$$

which is considered as a measure of the image reconstruction ability of the moments and adopted here.

The *alternative extended Simpson's rule* with order $N = 23$ is applied to compute the **Legendre** moments in this experiment. Table 4.1 and Figure 4.3 show the $\overline{e^2_{M_{max}}}$ values from the reconstructed Chinese characters from order 2 up to order 56. It should be noted that the $\overline{e^2_{M_{max}}}$ decreases monotonically in the cases of all five characters.

Figure 4.2 shows the five original Chinese characters and their reconstructed patterns. The first column illustrates five original characters. The second column to the ninth column display the reconstructed patterns of all characters in the first column with order 28, 32, 36, 40, 44, 48, 52 and 56, respectively.

| Order | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
|---|---|---|---|---|---|
| 2 | 0.047615 | 0.045822 | 0.046409 | 0.043936 | 0.044973 |
| 4 | 0.046324 | 0.045556 | 0.045387 | 0.043429 | 0.043358 |
| 6 | 0.045428 | 0.043609 | 0.044421 | 0.041650 | 0.042230 |
| 8 | 0.042815 | 0.040956 | 0.043409 | 0.041143 | 0.040772 |
| 10 | 0.040362 | 0.039399 | 0.041322 | 0.038537 | 0.039932 |
| 12 | 0.038164 | 0.037170 | 0.037279 | 0.036748 | 0.037710 |
| 14 | 0.032964 | 0.034942 | 0.034077 | 0.033165 | 0.034441 |
| 16 | 0.030828 | 0.032760 | 0.030799 | 0.030356 | 0.032472 |
| 18 | 0.027089 | 0.029455 | 0.028143 | 0.026889 | 0.029127 |
| 20 | 0.024840 | 0.025035 | 0.025198 | 0.023493 | 0.026409 |
| 22 | 0.021336 | 0.021264 | 0.022372 | 0.021577 | 0.023423 |
| 24 | 0.017141 | 0.019605 | 0.019870 | 0.019955 | 0.020928 |
| 26 | 0.014513 | 0.016060 | 0.017691 | 0.018034 | 0.018245 |
| 28 | 0.012146 | 0.012868 | 0.014819 | 0.015282 | 0.015386 |
| 30 | 0.010568 | 0.010343 | 0.012224 | 0.012753 | 0.012617 |
| 32 | 0.008775 | 0.008540 | 0.009367 | 0.010563 | 0.010127 |
| 34 | 0.007346 | 0.007377 | 0.007456 | 0.008500 | 0.008573 |
| 36 | 0.006547 | 0.006526 | 0.006485 | 0.007162 | 0.007298 |
| 38 | 0.005348 | 0.005645 | 0.005668 | 0.006265 | 0.006336 |
| 40 | 0.004564 | 0.004769 | 0.004799 | 0.005378 | 0.004980 |
| 42 | 0.003996 | 0.004293 | 0.004219 | 0.004653 | 0.004322 |
| 44 | 0.003504 | 0.003734 | 0.003576 | 0.004111 | 0.003786 |
| 46 | 0.003217 | 0.003181 | 0.003165 | 0.003584 | 0.003199 |
| 48 | 0.003048 | 0.002869 | 0.002811 | 0.003041 | 0.002865 |
| 50 | 0.002607 | 0.002613 | 0.002728 | 0.002691 | 0.002647 |
| 52 | 0.002408 | 0.002321 | 0.002394 | 0.002605 | 0.002384 |
| 54 | 0.002408 | 0.002556 | 0.002451 | 0.002332 | 0.002419 |
| 56 | 0.002377 | 0.002602 | 0.002415 | 0.002271 | 0.002176 |

Table 4.1: The values of normalized reconstruction errors for the five reconstructed Chinese characters.

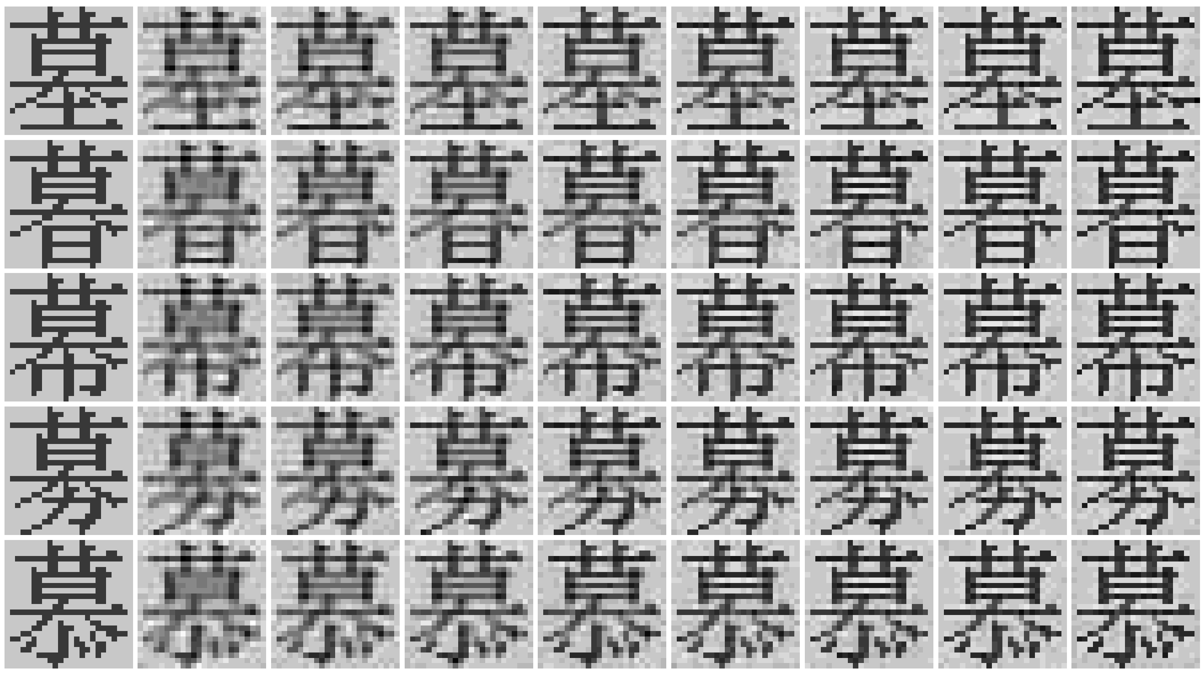


Figure 4.2: Five Chinese characters and their reconstructed patterns via **Legendre** moments.

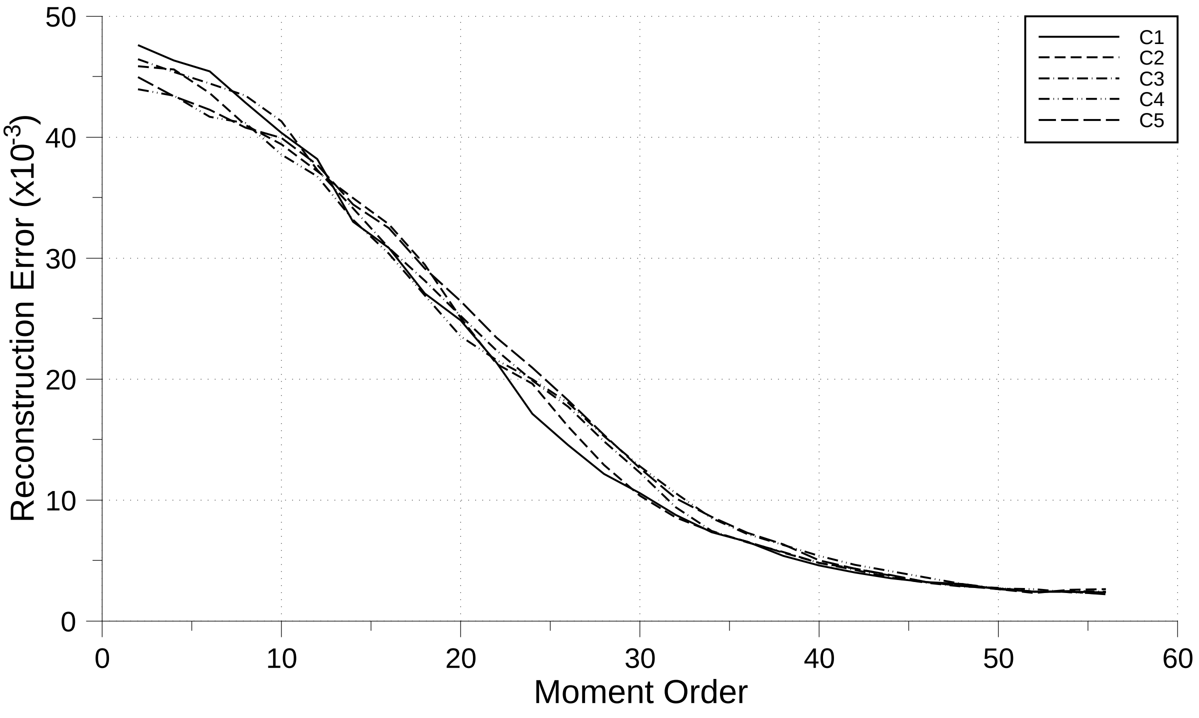


Figure 4.3: Normalized reconstruction errors for the five reconstructed Chinese characters.

Clearly, the numerical results shown in Table 4.1 and Figure 4.3 are concordant with the visual results presented in Figure 4.2.

## 4.3 Method of Zernike Moments

### 4.3.1 Theory of Image Reconstruction from Zernike Moments

As discussed in Chapter 2, the **Zernike** polynomials

$$V_{nm}(x,y) = V_{nm}(\rho sin\theta, \rho cos\theta) = R_{nm}(\rho)\, exp(jm\theta), \tag{4.14}$$

where the Radial polynomial $R_{nm}(\rho)$ is defined as

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s!\,(\frac{n+|m|}{2}-s)!\,(\frac{n-|m|}{2}-s)!}\, \rho^{n-2s}, \tag{4.15}$$

are a complete set of complex-valued functions orthogonal on the unit disk $x^2+y^2 \leq 1$:

$$\iint_{x^2+y^2\leq 1} [V_{nm}(x,y)]^*\, V_{pq}(x,y)\, dxdy = \frac{\pi}{n+1}\, \delta_{np}\, \delta_{mq}. \tag{4.16}$$

Subject to the discussion of orthogonal functions for the **Legendre** moments, the image function $f(x,y)$ can be expanded in terms of the **Zernike** polynomials over the unit disk as

$$f(x,y) = \sum_{n}^{\infty} \sum_{m} A_{nm}\, V_{nm}(x,y), \tag{4.17}$$

where $m$ takes on positive and negative integers subject to the conditions $n-|m| =$ even, and $|m| < n$.

We rewrite the definition of **Zernike** moments here for convenience:

$$A_{nm} = \frac{n+1}{\pi} \int\int_{x^2+y^2\leq 1} f(x,y)\, V_{nm}^*(\rho,\theta) dxdy. \tag{4.18}$$

If terms only up to the maximum **Zernike** moment $N_{max}$ are taken, then the truncated expansion is the approximation to $f(x,y)$:

$$f(x,y) \simeq \widehat{f}_{N_{max}}(x,y) = \sum_{n=0}^{N_{max}} \sum_{m} \widehat{A}_{nm}\, V_{nm}(x,y), \tag{4.19}$$

where $\widehat{A}_{nm}$ and $\widehat{f}_{N_{max}}(x,y)$ are the **Zernike** moment numerically computed from $f(x,y)$ and the reconstructed image from $f(x,y)$ with the maximum **Zernike** moment $N_{max}$,, while $m$ is subject to the conditions $n-|m| =$ even, and $|m|<n$.

Note that $V^*_{nm}(\rho,\theta) = V_{n,-m}(\rho,\theta)$, (4.19) can be expanded as

$$\begin{aligned}
\widehat{f}_{N_{max}}(x,y) &= \sum_{n=0}^{N_{max}} \sum_{m} \widehat{A}_{nm}\, V_{nm}(\rho,\theta) \\
&= \sum_{n=0}^{N_{max}} \sum_{m<0} \widehat{A}_{nm}\, V_{nm}(\rho,\theta) + \sum_{n=0}^{N_{max}} \sum_{m\geq 0} \widehat{A}_{nm}\, V_{nm}(\rho,\theta) \\
&= \sum_{n=0}^{N_{max}} \sum_{m>0} \widehat{A}_{n,-m}\, V_{n,-m}(\rho,\theta) \\
&\quad + \sum_{n=0}^{N_{max}} \sum_{m>0} \widehat{A}_{nm}\, V_{nm}(\rho,\theta) + \widehat{A}_{n0}\, V_{n0}(\rho,\theta) \\
\widehat{f}_{N_{max}}(x,y) &= \sum_{n=0}^{N_{max}} \sum_{m>0} [\widehat{A}^*_{nm}\, V^*_{nm}(\rho,\theta) + \widehat{A}_{nm}\, V_{nm}(\rho,\theta)] \\
&\quad + \widehat{A}_{n0}\, V_{n0}(\rho,\theta),
\end{aligned} \tag{4.20}$$

considering that

$$V_{nm}(\rho,\theta) = R_{nm}(\rho)\,(cos(m\theta) + j\, sin(m\theta))$$

and

$$V^*_{nm}(\rho,\theta) = R_{nm}(\rho)\,(cos(m\theta) - j\, sin(m\theta)).$$

Then (4.20) becomes

$$\begin{aligned}
\widehat{f}_{N_{max}}(x,y) &= \sum_{n=0}^{N_{max}} \sum_{m>0} \{(Re[\widehat{A}_{nm}] - j\, Im[\widehat{A}_{nm}])\, R_{nm}(\rho)\,(cos(m\theta) - j\, sin(m\theta)) \\
&\quad + (Re[\widehat{A}_{nm}] + j\, Im[\widehat{A}_{nm}])\, R_{nm}(\rho)\,(cos(m\theta) + j\, sin(m\theta))\} \\
&\quad + (Re[\widehat{A}_{n0}] + j\, Im[\widehat{A}_{n0}])\, R_{n0}(\rho) \\
&= \sum_{n=0}^{N_{max}} \sum_{m>0} R_{nm}(\rho)\, \{(Re[\widehat{A}_{nm}] - j\, Im[\widehat{A}_{nm}])\,(cos(m\theta) - j\, sin(m\theta)) \\
&\quad + (Re[\widehat{A}_{nm}] + j\, Im[\widehat{A}_{nm}])\,(cos(m\theta) + j\, sin(m\theta))\} \\
&\quad + (Re[\widehat{A}_{n0}] + j\, Im[\widehat{A}_{n0}])\, R_{n0}(\rho)
\end{aligned}$$

$$\begin{aligned}
\widehat{f}_{N_{max}}(x,y) &= \sum_{n=0}^{N_{max}} \sum_{m>0} 2\, R_{nm}(\rho)\,(Re[\widehat{A}_{nm}]\, cos(m\theta) - Im[\widehat{A}_{nm}]\, sin(m\theta)) \\
&+ (Re[\widehat{A}_{n0}] + j\, Im[\widehat{A}_{n0}])\, R_{n0}(\rho). \qquad (4.21)
\end{aligned}$$

The formula (4.21) is the basic equation employed in image reconstruction via the **Zernike** moments.

## 4.3.2 Reconstruction Error Analysis

In this section, a similar definition as in(4.9) is adopted to measure the error between the original image and its reconstructed version from **Zernike** moments.

$$Error(\widehat{f}_{N_{max}}) = \int\int_{x^2+y^2\leq 1} |\widehat{f}_{N_{max}}(x,y) - f(x,y)|^2 dxdy. \qquad (4.22)$$

Since

$$\begin{aligned}
\widehat{f}_{N_{max}}(x,y) - f(x,y) &= \sum_{n=0}^{N_{max}} \sum_{m} \widehat{A}_{nm} V_{nm}(x,y) \\
&\quad -[\sum_{n=0}^{N_{max}} \sum_{m} A_{nm} V_{nm}(x,y) + \sum_{n=N_{max+1}}^{\infty} \sum_{m} A_{nm} V_{nm}(x,y)] \\
&= \sum_{n=0}^{N_{max}} \sum_{m} V_{nm}(x,y)[\widehat{A}_{nm} - A_{nm}] \\
&\quad - \sum_{n=N_{max+1}}^{\infty} \sum_{m} A_{nm} V_{nm}(x,y), \qquad (4.23)
\end{aligned}$$

therefore it follows that

$$\begin{aligned}
Error(\widehat{f}_{N_{max}}) &= \int\int_{x^2+y^2\leq 1} |\widehat{f}_{N_{max}}(x,y) - f(x,y)|^2 dxdy \\
&= \int\int_{x^2+y^2\leq 1} \sum_{n=0}^{N_{max}} \sum_{m} |V_{nm}(x,y)|^2 |\widehat{A}_{nm} - A_{nm}|^2 dxdy \\
&\quad -2\int\int_{x^2+y^2\leq 1} \sum_{n=0}^{N_{max}} \sum_{m} V_{nm}(x,y)[\widehat{A}_{nm} - A_{nm}]\, dxdy \\
&\quad \int\int_{x^2+y^2\leq 1} \sum_{N_{max+1}}^{\infty} \sum_{m} A_{nm} V_{nm}(x,y)\, dxdy \\
&\quad + \int\int_{x^2+y^2\leq 1} \sum_{n=N_{max}}^{\infty} \sum_{m} |V_{nm}^{(}x,y)|^2 |A_{nm}|^2\, dxdy. \qquad (4.24)
\end{aligned}$$

It is clear that the second term on the right side of (4.24) is zero. So, the reconstruction error for **Zernike** moments consists of two terms:

$$\begin{aligned} Error(\widehat{f}_{N_{max}}) &= \int\int_{x^2+y^2\leq 1} \sum_{n=0}^{N_{max}} \sum_{m} |V_{nm}(x,y)|^2 |\widehat{A}_{nm} - A_{nm}|^2 \, dxdy \\ &+ \int\int_{x^2+y^2\leq 1} \sum_{n=N_{max}+1}^{\infty} \sum_{m} |V_{nm}(x,y)|^2 |A_{nm}|^2 \, dxdy. \quad (4.25) \end{aligned}$$

By recalling (4.16), we obtain:

$$Error(\widehat{f}_{N_{max}}) = \pi \sum_{n=0}^{N_{max}} \sum_{m} \frac{|\widehat{A}_{nm} - A_{nm}|^2}{n+1} + \pi \sum_{n=N_{max}}^{\infty} \sum_{m} \frac{|A_{nm}|^2}{n+1}, \qquad (4.26)$$

where the first part is from the approximation error in **Zernike** moment computing and the second term is due to truncating the higher order moments in image reconstruction.

## 4.3.3 Experimental Results

The same set of five Chinese characters used in the case of **Legendre** moments is employed in this section as the test images as well. Figure 4.4 illustrates the five characters on unit disks.

### Traditional Zernike Moment Method

The traditional **Zernike** moment method, which defines the **Zernike** moments

$$A_{nm} = \frac{n+1}{\pi} \int\int f(x,y) \, V_{nm}^{*}(\rho,\theta) dxdy$$

on the unit disk

$$x^2 + y^2 \leq 1,$$

is implemented in the image reconstruction. The simplest 1-dimensional formula, along with the two different types of 5-dimensional formulas, and the 13-dimensional formula with two different sets of coefficients are employed in the experiment.

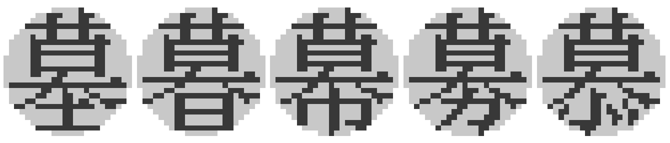


Figure 4.4: Five original Chinese characters used in image reconstruction via **Zernike** moments. From left to right are $C_1$, $C_2$, $C_3$, $C_4$, and $C_5$.

The normalized mean square errors

$$\overline{e^2} = \frac{\int\int |f(x,y) - \widehat{f}(x,y)|^2 dxdy}{\int\int [f(x,y)]^2 dxdy}, \qquad x^2 + y^2 \leq 1, \tag{4.27}$$

are adopted here to measure the qualities of the reconstructed images via the **Zernike** moments.

| Order | 1-D | 5-D(I) | 5-D(II) | 13-D(I) | 13-D(II) |
|---|---|---|---|---|---|
| 2 | 0.055213 | 0.054943 | 0.056078 | 0.054943 | 0.054943 |
| 4 | 0.055961 | 0.055864 | 0.057879 | 0.055864 | 0.055864 |
| 6 | 0.054160 | 0.054049 | 0.057373 | 0.053973 | 0.053973 |
| 8 | 0.050504 | 0.049167 | 0.056113 | 0.049230 | 0.049216 |
| 10 | 0.048752 | 0.046411 | 0.051591 | 0.046480 | 0.046633 |
| 12 | 0.046750 | 0.045407 | 0.047394 | 0.045691 | 0.045975 |
| 14 | 0.051584 | 0.047041 | 0.049077 | 0.048419 | 0.051099 |
| 16 | 0.065199 | 0.052803 | 0.053426 | 0.056944 | 0.064922 |
| 18 | 0.073065 | 0.060081 | 0.051446 | 0.070122 | 0.096181 |
| 20 | 0.085419 | 0.103251 | 0.052595 | 0.132682 | 0.224127 |
| 22 | 0.101229 | 0.268038 | 0.066272 | 0.386192 | 0.780797 |
| 24 | 0.125072 | 0.773207 | 0.089699 | 1.413850 | 2.847505 |
| 26 | 0.238904 | 2.473051 | 0.236481 | | |
| 28 | 0.395097 | | 0.416779 | | |
| 30 | 0.482546 | | 0.725626 | | |
| 32 | 0.584135 | | 0.669215 | | |
| 34 | 0.939857 | | 0.813116 | | |
| 36 | 1.307849 | | 1.167633 | | |
| 38 | 1.436626 | | 0.960680 | | |
| 40 | 1.590319 | | 1.497801 | | |

Table 4.2: Values of the normalized mean square errors from appling five different formulas to character $C_1$.

The Chinese character $C_1$ is used as the test image for all five different formulas. Table 4.2 and Figure 4.6 show the $\overline{e^2}$ values for different formulas when the

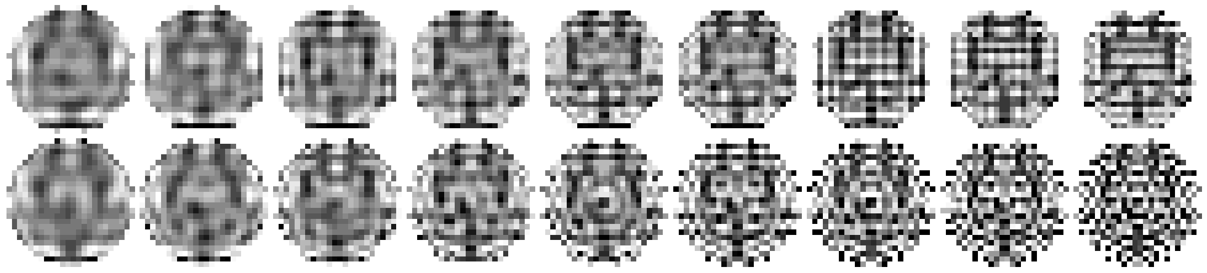


Figure 4.5: The Chinese character $C_1$ and its reconstructed patterns via **Zernike** moments.

reconstruction orders go up.

Only the simplest 1-dimensional formula and the 5-dimensional formula II, which is shown in Figure 3.4, provide relatively lower computing errors. The remaining three formulas certainly are not good candidates for image reconstruction because of the excessive computing errors.

Figure 4.5 illustrates the reconstructed images of $C_1$. The first row and the second row display the results from using the 1-dimensional formula and the 5-dimensional formula II, respectively. The patterns in the first column are reconstructed from order 14, then from left to right, they are the reconstructed images from order 16, 18, 20, 22, 24, 26, 28, and 30, respectively.

When the order is 26, Table 4.2 indicates that in terms of the normalized mean square error, the reconstructed image from the 5-dimensional formula II has lower error than that of 1-dimensional formula. However, the visual results are contrary. The reason is that the normalized mean square error treats all pixels equally, while the individual key pixels which contain more features affect the visual results more significantly.

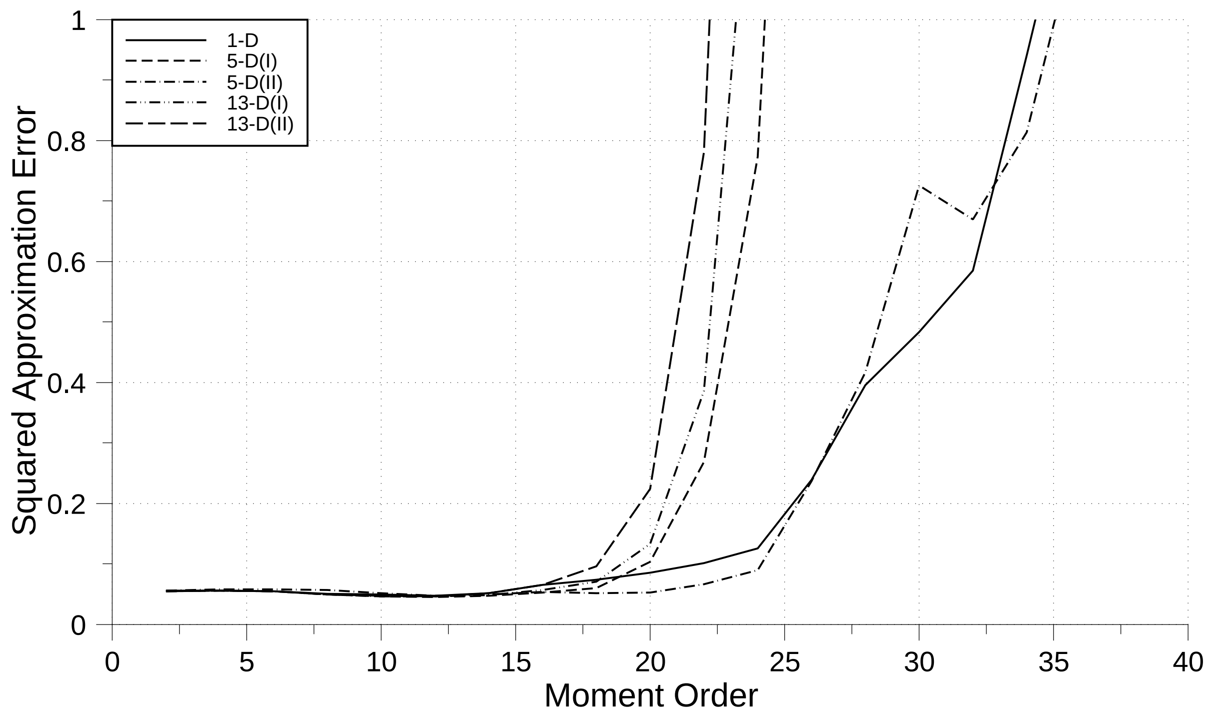


Figure 4.6: The normalized mean square errors from appling five different formulas to character $C_1$.

**Modified Zernike Moment Method**

We introduced a modified version of the **Zernike** moments in Chapter 3. We redefined the **Zernike** moments as

$$\widehat{A}_{nm} = \frac{n+1}{\pi} \sum_{x_i} \sum_{y_j} h_{A_{nm}}(x_i, y_j)\, f(x_i, y_j), \qquad x_i^2 + y_j^2 \leq 1 - \gamma, \tag{4.28}$$

where

$$h_{A_{nm}}(x_i, y_j) = \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} \int_{y_j - \frac{\Delta y}{2}}^{y_j + \frac{\Delta y}{2}} V_{nm}^*(\rho, \theta) dx dy, \tag{4.29}$$

and $\gamma$ is an adjustable factor. For example, in this research, we let

$$\gamma = \frac{\Delta x + \Delta y}{4} + \epsilon, \tag{4.30}$$

where $\epsilon$ is an arbitrary small number.

The main reason to adopt (4.30) is that when we use the multi-dimensional formulas to increase the approximate accuracy, we want to reduce the number of nodes which fall outside the unit disk. The price of adopting the new version is that the geometrical errors will become larger.

For the sake of convenient comparison, the same Chinese character $C_1$ is employed as the test image, and the normalized mean square error defined in (4.27) is adjusted to

$$\overline{e^2} = \frac{\int\int |f(x,y) - \widehat{f}(x,y)|^2 dx dy}{\int\int [f(x,y)]^2 dx dy}, \qquad x^2 + y^2 \leq 1 - \gamma, \tag{4.31}$$

to measure this new method.

Table 4.3 and Figure 4.8 show the $\overline{e^2}$ values of all five different formulas when the orders of the reconstructed images go up.

Table 4.3 and Figure 4.8 indicate that all five different formulas perform better than they did in the cases of traditional **Zernike** moments. Specifically, four multi-dimensional formulas produce lower computing errors than the simplest 1-dimensional does. Among the multi-dimensional formulas, the 5-dimensional formula

| Order | 1-D | 5-D(I) | 5-D(II) | 13-D(I) | 13-D(II) |
|---|---|---|---|---|---|
| 2 | 0.084803 | 0.084788 | 0.084140 | 0.084788 | 0.084788 |
| 4 | 0.086955 | 0.086269 | 0.085942 | 0.086269 | 0.086269 |
| 6 | 0.066534 | 0.065864 | 0.066526 | 0.065864 | 0.065871 |
| 8 | 0.056585 | 0.056943 | 0.057341 | 0.056943 | 0.056803 |
| 10 | 0.054370 | 0.053879 | 0.056148 | 0.054191 | 0.054339 |
| 12 | 0.050674 | 0.051415 | 0.053731 | 0.051345 | 0.051626 |
| 14 | 0.047111 | 0.047376 | 0.049006 | 0.047446 | 0.047712 |
| 16 | 0.038940 | 0.038674 | 0.041326 | 0.039002 | 0.039314 |
| 18 | 0.032694 | 0.032203 | 0.034136 | 0.032398 | 0.033232 |
| 20 | 0.030620 | 0.028749 | 0.030869 | 0.028476 | 0.029583 |
| 22 | 0.030019 | 0.024936 | 0.026573 | 0.024928 | 0.026721 |
| 24 | 0.024429 | 0.019283 | 0.022129 | 0.020561 | 0.023906 |
| 26 | 0.030760 | 0.019953 | 0.021957 | 0.020725 | 0.025505 |
| 28 | 0.039758 | 0.019696 | 0.018082 | 0.021513 | 0.028803 |
| 30 | 0.039696 | 0.016881 | 0.017466 | 0.018160 | 0.025365 |
| 32 | 0.058339 | 0.022152 | 0.018620 | 0.020936 | 0.028304 |
| 34 | 0.106035 | 0.038370 | 0.024998 | 0.032265 | 0.044382 |
| 36 | 0.123119 | 0.041910 | 0.026487 | 0.033606 | 0.048897 |
| 38 | 0.169365 | 0.061747 | 0.042706 | 0.047283 | 0.069645 |
| 40 | 0.329895 | 0.120624 | 0.055665 | 0.092343 | 0.138519 |
| 42 | 0.433146 | 0.144304 | 0.059244 | 0.160647 | 0.292140 |
| 44 | 0.633271 | 0.230363 | 0.072008 | 0.245778 | 0.454737 |
| 46 | 1.097263 | 0.452476 | 0.137053 | 0.486214 | 0.896195 |
| 48 | 1.398472 | 0.657138 | 0.206285 | 1.231595 | 1.659259 |

Table 4.3: Values of the normalized reconstruction errors from the reconstructed five Chinese characters with the new proposed **Zernike** moment technique.

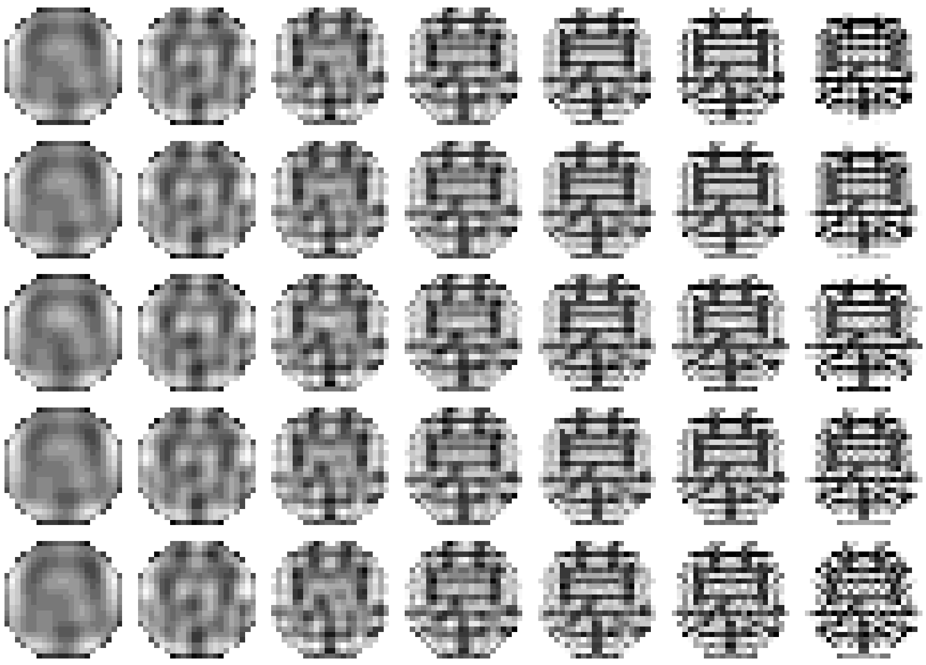


Figure 4.7: The Chinese character $C_1$ and its reconstructed patterns via the modified **Zernike** moments.

II which is shown in Figure 3.4, is superior to the other three formulas and is the best candidate for the image reconstruction under this specific situation.

The reason that the 5-dimensional formula II provides better result is that with the new condition for the **Zernike** moments, all nodes used in this formula fall inside the unit disk. For the 5-dimensional formula I and the 13-dimensional formulas, however, 16 and 68 nodes used in the computing fall outside the unit disk, respectively.

The reconstructed images from the Chinese character $C_1$ with five different formulas are shown in Figure 4.7. The first row shows the reconstructed patterns from the 1-dimensional formula, and the second, third, fourth, and fifth show those of 5-dimensional formula I, II, 13-dimensional formula I, and II, respectively. All images in the first column are reconstructed from order 10, then from left to right, are results from order 15, 20, 25, 30, 35, and 40.

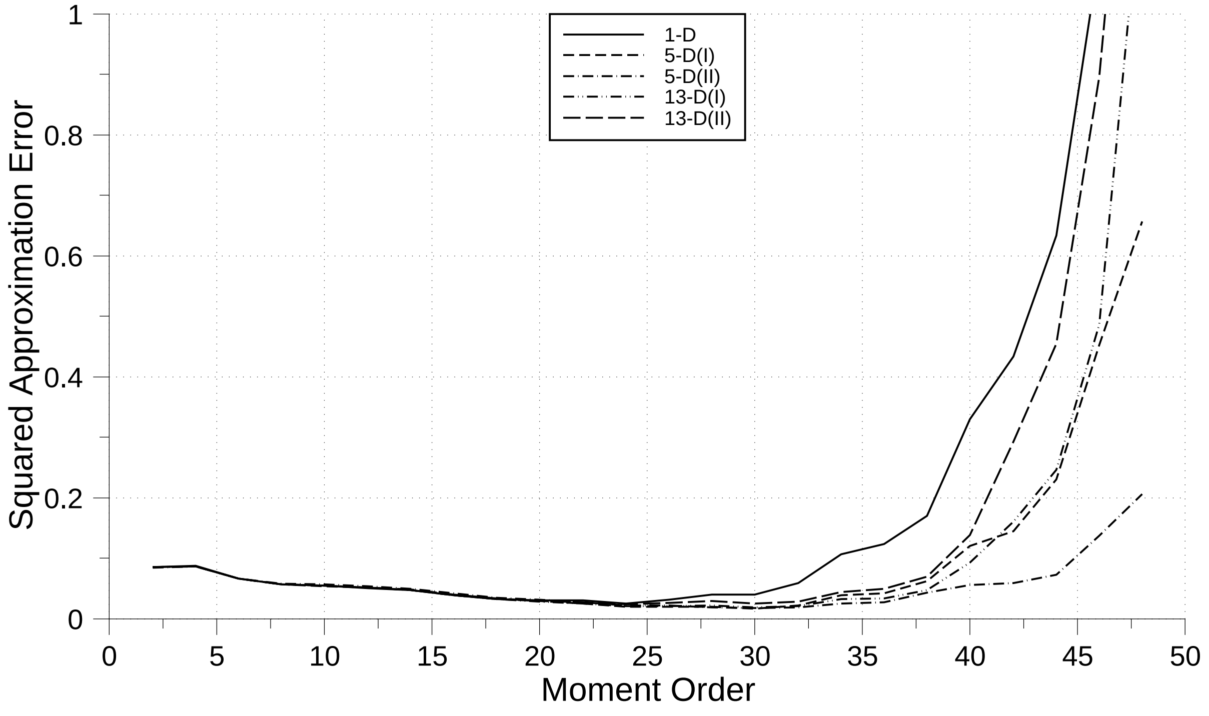


Figure 4.8: Normalized reconstruction errors from the reconstructed five Chinese characters via the new proposed **Zernike** moment technique.

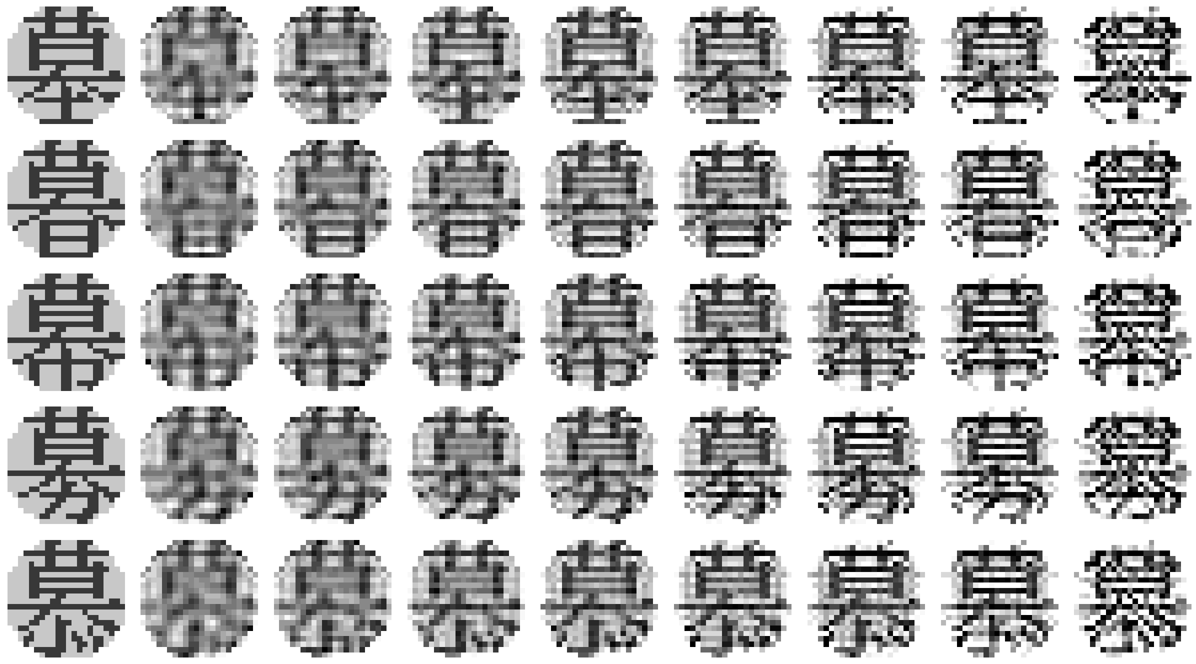


Figure 4.9: The five Chinese characters and their reconstructed patterns via the modified **Zernike** moments with 5-dimensional formula II.

By using the 5-dimensional formula II, we reconstructed all five Chinese characters with the **Zernike** moment order 20, 24, 28, 32, 36, 40, 44, and 48, respectively. Figure 4.9 shows the reconstructed images while Table 4.4 and Figure 4.8 list and illustrate the normalized mean square reconstruction errors of these patterns.

## 4.4 Conclusions

In this chapter, the image reconstructions via the **Legendre** moments and **Zernike** moments have been discussed.

### 4.4.1 Image Reconstruction via Legendre Moments

Since we have solved most of accuracy and efficiency problems related to the **Legendre** moment computing in Chapter 3, image reconstruction from the higher order **Legendre** moments results in a successful task.

| Order | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
|---|---|---|---|---|---|
| 20 | 0.030869 | 0.033030 | 0.032264 | 0.030916 | 0.032750 |
| 24 | 0.022129 | 0.026868 | 0.028001 | 0.028148 | 0.028903 |
| 28 | 0.018082 | 0.019077 | 0.024242 | 0.023891 | 0.020548 |
| 32 | 0.018620 | 0.018383 | 0.019420 | 0.020195 | 0.019606 |
| 36 | 0.026487 | 0.029725 | 0.033788 | 0.032982 | 0.031015 |
| 40 | 0.055665 | 0.058078 | 0.055500 | 0.057263 | 0.054554 |
| 44 | 0.072008 | 0.081936 | 0.079502 | 0.079267 | 0.080677 |
| 48 | 0.206285 | 0.196951 | 0.200966 | 0.214339 | 0.216396 |

Table 4.4: Values of the normalized reconstruction errors from the reconstructed five Chinese characters via the new proposed **Zernike** moment technique.

Five similar Chinese characters are used as the test images in the image reconstruction procedure. Numerical and visual results both show that the reconstructed images from the high order **Legendre** moments are very close to the original ones. When the order goes higher, the difference between the original image and its reconstructed pattern becomes smaller.

### 4.4.2 Image Reconstruction via Zernike Moments

In this chapter, we discussed the image reconstruction via the traditional **Zernike** moments and a proposed new **Zernike** moment method as well.

#### Traditional Zernike Moment Method

Five different cubature formulas are applied in the image reconstruction via traditional **Zernike** moment method. However, most of multi-dimensional formulas employed to increase the accuracy of the **Zernike** moment computing cannot provide the expected results in the image reconstruction procedure. The reason for the failure is that the pixels on the unit disk boundary may contain nodes falling out of the unit circle and which bring in excessive computing errors.

In this task, the simplest 1-dimensional formula provides relatively better reconstructed patterns than all the other multi-dimensional formulas do.

### New Proposed Zernike Moment Method

The same five formulas used in the traditional **Zernike** moment method are employed in the proposed new **Zernike** moment technique. As expected, all multi-dimensional formulas can reconstruct images with better qualities than the 1-dimensional formula does, and the 5-dimensional formula II obviously is the best approach.

In terms of image reconstruction, however, compared with the **Legendre** moment method, the **Zernike** moment method is severely handicapped. The reason is that the two major problems in the **Zernike** moment computation, geometrical error and approximation error, cannot be solved completely. Though carefully selecting the multi-dimensional cubature formulas can indeed reduce the computing errors and improve the quality of the reconstructed image, it is very unlikely that the performance of the **Zernike** moment method can reach the same level as that of the **Legendre** moment method.

# Chapter 5

# Reconstruction of Noisy Images via Moments

## 5.1 Introduction

Image reconstructions based on the orthogonal moments under noise free condition have been discussed in Chapter 4. However, in the presence of noise, the image reconstruction is expected to be more complicated.

It is interesting to consider how close we can recover the original image from a finite set of moments computed from the noisy data. Certainly, the higher order moments suffer greater degradation due to noise. On the other hand, higher moments are able to supply the detail information about the image [1][75]. These two opposite factors working against one another imply that there exists an optimal number of moments yielding the best possible representation of the image.

## 5.2 Legendre Moments

Two commonly used orthogonal moments for image reconstruction are **Zernike** moments and **Legendre** moments. In this chapter, the **Legendre** moments are employed for discussion. However, the results presented can be extended straightforwardly to other types of orthogonal moments.

As discussed in Chapter 4, if only **Legendre** moments $\lambda_{mn}$ of order $\leq M_{max}$ are given, the original image function $f(x,y)$ can be approximated by a truncated series:

$$\widehat{f}_{M_{max}}(x,y) = \sum_{m=0}^{M_{max}} \sum_{n=0}^{m} \widehat{\lambda}_{m-n,n} \, P_{m-n}(x) \, P_n(y). \tag{5.1}$$

Clearly, the square reconstruction error

$$Error(\widehat{f}_{M_{max}}) = \int\int [\widehat{f}_{M_{max}}(x,y) - f(x,y)]^2 \, dxdy \tag{5.2}$$

goes to zero as $M_{max} - > \infty$, see formula (4.12). That is, by employing higher order moments one can make the reconstruction error arbitrarily small. However, this scheme breaks down if the image is contaminated by noise. The noise affects the higher order moments greater than it does to the lower order moments[75]. Therefore, given the minimal value of the reconstruction error, an optimal number of moments exists. In other words, when noise is involved, the square reconstruction error will initially decrease (not necessarily in a monotonic way) down to a certain number of moments and then increase to infinity as $N - > \infty$.

## 5.3 The Reconstruction Error

Let $g(x,y)$ be the noisy degraded version of $f(x,y)$ and adopt the following simple image observation model

$$g(x,y) = f(x,y) + z(x,y), \tag{5.3}$$

where $z(x,y)$ is a Gaussian random process with zero mean and finite variance $\sigma^2$.

From the discussion in Chapter 3, the **Legendre** moments of the noisy version $g(x,y)$ of $f(x,y)$ can be obtained numerically by the formula

$$\widetilde{\lambda}_{mn} = \frac{(2m+1)(2n+1)}{4} \sum_{i=1}^{m} \sum_{j=1}^{n} h_{\widetilde{\lambda}_{mn}}(x_i, y_j) \, g(x_i, y_j) \tag{5.4}$$

where $\widetilde{\lambda}_{mn}$ presents the **Legendre** moments obtained from the noisy image $g(x, y)$ and

$$h_{\widetilde{\lambda}_{mn}}(x_i, y_j) = \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} \int_{y_j - \frac{\Delta y}{2}}^{y_j + \frac{\Delta y}{2}} P_m(x)\, P_n(y)\, dxdy. \tag{5.5}$$

Then, if the order $\leq M_{max}$ is given, the noisy image $g(x, y)$ can be reconstructed by

$$\widehat{g}_{M_{max}}(x, y) = \sum_{m=0}^{M_{max}} \sum_{n=0}^{m} \widetilde{\lambda}_{m-n,n}\, P_{m-n}(x)\, P_n(y). \tag{5.6}$$

Since

$$\mathbf{E} g(x_i, y_j) = f(x_i, y_j), \tag{5.7}$$

we have

$$\mathbf{E}\widetilde{\lambda}_{mn} = \widehat{\lambda}_{mn}. \tag{5.8}$$

If we write $\widetilde{\lambda}_{mn}$ as

$$\widetilde{\lambda}_{mn} = \widetilde{\lambda}_{mn} - \mathbf{E}\widetilde{\lambda}_{mn} + \mathbf{E}\widetilde{\lambda}_{mn},$$

then, it follows

$$\widetilde{\lambda}_{mn} = [\widetilde{\lambda}_{mn} - \mathbf{E}\widetilde{\lambda}_{mn}] + \widehat{\lambda}_{mn}. \tag{5.9}$$

Similar to the case of non-noise in (4.12), $Error(\widehat{g}_{M_{max}})$ can be writen as

$$\begin{aligned} Error(\widehat{g}_{M_{max}}) &= \int_{-1}^{1}\int_{-1}^{1} [\widehat{g}_{M_{max}}(x, y) - f(x, y)]^2\, dxdy \\ &= 4 \sum_{m=0}^{M_{max}} \sum_{n=0}^{m} \frac{1}{2(m-n)+1}\, \frac{1}{2n+1}\, [\widetilde{\lambda}_{m-n,n} - \lambda_{m-n,n}]^2 \\ &\quad + 4 \sum_{m=M_{max}+1}^{\infty} \sum_{n=0}^{m} \frac{1}{2(m-n)+1}\, \frac{1}{2n+1}\, \lambda_{m-n,n}^2. \end{aligned} \tag{5.10}$$

Therefore, $\mathbf{E}(Error(\widehat{g}_{M_{max}}))$ has the form of

$$\begin{aligned} \mathbf{E}(Error(\widehat{g}_{M_{max}})) &= 4 \sum_{m=0}^{M_{max}} \sum_{n=0}^{m} \frac{1}{2(m-n)+1}\, \frac{1}{2n+1}\, \mathbf{E}[\widetilde{\lambda}_{m-n,n} - \lambda_{m-n,n}]^2 \\ &\quad + 4 \sum_{m=M_{max}+1}^{\infty} \sum_{n=0}^{m} \frac{1}{2(m-n)+1}\, \frac{1}{2n+1}\, \mathbf{E}(\lambda_{m-n,n}^2). \end{aligned} \tag{5.11}$$

Considering (5.9), we have

$$\begin{aligned}\mathbf{E}[\widetilde{\lambda}_{m-n,n} - \lambda_{m-n,n}]^2 &= \mathbf{E}[(\widetilde{\lambda}_{m-n,n} - \mathbf{E}\widetilde{\lambda}_{m-n,n}) + (\widehat{\lambda}_{m-n,n} - \lambda_{m-n,n})]^2 \\ &= \mathbf{E}[(\widetilde{\lambda}_{m-n,n} - \mathbf{E}\widetilde{\lambda}_{m-n,n})^2 + (\widehat{\lambda}_{m-n,n} - \lambda_{m-n,n})^2 \\ &\quad -2\mathbf{E}(\widetilde{\lambda}_{m-n,n} - \mathbf{E}\widetilde{\lambda}_{m-n,n})(\widehat{\lambda}_{m-n,n} - \lambda_{m-n,n}). \qquad (5.12)\end{aligned}$$

Since

$$\mathbf{E}(\widetilde{\lambda}_{m-n,n} - \mathbf{E}\widetilde{\lambda}_{m-n,n})(\widehat{\lambda}_{m-n,n} - \lambda_{m-n,n}) = 0,$$

it follows

$$\begin{aligned}\mathbf{E}[\widetilde{\lambda}_{m-n,n} - \lambda_{m-n,n}]^2 &= \mathbf{E}(\widetilde{\lambda}_{m-n,n} - \mathbf{E}\widetilde{\lambda}_{m-n,n})^2 + \mathbf{E}(\widehat{\lambda}_{m-n,n} - \lambda_{m-n,n})^2 \\ &= var(\widetilde{\lambda}_{m-n,n}) + (\widehat{\lambda}_{m-n,n} - \lambda_{m-n,n})^2. \qquad (5.13)\end{aligned}$$

Then, (5.11) becomes

$$\begin{aligned}\mathbf{E}(Error(\widehat{g}_{M_{max}})) &= 4\sum_{m=0}^{M_{max}}\sum_{n=0}^{m}\frac{1}{2(m-n)+1}\,\frac{1}{2n+1}\,var(\widetilde{\lambda}_{m-n,n}) \\ &\quad +4\sum_{m=0}^{M_{max}}\sum_{n=0}^{m}\frac{1}{2(m-n)+1}\,\frac{1}{2n+1}\,(\widehat{\lambda}_{m-n,n} - \lambda_{m-n,n})^2 \\ &\quad +4\sum_{m=M_{max}+1}^{\infty}\sum_{n=0}^{m}\frac{1}{2(m-n)+1}\,\frac{1}{2n+1}\,\lambda_{m-n,n}^2. \qquad (5.14)\end{aligned}$$

The first term on the right-hand side of (5.14) depends on the noise added to the original image. When the noise increases, it increases too. The second term on the right side can be viewed as a matching measure between $\widehat{\lambda}_{m-n,n}$ and $\lambda_{mn}$ based on the total $\frac{(N+2)(N+1)}{2}$ moments, while the last term comes from truncating higher order moments in reconstruction.

Comparing (4.12) with (5.14), the main difference between the cases of the absence and presence of noise is focused on the first term on the right side of (5.14). In terms of the sensitivity to noise, the higher order **Legendre** moments are more sensitive.

From (5.14) we can see that when the order of $M_{max}$ increases, the sums of $var(\widetilde{\lambda}_{m-n,n})$ and $[\widehat{\lambda}_{mn} - \lambda_{mn}]^2$ increase. On the other hand, however, the third

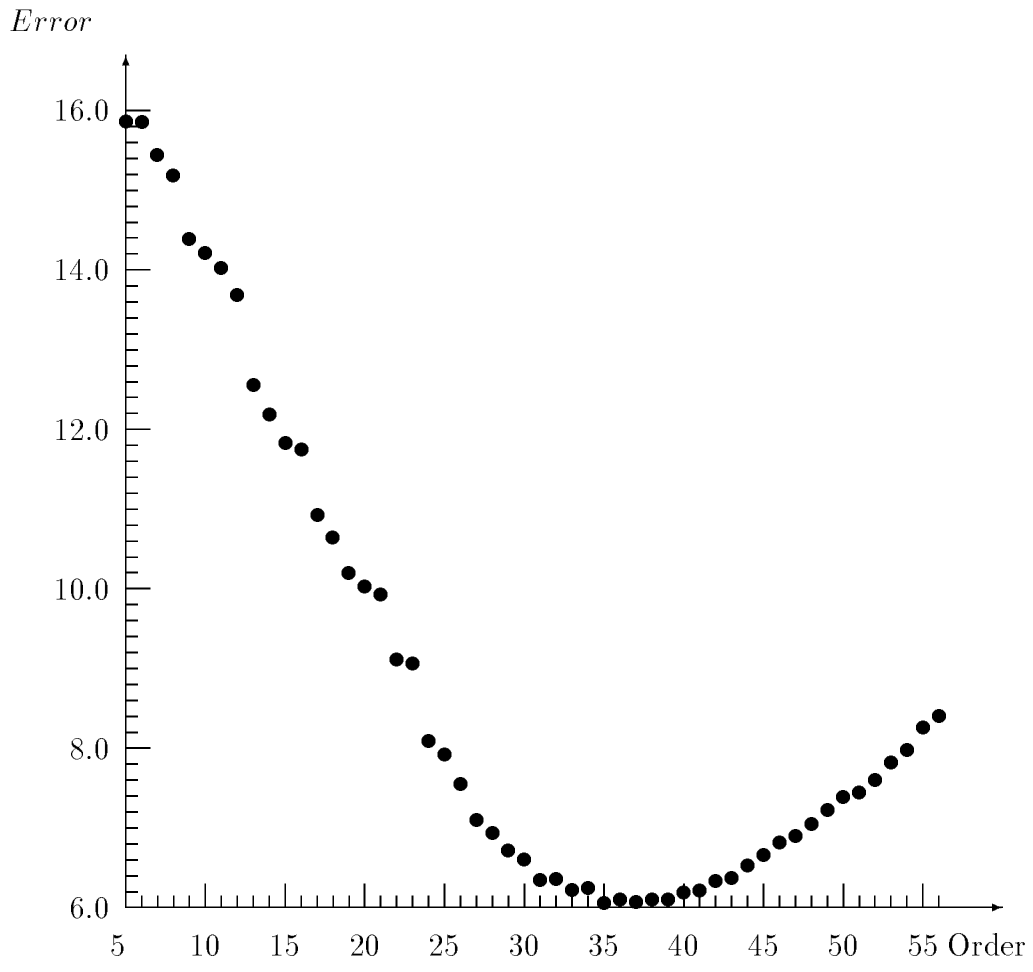


Figure 5.1: Square error $Error(\tilde{g}_{M_{max}})$, $\sigma^2 = 4.0$.

term on the right-hand side of (5.14) decreases when the order of $M_{max}$ increases. These two factors against each other, indicating that the square reconstruction error $Error(\widetilde{g})$ will initially decrease down to an optimal number of moments and then increase.

In order to verify such properties, the Chinese character $C_1$ is employed as the testing pattern in our experiment. Figure 5.1 shows the trend of the squared reconstruction error $Error(\widetilde{g}_{M_{max}})$ averaged on 10 runs with $\sigma^2 = 4.0$. As expected, the error decreases first, reaches minimum at $N = 35$, then increases. Table 5.1 lists the numerical values of $Error(\widehat{g}_{M_{max}})$. Figure 5.2 illustrates the noisy image of $C_1$ and its reconstructed versions from order 4 up to order 56, from left to right, first row to last row, respectively.

| Order | $I(N)$ | Order | $I(N)$ | Order | $I(N)$ |
|---|---|---|---|---|---|
| 3 | 1.628e+01 | 21 | 9.928e+00 | 39 | 6.099e+00 |
| 4 | 1.604e+01 | 22 | 9.110e+00 | 40 | 6.193e+00 |
| 5 | 1.587e+01 | 23 | 9.067e+00 | 41 | 6.217e+00 |
| 6 | 1.586e+01 | 24 | 8.097e+00 | 42 | 6.323e+00 |
| 7 | 1.544e+01 | 25 | 7.924e+00 | 43 | 6.377e+00 |
| 8 | 1.518e+01 | 26 | 7.556e+00 | 44 | 6.529e+00 |
| 9 | 1.439e+01 | 27 | 7.101e+00 | 45 | 6.660e+00 |
| 10 | 1.421e+01 | 28 | 6.939e+00 | 46 | 6.814e+00 |
| 11 | 1.403e+01 | 29 | 6.718e+00 | 47 | 6.898e+00 |
| 12 | 1.369e+01 | 30 | 6.610e+00 | 48 | 7.046e+00 |
| 13 | 1.256e+01 | 31 | 6.349e+00 | 49 | 7.223e+00 |
| 14 | 1.219e+01 | 32 | 6.359e+00 | 50 | 7.386e+00 |
| 15 | 1.183e+01 | 33 | 6.219e+00 | 51 | 7.444e+00 |
| 16 | 1.175e+01 | 34 | 6.243e+00 | 52 | 7.606e+00 |
| 17 | 1.093e+01 | 35 | 6.065e+00 | 53 | 7.818e+00 |
| 18 | 1.065e+01 | 36 | 6.103e+00 | 54 | 7.982e+00 |
| 19 | 1.020e+01 | 37 | 6.069e+00 | 55 | 8.258e+00 |
| 20 | 1.003e+01 | 38 | 6.105e+00 | 56 | 8.407e+00 |

Table 5.1: Square reconstruction error $Error(\widetilde{g}_{M_{max}})$ with $\sigma^2 = 4.0$.

Obviously, the second and third terms in (5.14) are not affected by the noise, therefore, when the noise increases or decreases, the sum of $[\widehat{\lambda}_{mn} - \lambda_{mn}]^2$ is the

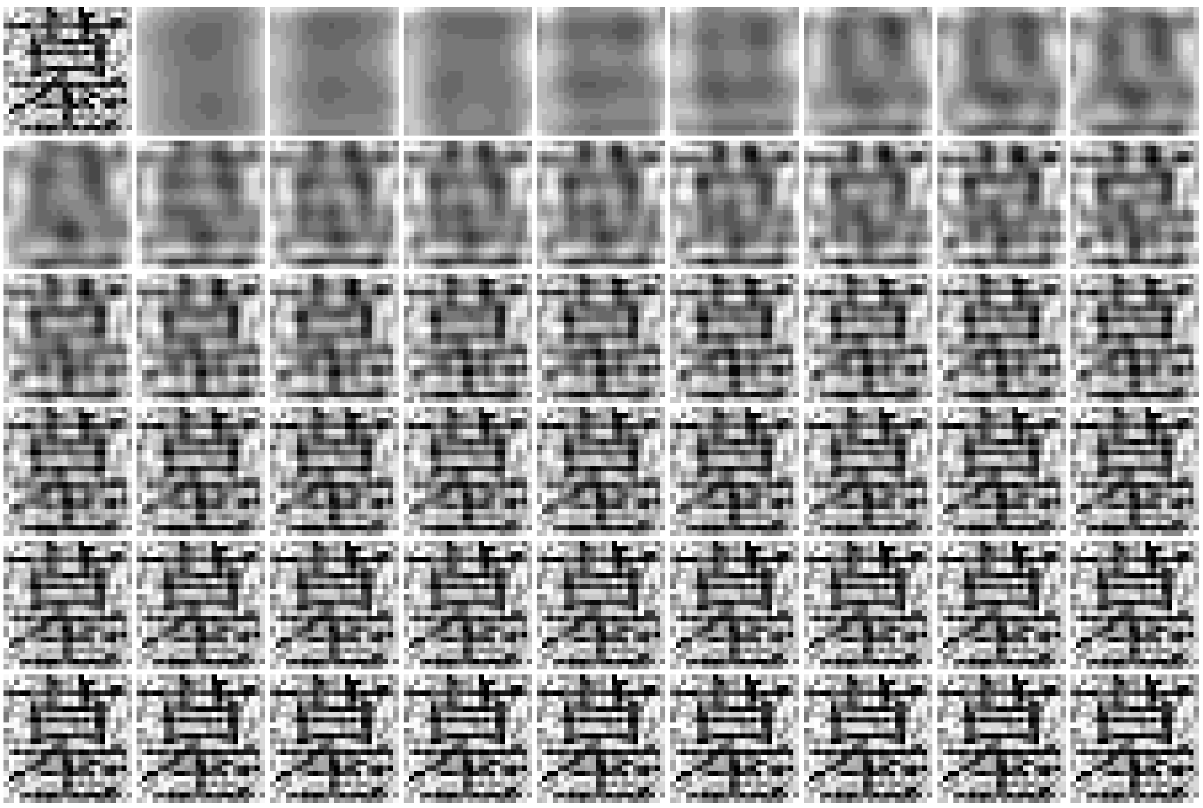


Figure 5.2: Noisy version of $C_1$, with $\sigma^2 = 4.0$, and its reconstructed versions.

only factor deciding the position of the optimal number of moments. Since the noise affects the higher order **Legendre** moments more than it does to the lower ones, the reconstruction error $Error(\widehat{g}_{M_{max}})$ will increase faster when the higher level noise is involved. These discussions lead to the conclusion that when the level of noise increases, the optimal number of moments for the least reconstruction error becomes smaller. An experiment was designed to verify this assumption and the result is illustrated in Figure 5.3. The same Chinese character $C_1$ and the noise model shown in (5.3) are employed. The result is averaged on 10 runs and the noise varies from $\sigma^2 = 4.0$ to $\sigma^2 = 25.0$. Due to the nature of this experiment, the computation involved is very large. It took 260 hours for a 33 MHz 486 computer to complete the task. It is fair to say that the amount of computation involved in this experiment has reached the limitation of a current personal computer.

## 5.4 Data-Driven Selection of the Optimal Number

It is very interesting to consider how to select a "good" optimal number $N$ directly from the available $g(x_i, y_j)$. Ideally, it is expected to have $N_0^*$ to minimize the square reconstruction error. Notice that $N_0^*$ is a function of the data at hand.

This, in turn, is equivalent to taking the minimizer of the following criteria

$$\begin{aligned}
Error(\widehat{g}_{M_{max}}) - \iint f^2(x,y)dxdy &= 4\sum_{m=0}^{M_{max}}\sum_{n=0}^{m}\frac{1}{2(m-n)+1}\,\frac{1}{2n+1}[\widetilde{\lambda}_{m-n,n}-\lambda_{m-n,n}]^2 \\
&+4\sum_{m=M_{max}+1}^{\infty}\sum_{n=0}^{m}\frac{1}{2(m-n)+1}\,\frac{1}{2n+1}\lambda_{m-n,n}^2 \\
&-4\sum_{m=0}^{\infty}\sum_{n=0}^{m}\frac{1}{2(m-n)+1}\,\frac{1}{2n+1}\lambda_{m-n,n}^2 \\
Error(\widehat{g}_{M_{max}}) - \iint f^2(x,y)dxdy &= 4\sum_{m=0}^{M_{max}}\sum_{n=0}^{m}\frac{1}{2(m-n)+1}\,\frac{1}{2n+1}[\widetilde{\lambda}_{m-n,n}-\lambda_{m-n,n}]^2 \\
&-4\sum_{m=0}^{M_{max}}\sum_{n=0}^{m}\frac{1}{2(m-n)+1}\,\frac{1}{2n+1}\lambda_{m-n,n}^2. \qquad (5.15)
\end{aligned}$$

However, the solution of equation (5.15) is not feasible since the $\lambda_{mn}$'s are unknown.

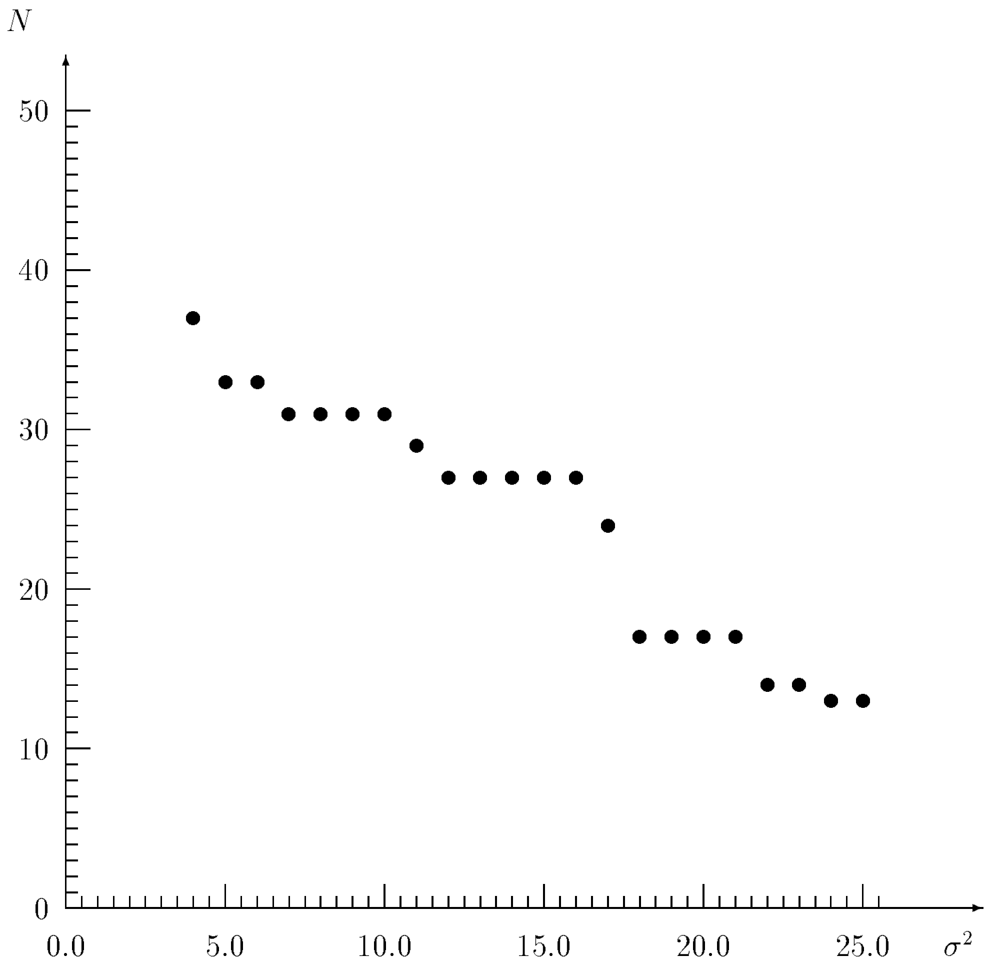


Figure 5.3: Optimal moments numbers.

If $\lambda_{mn}$ is replaced by $\widetilde{\lambda}_{mn}$, equation (5.15) will yield the unacceptable solution $N = \infty$.

To overcome this difficulty, a resampling technique utilizing the cross-validation methodology has been introduced and the asymptotic optimality of such a selection has been proved [56][57].

Other possible techniques to solve this problem include the utilization of discrete measures with penalty factors. For instance, other than the $Error(\widehat{g}_{M_{max}})$ criterion, its discrete approximation

$$ED(M) = \Delta x \Delta y \sum_{i=1}^{n} \sum_{j=1}^{m} [g(x_i, y_j) - \widehat{g}_{M_{max}}(x_i, y_j)]^2 \tag{5.16}$$

can be used[24].

The empirical selectors corresponding to $ED(M)$ are of the form

$$\widehat{ED}(M) = \Delta x \Delta y \sum_{i=1}^{n} \sum_{j=1}^{m} [g(x_i, y_j) - \widehat{g}_{M_{max}}(x_i, y_j)]^2 \Phi(N), \tag{5.17}$$

i.e., it is a penalized version of the residual error

$$\Delta x \Delta y \sum_{i=1}^{n} \sum_{j=1}^{m} [g(x_i, y_j) - \widehat{f}_{M_{max}}(x_i, y_j)]^2,$$

see [24].

In the case of Gaussian noise, the prescription we proposed for the penalty factor $\Phi(N)$ is

$$\Phi(N) = [1 - F(\sigma^2) L(N) \Delta x \Delta y]^{-p}, \tag{5.18}$$

where $L(N)$ is the total number of moments used in $\widehat{g}_{M_{max}}$, e.g., $L(N) = \frac{(N+1)(N+2)}{2}$ for **Legendre** moments. With carefully selected $p$ and $F(\sigma^2)$, significant simulating results of Figure 5.3 can be expected.

Generally speaking, the automatic selection of the optimal number $N_0^*$ from the data at hand is still an open problem. Though some initial experimental results are quite positive on this task[58][59], due to the extreme amount of computation involved, we could not provide the full scale research on this issue based on the available equipment. For a related problem in the context of image restoration, we refer to [29].

# Chapter 6

# Character Recognition via Moments

## 6.1 Introduction

Character recognition is believed to be typical of many other practical problems that depend on general shapes rather than details of the image. The recognition of characters from imagery may be accomplished by identifying an unknown character as a member of a set of known characters. Various character recognition techniques have been utilized to abstract characterizations for efficient character representations[25][62]. Such characterizations are defined by measurable features extracted from the characters. Therefore, the effectiveness of the technique for a given application is dependent on the ability of a given technique to uniquely represent the character from the available information. Since no one single technique will be effective for all recognition problems, the choice of character characterization is driven by the requirements of a specific recognition task.

Based on the **Uniqueness Theorem**[37], the double moment sequence is uniquely determined by an image function $f(x, y)$. This nature makes the method of moments a proper candidate in character recognition.

## 6.2 Character Recognition via Central Moments

In consideration of the fact that there is no inverse problem involved in the classification of visual patterns and characters, and the property of invariance under translation, the classical moment is discussed in this chapter for the purpose of pattern recognition.

As mentioned in Chapter 2, the central moments $\mu_{pq}$ are defined in (2.5)

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - \bar{x})^p \, (y - \bar{y})^q \, f(x, y) \, dx dy,$$

where

$$\bar{x} = \frac{M_{10}}{M_{00}}, \qquad \bar{y} = \frac{M_{01}}{M_{00}},$$

and $M_{pq}$ are the classic moments defined in (2.4)

$$M_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p \, y^q \, f(x, y) dx dy.$$

Hu demonstrated the utility of moment techniques through a simple pattern recognition experiment[37]. The first two moment invariants were used to represent several known digitized patterns in a two-dimensional feature space. The experiment was performed by using a set of 26 capital letters as input patterns. In the two-dimensional feature space, all the points representing each of the characters were fairly distinct except those of M and W.

Compared with the set of English letters, the Chinese character set is large, and in terms of character recognition, is more difficult to classify. In this section, similar to Hu's experiment, a simulation program of a character recognition model using two moment invariants, has been proposed. The following two moment functions

$$X_1 = \sqrt{\mu_{20} + \mu_{02}} \tag{6.1}$$

and

$$X_2 = \sqrt{(\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2} \tag{6.2}$$

are used to compute the representations of all known characters in the feature space. Therefore, each point of $(X_1, X_2)$ represents one Chinese character in the image plane $(x, y)$.

Considering the similarity, first, we employ the set of Chinese characters used before. Figure 6.1 shows these characters.

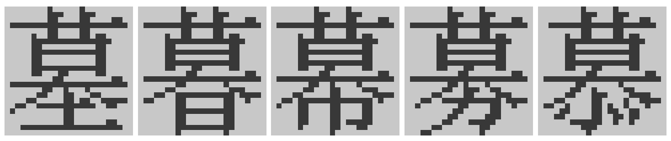


Figure 6.1: Five original Chinese characters used for testing.

The values of $X_1$ and $X_2$ are given in Table 6.1 and the representations of the five Chinese characters are shown in Figure 6.2.

| Characters | $X_1$ | $X_2$ |
|---|---|---|
| $C_1$ | 3.5328 | 0.0933 |
| $C_2$ | 3.5440 | 0.0818 |
| $C_3$ | 3.5433 | 0.0616 |
| $C_4$ | 3.5574 | 0.0254 |
| $C_5$ | 3.5559 | 0.0794 |

Table 6.1: Values of the five Chinese characters in the central moment feature space.

Figure 6.2 shows that the representations of the five Chinese characters are quite close to each other in the two dimensional $(X_1, X_2)$ feature space. From the classification point of view, this disadvantage certainly will limit the usage of the central moment method in Chinese character recognition tasks.

Then, randomly, we selected 90 Chinese characters as the testing samples. Figure 6.3 shows these 90 Chinese characters. In Figure 6.3, we call the first sample from the left in the first row $S_1$, the second sample from the left on the same row $S_2$, and so on. For example, $S_{38}$ will be the second sample from the left on the fifth row; and $S_{77}$ is the fifth character from the left on the ninth row.

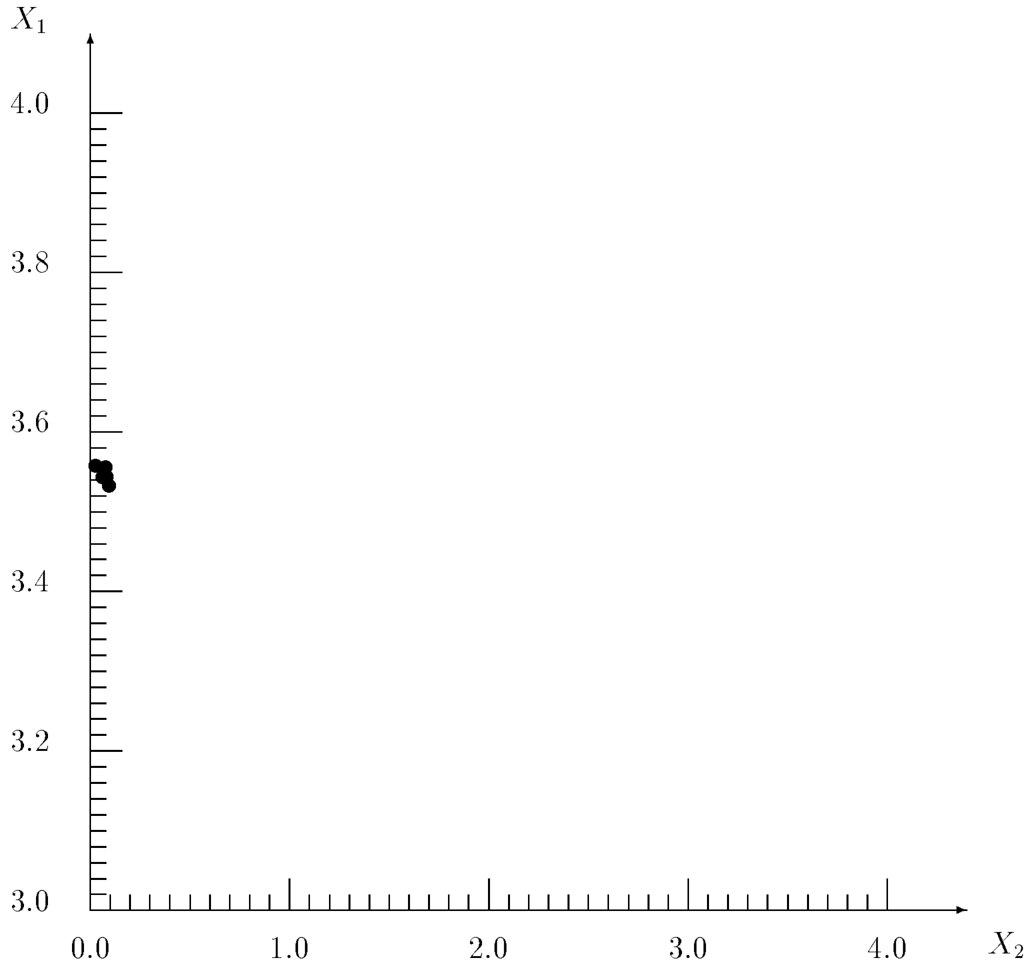


Figure 6.2: Representations of the five Chinese characters in the central moment feature space.

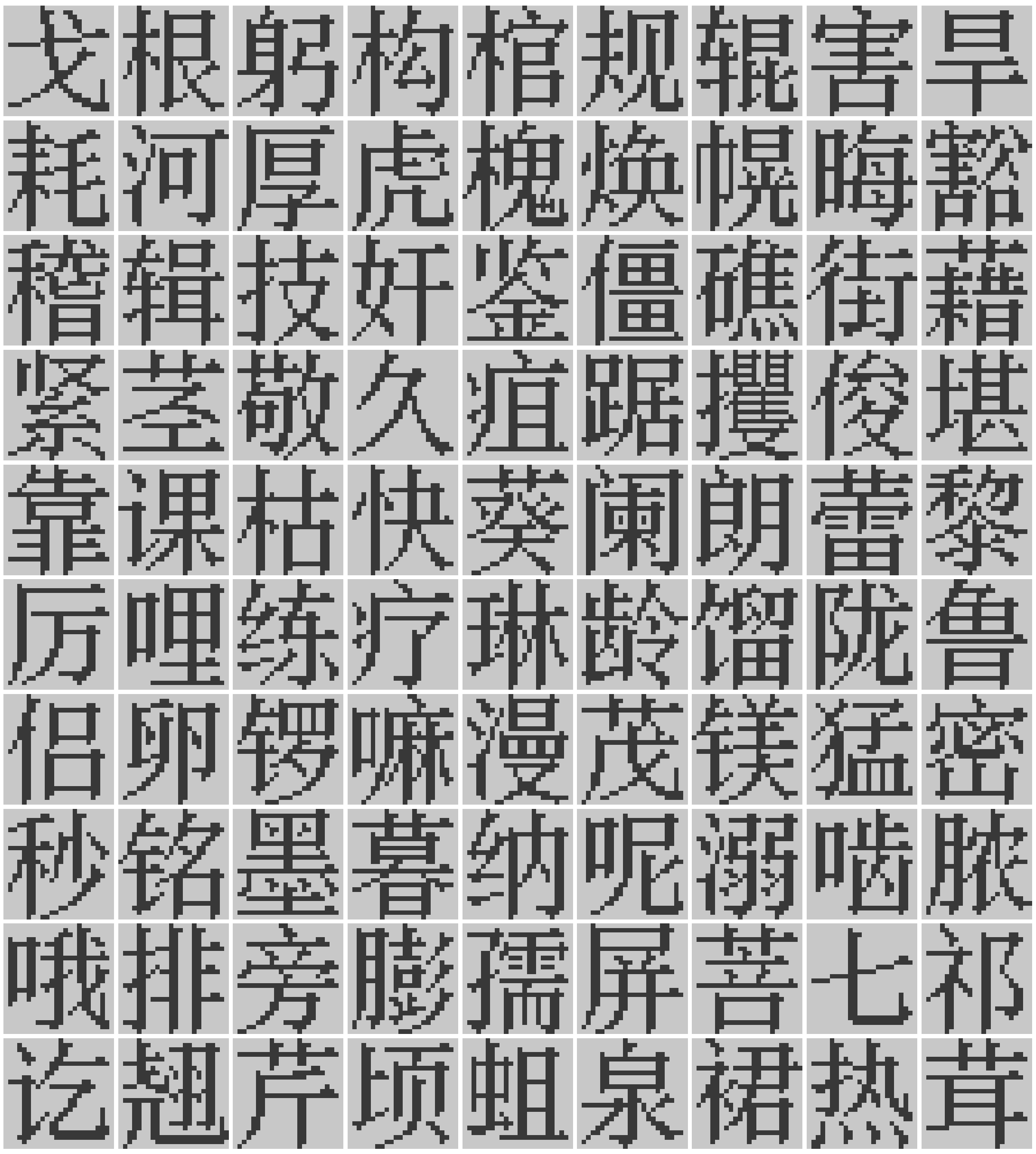


Figure 6.3: Ninety Chinese characters.

The values of $X_1$ and $X_2$ are listed in Table 6.2 and the representations of these 90 Chinese characters in the central moment feature space are plotted in Figure 6.4.

Obviously, similar to the results shown in Table 6.1 and Figure 6.2, the two central moment functions, $X_1$ and $X_2$, cannot recognize those Chinese characters successfully.

## 6.3 Character Recognition with Legendre Moments

The **Legendre** moments do not have the property of invariance under translation. However, compared with the classical moments, the same order of the **Legendre** moment contains more terms than that of the central moment does. Therefore, in terms of classification, the **Legendre** moments contain more information than the central moments do.

Similar to the two classification measures defined in (6.1) and (6.2), the following two **Legendre** functions

$$Y_1 = \sqrt{\lambda_{20} + \lambda_{02}} \tag{6.3}$$

and

$$Y_2 = \sqrt{(\lambda_{30} - 3\lambda_{12})^2 + (3\lambda_{21} - \lambda_{03})^2} \tag{6.4}$$

are employed in our new recognition model, where $\lambda_{mn}$'s are the **Legendre** moments defined in (2.31).

First, we use the same set of five Chinese characters shown in Figure 6.1. The values of all five Chinese characters in the two-dimensional **Legendre** moment feature space $(Y_1, Y_2)$ are listed and illustrated in Table 6.3 and Figure 6.5, respectively.

We can see that the five Chinese characters are well separated in the two-dimensional **Legendre** moment feature space $(Y_1, Y_2)$. In other words, in this particular Chinese character recognition task, the **Legendre** moment technique is superior.

| Sample | $X_1$ | $X_2$ | Sample | $X_1$ | $X_2$ | Sample | $X_1$ | $X_2$ |
|---|---|---|---|---|---|---|---|---|
| $S_1$ | 3.6184 | 0.2566 | $S_{31}$ | 3.6512 | 0.2411 | $S_{61}$ | 3.5429 | 0.0565 |
| $S_2$ | 3.5487 | 0.0162 | $S_{32}$ | 3.5564 | 0.1498 | $S_{62}$ | 3.5485 | 0.1071 |
| $S_3$ | 3.5474 | 0.0029 | $S_{33}$ | 3.4875 | 0.0674 | $S_{62}$ | 3.5618 | 0.0980 |
| $S_4$ | 3.5504 | 0.0769 | $S_{34}$ | 3.4743 | 0.1785 | $S_{64}$ | 3.5798 | 0.1332 |
| $S_5$ | 3.5551 | 0.0811 | $S_{35}$ | 3.5606 | 0.0598 | $S_{65}$ | 3.5286 | 0.1251 |
| $S_6$ | 3.5339 | 0.2560 | $S_{36}$ | 3.5578 | 0.1511 | $S_{66}$ | 3.4988 | 0.1657 |
| $S_7$ | 3.5092 | 0.1201 | $S_{37}$ | 3.5509 | 0.0610 | $S_{67}$ | 3.5440 | 0.0818 |
| $S_8$ | 3.5725 | 0.1149 | $S_{38}$ | 3.5382 | 0.1388 | $S_{68}$ | 3.5354 | 0.0843 |
| $S_9$ | 3.6230 | 0.0114 | $S_{39}$ | 3.5589 | 0.0596 | $S_{69}$ | 3.5327 | 0.2650 |
| $S_{10}$ | 3.5499 | 0.1434 | $S_{40}$ | 3.5840 | 0.0639 | $S_{70}$ | 3.5091 | 0.0972 |
| $S_{11}$ | 3.5749 | 0.0771 | $S_{41}$ | 3.5689 | 0.0613 | $S_{71}$ | 3.5424 | 0.2206 |
| $S_{12}$ | 3.5591 | 0.0477 | $S_{42}$ | 3.4838 | 0.1280 | $S_{72}$ | 3.5314 | 0.1080 |
| $S_{13}$ | 3.5577 | 0.2699 | $S_{43}$ | 3.5259 | 0.0878 | $S_{73}$ | 3.5216 | 0.2923 |
| $S_{14}$ | 3.4994 | 0.1735 | $S_{44}$ | 3.5056 | 0.1187 | $S_{74}$ | 3.5340 | 0.0785 |
| $S_{15}$ | 3.5576 | 0.1663 | $S_{45}$ | 3.5435 | 0.1615 | $S_{75}$ | 3.5996 | 0.0398 |
| $S_{16}$ | 3.4926 | 0.1638 | $S_{46}$ | 3.5750 | 0.0304 | $S_{76}$ | 3.5075 | 0.1668 |
| $S_{17}$ | 3.5146 | 0.1091 | $S_{47}$ | 3.5053 | 0.3824 | $S_{77}$ | 3.5067 | 0.0730 |
| $S_{18}$ | 3.5050 | 0.0607 | $S_{48}$ | 3.5609 | 0.1297 | $S_{78}$ | 3.5466 | 0.0675 |
| $S_{19}$ | 3.5035 | 0.0953 | $S_{49}$ | 3.6086 | 0.1638 | $S_{79}$ | 3.5668 | 0.0329 |
| $S_{20}$ | 3.5339 | 0.0733 | $S_{50}$ | 3.5615 | 0.0674 | $S_{80}$ | 3.6527 | 0.1474 |
| $S_{21}$ | 3.5569 | 0.0480 | $S_{51}$ | 3.5390 | 0.1286 | $S_{81}$ | 3.5889 | 0.0140 |
| $S_{22}$ | 3.5958 | 0.1324 | $S_{52}$ | 3.4933 | 0.1307 | $S_{82}$ | 3.5776 | 0.1211 |
| $S_{23}$ | 3.5467 | 0.0199 | $S_{53}$ | 3.4940 | 0.1729 | $S_{83}$ | 3.4619 | 0.2198 |
| $S_{24}$ | 3.5022 | 0.1514 | $S_{54}$ | 3.5621 | 0.0845 | $S_{84}$ | 3.5986 | 0.0416 |
| $S_{25}$ | 3.5264 | 0.1419 | $S_{55}$ | 3.5675 | 0.0680 | $S_{85}$ | 3.5646 | 0.1387 |
| $S_{26}$ | 3.5378 | 0.0837 | $S_{56}$ | 3.5750 | 0.1012 | $S_{86}$ | 3.5460 | 0.0678 |
| $S_{27}$ | 3.5033 | 0.1533 | $S_{57}$ | 3.5387 | 0.2538 | $S_{87}$ | 3.6025 | 0.0930 |
| $S_{28}$ | 3.5615 | 0.0590 | $S_{58}$ | 3.5175 | 0.2213 | $S_{88}$ | 3.5371 | 0.1616 |
| $S_{29}$ | 3.5787 | 0.1090 | $S_{59}$ | 3.5182 | 0.1422 | $S_{89}$ | 3.5542 | 0.2515 |
| $S_{30}$ | 3.5322 | 0.0200 | $S_{60}$ | 3.5571 | 0.2579 | $S_{90}$ | 3.5860 | 0.1156 |

Table 6.2: Values of the ninety Chinese characters in the central moment feature space.

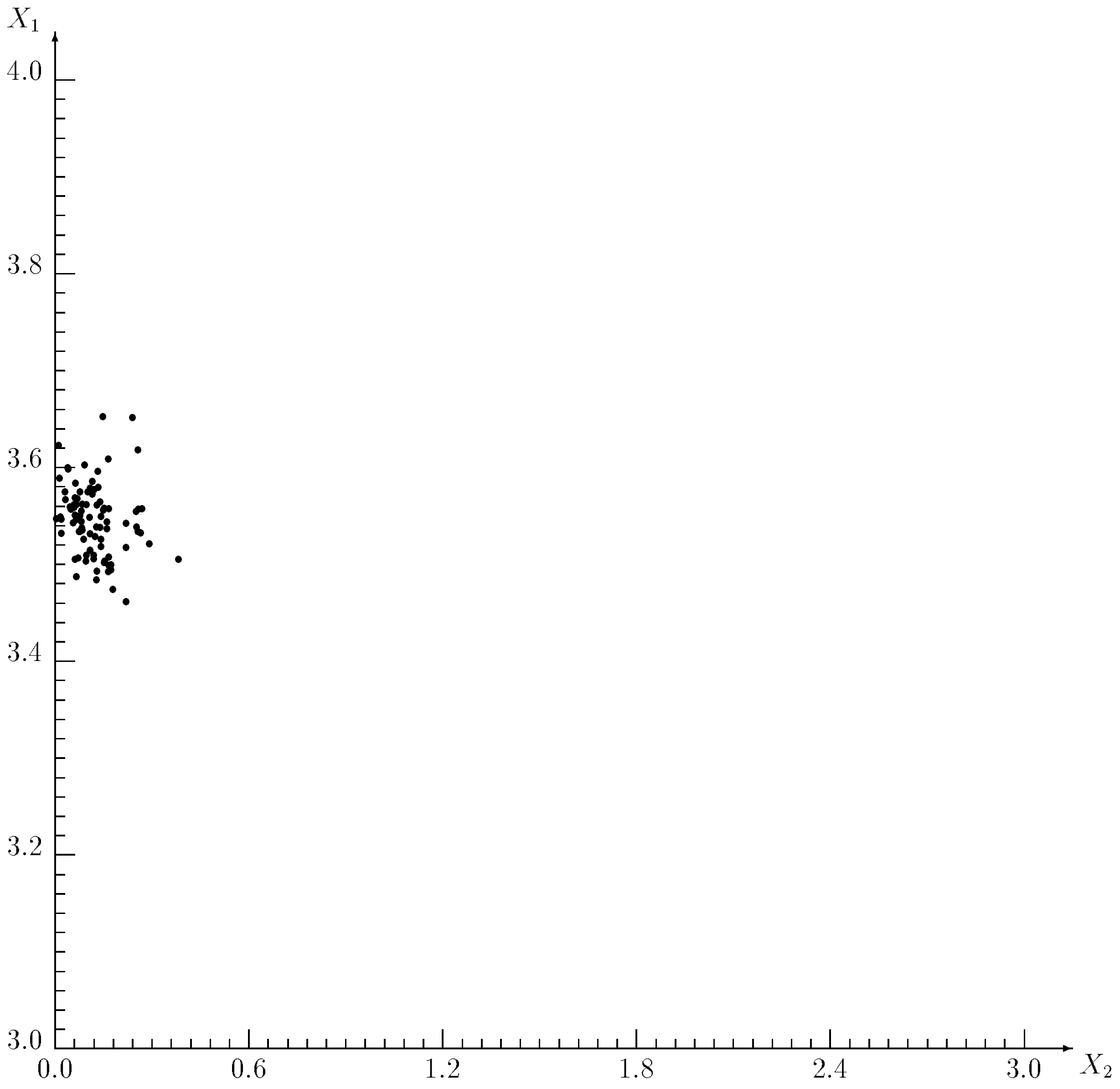


Figure 6.4: Representations of the ninety Chinese characters in the central moment feature space.

| Characters | $Y_1$ | $Y_2$ |
|---|---|---|
| $C_1$ | 2.0204 | 1.6965 |
| $C_2$ | 2.1636 | 0.2535 |
| $C_3$ | 2.1899 | 1.7548 |
| $C_4$ | 2.2032 | 2.1281 |
| $C_5$ | 2.1653 | 3.2418 |

Table 6.3: Values of the five Chinese characters in the **Legendre** moment feature space.

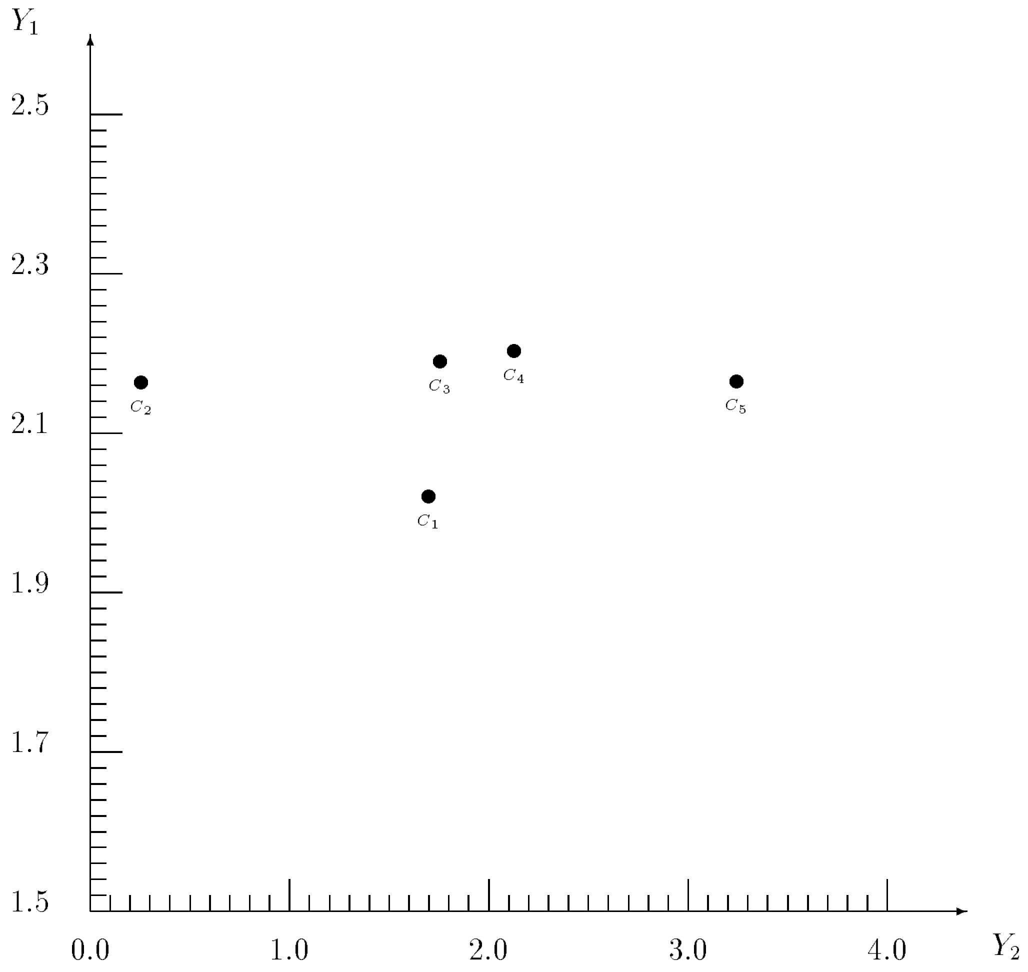


Figure 6.5: Representations of the five Chinese characters in the **Legendre** moment feature space.

Then, the ninety randomly selected Chinese characters shown in Figure 6.3 are employed as the testing patterns. Table 6.4 displays the values of these 90 Chinese characters in the **Legendre** moment feature space, and Figure 6.6 plots these representations in the two-dimensional $(Y_1, Y_2)$ plane.

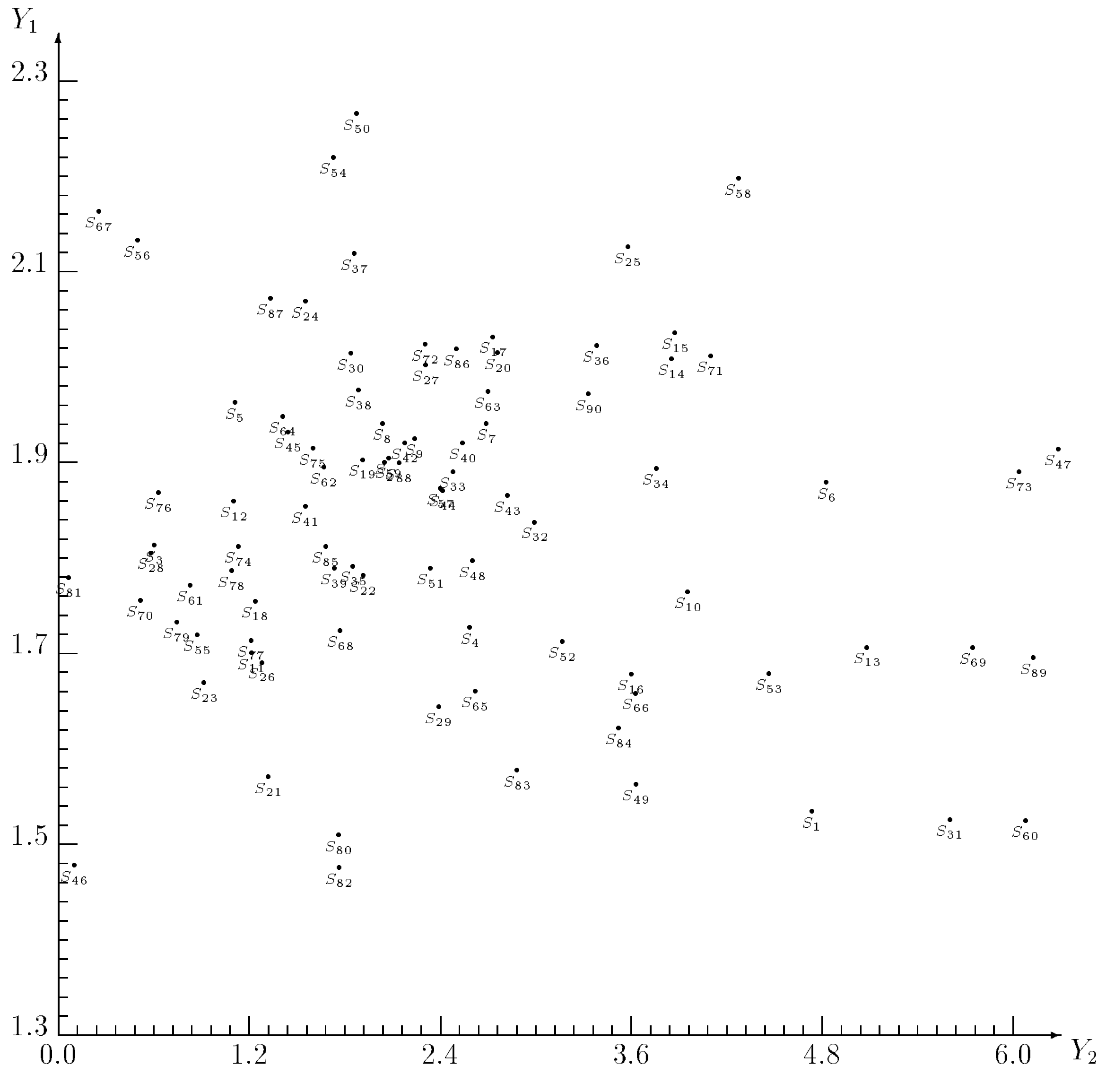


Figure 6.6: Representations of the ninety Chinese characters in the **Legendre** moment feature space.

Figure 6.6 shows that most of Chinese characters are well separated. However,

| Sample | $Y_1$ | $Y_2$ | Sample | $Y_1$ | $Y_2$ | Sample | $Y_1$ | $Y_2$ |
|---|---|---|---|---|---|---|---|---|
| $S_1$ | 1.5346 | 4.7334 | $S_{31}$ | 1.5260 | 5.6012 | $S_{61}$ | 1.7716 | 0.8286 |
| $S_2$ | 1.9007 | 2.0506 | $S_{32}$ | 1.8379 | 2.9933 | $S_{62}$ | 1.8951 | 1.6680 |
| $S_3$ | 1.8135 | 0.6016 | $S_{33}$ | 1.8906 | 2.4768 | $S_{63}$ | 1.9745 | 2.6982 |
| $S_4$ | 1.7271 | 2.5868 | $S_{34}$ | 1.8940 | 3.7551 | $S_{64}$ | 1.9486 | 1.4085 |
| $S_5$ | 1.9632 | 1.1077 | $S_{35}$ | 1.7915 | 1.8526 | $S_{65}$ | 1.6608 | 2.6182 |
| $S_6$ | 1.8792 | 4.8260 | $S_{36}$ | 2.0227 | 3.3827 | $S_{66}$ | 1.6585 | 3.6280 |
| $S_7$ | 1.9411 | 2.6879 | $S_{37}$ | 2.1192 | 1.8604 | $S_{67}$ | 2.1636 | 0.2535 |
| $S_8$ | 1.9408 | 2.0360 | $S_{38}$ | 1.9756 | 1.8860 | $S_{68}$ | 1.7241 | 1.7722 |
| $S_9$ | 1.9248 | 2.2374 | $S_{39}$ | 1.7896 | 1.7319 | $S_{69}$ | 1.7064 | 5.7469 |
| $S_{10}$ | 1.7645 | 3.9552 | $S_{40}$ | 1.9206 | 2.5398 | $S_{70}$ | 1.7558 | 0.5146 |
| $S_{11}$ | 1.7005 | 1.2156 | $S_{41}$ | 1.8540 | 1.5533 | $S_{71}$ | 2.0121 | 4.0992 |
| $S_{12}$ | 1.8596 | 1.1031 | $S_{42}$ | 1.9207 | 2.1761 | $S_{72}$ | 2.0239 | 2.3041 |
| $S_{13}$ | 1.7067 | 5.0808 | $S_{43}$ | 1.8654 | 2.8219 | $S_{73}$ | 1.8900 | 6.0365 |
| $S_{14}$ | 2.0086 | 3.8532 | $S_{44}$ | 1.8708 | 2.4153 | $S_{74}$ | 1.8122 | 1.1317 |
| $S_{15}$ | 2.0362 | 3.8763 | $S_{45}$ | 1.9319 | 1.4443 | $S_{75}$ | 1.9149 | 1.5990 |
| $S_{16}$ | 1.6787 | 3.5999 | $S_{46}$ | 1.4783 | 0.0995 | $S_{76}$ | 1.8685 | 0.6289 |
| $S_{17}$ | 2.0319 | 2.7340 | $S_{47}$ | 1.9144 | 6.2822 | $S_{77}$ | 1.7135 | 1.2120 |
| $S_{18}$ | 1.7548 | 1.2384 | $S_{48}$ | 1.7970 | 2.6013 | $S_{78}$ | 1.7870 | 1.0894 |
| $S_{19}$ | 1.9031 | 1.9106 | $S_{49}$ | 1.5632 | 3.6298 | $S_{79}$ | 1.7331 | 0.7432 |
| $S_{20}$ | 2.0153 | 2.7606 | $S_{50}$ | 2.2657 | 1.8740 | $S_{80}$ | 1.5097 | 1.7610 |
| $S_{21}$ | 1.5705 | 1.3187 | $S_{51}$ | 1.7894 | 2.3395 | $S_{81}$ | 1.7795 | 0.0637 |
| $S_{22}$ | 1.7817 | 1.9157 | $S_{52}$ | 1.7126 | 3.1672 | $S_{82}$ | 1.4753 | 1.7641 |
| $S_{23}$ | 1.6696 | 0.9155 | $S_{53}$ | 1.6793 | 4.4645 | $S_{83}$ | 1.5775 | 2.8834 |
| $S_{24}$ | 2.0691 | 1.5532 | $S_{54}$ | 2.2200 | 1.7281 | $S_{84}$ | 1.6220 | 3.5224 |
| $S_{25}$ | 2.1259 | 3.5769 | $S_{55}$ | 1.7195 | 0.8730 | $S_{85}$ | 1.8122 | 1.6799 |
| $S_{26}$ | 1.6907 | 1.2791 | $S_{56}$ | 2.1327 | 0.4946 | $S_{86}$ | 2.0188 | 2.5025 |
| $S_{27}$ | 2.0025 | 2.3081 | $S_{57}$ | 1.8733 | 2.3991 | $S_{87}$ | 2.0721 | 1.3334 |
| $S_{28}$ | 1.8056 | 0.5816 | $S_{58}$ | 2.1976 | 4.2732 | $S_{88}$ | 1.8996 | 2.1385 |
| $S_{29}$ | 1.6444 | 2.3904 | $S_{59}$ | 1.9050 | 2.0736 | $S_{89}$ | 1.6954 | 6.1273 |
| $S_{30}$ | 2.0144 | 1.8385 | $S_{60}$ | 1.5251 | 6.0759 | $S_{90}$ | 1.9719 | 3.3317 |

Table 6.4: Values of the ninety Chinese characters in the **Legendre** moment feature space.

it is observed that two characters, $S_{44}$ and $S_{57}$, are very close to each other in the **Legendre** moment feature space. Although the results shown in Figure 6.6 are indeed better than those of Figure 6.4, yet the **Legendre** moment two-dimensional feature space cannot be used as a successful technique to recognize a specific Chinese character from the whole Chinese character set.

One option to improve the **Legendre** moment technique is to add a new feature to the feature space. We use the following equation, which is based on (2.16), as the third feature:

$$Y_3 = \sqrt{(\lambda_{20} - \lambda_{02})^2 + 4\lambda_{11}^2}. \tag{6.5}$$

In this new three-dimensional **Legendre** moment feature space, characters $S_{44}$ and $S_{57}$ have $Y_3$ values 0.6933 and 1.6876, respectively. Therefore, all Chinese characters shown in Figure 6.3 can be separated successfully. Table 6.5 displays the values of $Y_1$, $Y_2$, and $Y_3$ for all ninety Chinese characters.

## 6.4 Conclusions

In this chapter, we have discussed character recognition via moment methods and compared the well known central moment feature space with the proposed new **Legendre** moment feature spaces for Chinese character recognition.

The two-dimensional central moment feature space was used by Hu[37] to recognize 26 English capital letters. The experiment performed fairly well except that the distance between two points representing letters M and W in the feature space is very close.

Compared with the set of English letters, however, the set of Chinese characters is larger and more difficult to classify. Two sets of Chinese characters, one including five similar Chinese characters and the other containing ninety randomly selected characters, are used as the input patterns to a simulation program based on the

| Sample | 1 | 2 | 3 | Sample | 1 | 2 | 3 |
|---|---|---|---|---|---|---|---|
| 1 | 1.5346 | 4.7334 | 2.6002 | 46 | 1.4783 | 0.0995 | 1.0175 |
| 2 | 1.9007 | 2.0506 | 1.8604 | 47 | 1.9144 | 6.2822 | 1.9013 |
| 3 | 1.8135 | 0.6016 | 2.0815 | 48 | 1.7970 | 2.6013 | 2.5707 |
| 4 | 1.7271 | 2.5868 | 3.0566 | 49 | 1.5632 | 3.6298 | 1.3954 |
| 5 | 1.9632 | 1.1077 | 1.5408 | 50 | 2.2657 | 1.8740 | 2.8191 |
| 6 | 1.8792 | 4.8260 | 2.8729 | 51 | 1.7894 | 2.3395 | 2.3448 |
| 7 | 1.9411 | 2.6879 | 2.7054 | 52 | 1.7126 | 3.1672 | 0.5698 |
| 8 | 1.9408 | 2.0360 | 0.9139 | 53 | 1.6793 | 4.4645 | 2.8409 |
| 9 | 1.9248 | 2.2374 | 2.0778 | 54 | 2.2200 | 1.7281 | 2.2174 |
| 10 | 1.7645 | 3.9552 | 3.2093 | 55 | 1.7195 | 0.8730 | 1.5833 |
| 11 | 1.7005 | 1.2156 | 2.4936 | 56 | 2.1327 | 0.4964 | 2.7262 |
| 12 | 1.8596 | 1.1031 | 1.4362 | 57 | 1.8733 | 2.3991 | 1.6876 |
| 13 | 1.7067 | 5.0808 | 2.4617 | 58 | 2.1976 | 4.2732 | 2.5669 |
| 14 | 2.0086 | 3.8532 | 3.1573 | 59 | 1.9050 | 2.0736 | 1.9350 |
| 15 | 2.0362 | 3.8763 | 3.6982 | 60 | 1.5251 | 6.0759 | 2.4210 |
| 16 | 1.6787 | 3.5999 | 2.3467 | 61 | 1.7716 | 0.8286 | 1.7164 |
| 17 | 2.0319 | 2.7304 | 3.7947 | 62 | 1.8951 | 1.6680 | 1.1195 |
| 18 | 1.7548 | 1.2384 | 0.7203 | 63 | 1.9745 | 2.6982 | 1.4327 |
| 19 | 1.9031 | 1.9106 | 1.6195 | 64 | 1.9486 | 1.4085 | 2.4803 |
| 20 | 2.0153 | 2.7606 | 2.7951 | 65 | 1.6608 | 2.6182 | 1.8849 |
| 21 | 1.5705 | 1.3187 | 1.3711 | 66 | 1.6585 | 3.6280 | 1.2719 |
| 22 | 1.7817 | 1.9157 | 1.3440 | 67 | 2.1636 | 0.2535 | 1.3685 |
| 23 | 1.6696 | 0.9155 | 0.7421 | 68 | 1.7241 | 1.7722 | 2.7329 |
| 24 | 2.0691 | 1.5532 | 1.2411 | 69 | 1.7064 | 5.7469 | 3.1251 |
| 25 | 2.1259 | 3.5769 | 3.1406 | 70 | 1.7558 | 0.5146 | 1.4417 |
| 26 | 1.6907 | 1.2791 | 1.5703 | 71 | 2.0121 | 4.0992 | 3.4287 |
| 27 | 2.0025 | 2.3081 | 1.9807 | 72 | 2.0239 | 2.3041 | 2.6732 |
| 28 | 1.8056 | 0.5816 | 0.7381 | 73 | 1.8900 | 6.0365 | 3.4701 |
| 29 | 1.6444 | 2.3904 | 1.2175 | 74 | 1.8122 | 1.1317 | 1.6745 |
| 30 | 2.0144 | 1.8385 | 2.2021 | 75 | 1.9149 | 1.5990 | 1.5343 |
| 31 | 1.5260 | 5.6012 | 3.0475 | 76 | 1.8685 | 0.6289 | 1.5335 |
| 32 | 1.8379 | 2.9933 | 1.8440 | 77 | 1.7135 | 1.2120 | 1.6727 |
| 33 | 1.8906 | 2.4768 | 2.5081 | 78 | 1.7870 | 1.0894 | 1.4135 |
| 34 | 1.8940 | 3.7551 | 1.6790 | 79 | 1.7331 | 0.7432 | 0.8510 |
| 35 | 1.7915 | 1.8526 | 2.0289 | 80 | 1.5097 | 1.7610 | 2.0634 |
| 36 | 2.0227 | 3.3827 | 2.2726 | 81 | 1.7795 | 0.0637 | 1.2769 |
| 37 | 2.1192 | 1.8604 | 1.7906 | 82 | 1.4753 | 1.7641 | 1.2034 |
| 38 | 1.9756 | 1.8860 | 2.2459 | 83 | 1.5775 | 2.8834 | 1.7849 |
| 39 | 1.7896 | 1.7319 | 1.4725 | 84 | 1.6220 | 3.5224 | 2.3445 |
| 40 | 1.9206 | 2.5398 | 2.7447 | 85 | 1.8122 | 1.6799 | 1.8171 |
| 41 | 1.8540 | 1.5533 | 1.5474 | 86 | 2.0188 | 2.5025 | 2.2536 |
| 42 | 1.9207 | 2.1761 | 2.8264 | 87 | 2.0721 | 1.3334 | 1.8721 |
| 43 | 1.8654 | 2.8219 | 2.6819 | 88 | 1.8996 | 2.1385 | 1.6345 |
| 44 | 1.8708 | 2.4153 | 0.6933 | 89 | 1.6954 | 6.1273 | 3.7291 |
| 45 | 1.9319 | 1.4443 | 0.9003 | 90 | 1.9719 | 3.3317 | 2.5255 |

Table 6.5: Values of the ninety Chinese characters in the **Legendre** moment three-dimensional feature space.

central moment technique. The results show that most of the representations of the Chinese characters in the Central moment feature space are crowded to a small area in the two-dimensional central moment feature plane. Therefore, in both cases, it is impossible to recognize those Chinese characters successfully.

We proposed some new **Legendre** moment feature spaces in this chapter. First, a two-dimensional **Legendre** moment feature space was developed and applied. The same two sets of Chinese characters are employed as the input patterns. For the set of five similar characters, the experiment demonstrated that all five representations in the **Legendre** moment feature space are well separated. The performance of recognizing ninety randomly selected Chinese characters with the **Legendre** moment feature space is much more refined than that of the central moment feature space as well. However, the distance in the two-dimensional **Legendre** moment feature space between two characters, $S_{44}$ and $S_{57}$, is quite small. This can be a potential problem in a full scale Chinese character recognition application.

To improve the recognizing ability, we added one new feature to the two-dimensional **Legendre** moment feature space. The new three-dimensional feature space is able to separate all ninety randomly selected Chinese characters easily.

It is noted that the highest order **Legendre** polynomials involved in the three-dimensional **Legendre** moment feature space is 3. With the development of the better VLSI moment generator chips[6], a hardware device for Chinese character recognition becomes possible.

Because of some technical reasons, we cannot obtain the whole set (more than 50,000) of Chinese characters and test all of them individually. However, with a possible fourth feature being added to the three-dimensional **Legendre** moment feature space, we are very optimistic to say that the **Legendre** moment technique can solve the Chinese character recognition problem.

With the discussions and the experimental results we had in this chapter, we are confident that feature spaces based on **Legendre** moments are the right direction to solve the practical Chinese character recognition problem.

# Chapter 7

# Conclusions and Recommendations

## 7.1 Conclusions

We have been concerned here with moment methods in image analysis. We found that a fundamental element of moment methods, accuracy in moment computing, had not attracted the attention it deserved. We have proposed and implemented several procedures to increase the accuracy in **Legendre** and **Zernike** moments computing.

Efforts made to reduce computing errors in **Legendre** moments turned out to be very successful. Primarily, we have solved the problem of computation errors related to the **Legendre** moment computing. Meanwhile, by working out up to order 55 **Legendre** polynomials, we reduced the moment computation time dramatically and made the utilization of higher order **Legendre** moments practically possible.

Based on these improvements, we performed image reconstruction via **Legendre** moments. We found that the reconstructed images were very close to the original image numerically and visually. The quality of reconstructed images is superior to all published results.

The computation errors of **Zernike** moments have been investigated as well. Because of the nature of the **Zernike** moments computing, there are two types of major errors, geometric and approximate, in the computation. Adopting the result from a classical problem in *Number Theory*, *The Lattice Points of a Circle*, we concluded that the geometric error in **Zernike** moment computing cannot be completely removed. We also proposed several procedures to reduce the approximate errors in **Zernike** moment computing. Though improvement has been obtained, none of them works flawlessly. We concluded that the lack of efficient measures to reduce both geometric and approximate errors effectively would impede further utilization of the **Zernike** moments.

Image reconstruction via **Zernike** moments was performed as well. Applying the best formula proposed, we reconstructed some images from their original versions with reasonable quality. The reconstructed images via **Zernike** moments indeed have better qualities than the results published previously, but, they are simply not as good as those images reconstructed via **Legendre** moments.

We have been also concerned here with reconstructing images from a finite set of moments computed from the noisy observed data. We conclude that there exists an optimal number of moments yielding the best possible representation of the original image without noise.

Finally, we discussed the recognition of Chinese characters via moment methods. We concluded that the method of **Legendre** moment works quite well for the Chinese character interpretation. Since the highest order **Legendre** polynomials involved in the Chinese character recognition task is 3, with the developments in the area of VLSI moment generator chips, a hardware device for Chinese character recognition becomes technically possible.

## 7.2 Recommendations

After reviewing the results from this research, we have a few recommendations for further study.

A visible extension of two-dimensional image reconstruction is the reconstruction task in three-dimensional space. Since the prime accuracy and efficiency problems of computing high order of **Legendre** moments have been solved in this thesis, there is no real technical difficulty for reconstructing a three-dimensional image via the **Legendre** moments.

Though we cannot reduce the geometric error in the **Zernike** moment computing effectively, we can, however, reduce the approximation error further by developing new formulas to calculate integrations for all pixels along the boundary of the unit circle. This could be a challenging task, but must be solved before the further full scale utilization of the **Zernike** moments.

Practically, we can build a database including all **Legendre** moment space features covering the whole Chinese character set without real technical difficulty. This will be the first important step to develop a reading machine for the Chinese language, which is one the most difficult languages in terms of artificial intelligence reading.

# Bibliography

[1] Y.S. Abu-Mostafa and D. Psaltis, *Recognitive aspects of moment invariants*, IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-6, pp. 698-706, Nov. 1984.

[2] Y.S. Abu-Mostafa and D. Psaltis, *Image normalization by complex moments*, IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-7, pp. 46-55, Jan. 1985.

[3] C.R. Adams and J.A. Clarkson, *Properties of functions f(x,y) of bounded variation*, Trans. of American Math. Soc. Vol. 36, pp. 711-730, 1934.

[4] B. Alpert and V. Rokhlin, A Fast Algorithm for the Evaluation of Legendre Expansions, Research Report YALEU/DCS/RR-671, January 1989.

[5] F.L. Alt, *Digital pattern recognition by moments*, J. Assoc. Computing Machinery, Vol. 9, pp. 240-258, 1962.

[6] R.L. Andersson, *Real-Time Gray-Scale Video Processing Using a Moment-Generating Chip*, IEEE Journal of Robotics and Automation, Vol. RA-1, No. 2, June 1985.

[7] B. Bamiech and R.J.P. De Figueiredo, *A general moment-invariants/attributed-graph method for three-dimensional object recognition from a single image*, IEEE Journal of Robotics and Automation, Vol. RA-2, pp. 31-41, 1986.

[8] D.E. Barton and F.N. David, *Randomisation bases for multivariate tests: I. The bivariate case, randomness of N points in a plane*, Bulletin of the International Statistical Institute, 37(1), pp. 158-159, (2), 455-467, 1962.

[9] S.O. Belkasim, M. Shridhar and M. Ahmadi, *Pattern Recognition With Moment Invariants: A Comparative Study And New Results*, Pattern Recognition, Vol. 24, No. 12, pp. 1117-1138, 1991.

[10] M. Bertero, C. DeMol and E.R. Pike, *Linear inverse problems with discrete data, I: general formulation and singular system analysis*, Inverse Problems, Vol. 1, pp. 301-330, 1985.

[11] M. Bertero, T.A. Poggio and V. Torre, *Ill-posed problems in early vision*, Proc. of the IEEE, Vol. 76, pp. 865-885, 1988.

[12] A.B. Bhatia and E. Wolf, *Proc. Camb. Phil. Soc.*, Vol. 50, pp. 40-48, 1954.

[13] M. Born and E. Wolf, *Principles of Optics*, Pergamon Press, Oxford, 1975.

[14] G.E.P. Box and D.R. Cox, *An analysis of transformations*, J.R. Stat. Soc. B26, pp. 211-246, 1964.

[15] J.F. Boyce and W.J. Hossack, *Moment invariants for pattern recognition*, Pattern Recognition Lett., Vol. 1, no. 5-6, pp. 451-456, July 1983.

[16] D. Casasent, *Advanced optical processors for multiple degree-of-freedom object recognition*, IEEE Trans. Aerospace and Electronic System, Vol. 24, pp. 608-618, 1988.

[17] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Vol. I. New York: Interscience, 1953.

[18] P.J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, New York, 1975.

[19] Stanley R. Deans, *The Radon Transform and Some of its Applications*, Wiley, New York, 1983.

[20] A. Devinatz, *Two Parameter Moment Problems*, Duke Math. J. vol. 24, 1957, pp. 481-498.

[21] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[22] K.J. Dudani, K.J. Breeding, and R.B. McGhee, *Aircraft identification by moment invariants*, IEEE rans. Comput. Vol. C-26, pp. 39-46, Jan. 1977.

[23] H. Engels, *Numerical quadrature and cubature*, London: Academic Press Inc., 1980.

[24] R.L. Eubank, *Spline Smoothing and Nonparametric Regression*, Marcel Dekker, New York, 1988.

[25] V.K. Govindan and A.P. Shivaprasad, *Character Recognition – A Review*, Patt. Recogn., Vol. 23, no. 7, pp. 671-683, 1990.

[26] F. Hausdorff, *Summationsmethoden und Momentfolgen.* I, Mathematische Zeitschrift, vol. 9(1921), pp. 75-109.

[27] A. Goshtaby, *Template matching in rotated images*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, pp. 338-344, 1985.

[28] P. Hall, *Comparison of two orthogonal series methods of estimating a density and its derivatives on an interval*, J. Multivariate Anal., Vol. 12, pp. 432-449, 1982.

[29] P. Hall and J. Koch, *On the Feasibility of Cross-validation in Image Analysis*, SIAM J. Appl. Math., Vol. 52, pp. 292-313, 1992.

[30] W. Härdle, P. Hall, and S. Marron, *How far are automatically chosen regression smoothing parameters from their optimum*, Journal of American Statistical Association, Vol. 83, pp. 86-101, 1988.

[31] G. H. Hardy, *On the Expression of a Number as the Sum of Two Squares*, Quart. J. of Math. Soc. (2), 15 (1916), pp. 192-213.

[32] G. H. Hardy and E. Landau, *The Lattice Points of a Circle*, Proc. Royal Soc. (A), 105 (1924), pp. 245-258.

[33] M. Hatamian, *A Real-time Two-dimensional Moment Generating Algorithm And Its Single Chip Implementation*, IEEE Trans. Acoust. Speech Signal Process. ASSP-34, 1986, pp. 99-126.

[34] T.H. Hildebrandt and L.J. Schoenberg, *On Linear Functional Operations and the Moment Problem for A Finite Interval in One or Several Dimensions*, Annals of Mathematics, ser 2, vol. 34(1933), pp. 317-328.

[35] B.K.P. Horn, *Robot Vision*, The MIT Press, 1986.

[36] M.K. Hu, *Pattern recognition by moment invariants, proc.* IRE 49, 1961, 1428.

[37] M.K. Hu, *Visual problem recognition by moment invariant*, IRE Trans. Inform. Theory, Vol. IT-8, pp. 179-187, Feb. 1962.

[38] Loo-Keng Hua, *The Lattice-points in a Circle*, Quart. J. of Math. (Oxford), 13 (1942), pp. 18-29.

[39] A. E. Ingham, *On Two Classical Lattice Point Problems*, Proc. Cambridge Phil. Soc. 36 (1940), pp. 131-138.

[40] H. Iwaniec and C. J. Mozzochi, *On the Divisor and Circle Problems*, Journal of Number Theory, 29 (1988), pp. 60-93.

[41] D. Jackson, *The Theory of Approximation*, Amer. Math. Soc., Colloq. Amer. Math. Soc., Providence, R.I., 1930.

[42] A.K. Jain, *Fundamentals of Digital Image Processing*, Englewood Cliffs, New Jersey, Prentice-Hall, 1989.

[43] X.Y. Jiang and H. Bunke, *Simple and Fast Computation of Moments*, Pattern Recognition, Vol. 24, No. 8, pp. 801-806, 1991.

[44] Behrooz Kamgar-Parsi and Behzad Kamgar-Parsi, *Evaluation of Quantization Error in Computer Vision*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 11, No.9, pp. 929-940, 1989.

[45] A. Khotanzad and Y.H. Hong, *Rotation Invariant Image Recognition Using Features Selected via a Systematic Method*, Pattern Recognition 23, 1990, pp. 1089-1101.

[46] A. Khotanzad and Y.H. Hong, *Invariant Image Recognition by Zernike Moments*, IEEE Trans. Pattern Anal. Mach. Intelligence PAMI-12, 1990, pp. 489-498.

[47] E. Landau, *Vorlesungen über Zahlentheorie* (Leipzig, 1927), bd. 2, 183-308.

[48] Ed. Henry J. Landau, *Moments in Mathematics*, American Mathematical Society, Providence, Rhode Island, 1987.

[49] Bing-Cheng Li and Jun Shen, *Fast Computation of Moment Invariants*, Pattern Recognition, Vol. 24, No. 8, pp. 807-813, 1991.

[50] C.L. Mallows, *Some Comments on $C_p$*, Technometrics, Vol. 15, No. 4, pp. 661-675, Nov. 1973.

[51] K.V. Mardia and T.J. Hainsworth, Statistical aspects of moment invariants in image analysis, Journal of Applied Statistics, Vol. 16, No.3, 1989.

[52] A. D. Myskis, *Advanced Mathematics for Engineers*, MIR, Moscow, 1975.

[53] Sankar K. Pal and Dwijesh K. Dutta Majumder, *Fuzzy – Mathematical Approach To Pattern Recognition*, John Wiley & sons, New Delhi, 1986.

[54] Rallis C. Papademetriou, *Reconstructing with Moments*, 11th IAPR International Conference On Pattern Recognition, pp. 476-480, Aug.30 - Sept.3, 1992.

[55] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 1984.

[56] M. Pawlak, *On the reconstruction aspects of moment descriptions*, IEEE Symposium on Information Theory, San Diego, January 1990.

[57] M. Pawlak, *On the reconstruction aspects of moment descriptors*, IEEE Trans. Information Theory, Vol. 38, No. 6, pp. 1698-1708, November, 1992.

[58] M. Pawlak and X. Liao, *On Image Analysis via Orthogonal Moments*, Vision Interface '92, pp. 253-258, May, 1992.

[59] M. Pawlak and X. Liao, *On Image Analysis By Orthogonal Moments*, 11th IAPR International Conference On Pattern Recognition, pp. 549-552, Aug.30 - Sept.3, 1992.

[60] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, 1988.

[61] Kendall Preston, Jr. , *The Abingdon Cross Benchmark Survey*, Computer, IEEE Computer Society, July, 1989.

[62] R.J. Prokop and A.P. Reeves, *A Survey of Moment-Based Techniques for Unoccluded Object Representation and Recognition*, Graphical Models And Image Processing, Vol. 54, No. 5, September, pp. 438-460, 1992.

[63] S.S. Reddi, *Radial and angular moment invariants for image identification*, IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-3, pp. 240-4, Mar. 1981.

[64] A.P. Reeves, *A Parallel Mesh Moment Computer*, Proceedings of the 6th International Conference on Pattern Recognition, October 1982, pp. 465-467.

[65] A.P. Reeves, R.J. Prokop, S.E. Andrews and F.P. Kuhl, *Three-Dimensional Shape Analysis Using Moments and Fourier Descriptors*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 10, No. 6, pp. 937-943, November 1988.

[66] T.H. Reiss, *The Revised Fundamental Theorem of Moment Invariants*, IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-13, No. 8, August 1991, pp. 830-834.

[67] F.A. Sadjadi and E.L. Hall, *Three-dimensional moment invariants*, IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-2, pp. 127-136, Mar. 1980.

[68] G. Sansone, *Orthogonal Functions*, Dover Publications, Inc., New York, 1991.

[69] F.W. Smith and M.H. Wright, *Automatic ship photo interpretation by the method of moments*, IEEE Trans. Comput., Vol. C-20, pp. 1089-1094, Sept. 1971.

[70] G. Szegö, *Orthogonal Polynomials*, Vol 23, 4th Ed., American Mathematical Society Colloquium Publications, Providence, R.I., 1975.

[71] G. Talenti, *Recovering a function from a finite number of moments*, Inverse Problems, Vol. 3, pp. 501-517, 1987.

[72] R.W. Taylor and A.P. Reeves, *Three-dimensional image transforms in moment space*, in Proceedings of the IEEE Computer Society Workshop on Computer Vision, 1987, pp. 366-368.

[73] M.R. Teague, *Image analysis via the general theory of moments*, J. Optical Soc. Am., Vol. 70, pp. 920-930, August 1980.

[74] C.H. Teh and R.T. Chin, *On digital approximation of moment invariants*, Computer Vision, Graphics, and Image Processing, Vol. 33, pp. 318-326, 1986.

[75] C.H. Teh and R.T. Chin, *On image analysis by the methods of moments*, IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-10, pp. 496-512, July 1988.

[76] E. C. Titchmarsh, *The Lattice-points in a Circle*, Proc. London Math. Soc. (2), 38 (1935), pp. 96-115.

[77] J. R. Wilton, *The Lattice Points of a Circle: an Historical Account of the Problem*, Messenger of Math. 58 (1929), pp. 67-80.

[78] Yu. V. Vorobyev, *Method of Moments in Applied Mathematics*, New York, Gordon and Breach Science Publishers, 1965.

[79] F. Zernike, *Physica*, Vol. 1, p. 689, 1934.

[80] X. Zhou, *Digital Subtraction Mannography via Geometric Unwarping for Detection of Early Breast Cancer*, Ph.D. dissertation, Thesis Z4946, The University of Manitoba, 1991.

# Appendix A

**Some of the Higher Order Legendre Polynomials:**

$$\begin{aligned}P_{10}(x) &= \frac{1}{2^{10}}(-252 + 13,860x^2 - 120,120x^4 + 360,360x^6 \\ &\quad -437,580x^8 + 184,756x^{10}) \qquad (A.1)\end{aligned}$$

$$\begin{aligned}P_{11}(x) &= \frac{x}{2^{11}}(-5,544 + 120,120x^2 - 720,720x^4 + 1,750,320x^6 \\ &\quad -1,847,560x^8 + 705,432x^{10}) \qquad (A.2)\end{aligned}$$

$$\begin{aligned}P_{12}(x) &= \frac{1}{2^{12}}(924 - 72,072x^2 + 900,900x^4 - 4,084,080x^6 \\ &\quad +8,314,020x^8 - 7,759,752x^{10} + 2,704,156x^{12}) \qquad (A.3)\end{aligned}$$

$$\begin{aligned}P_{13}(x) &= \frac{x}{2^{13}}(24,024 - 720,720x^2 + 6,126,120x^4 \\ &\quad -22,170,720x^6 + 38,798,760x^8 - 32,449,872x^{10} \\ &\quad +10,400,600x^{12}) \qquad (A.4)\end{aligned}$$

$$\begin{aligned}P_{14}(x) &= \frac{1}{2^{14}}(-3,432 + 360,360x^2 - 6,126,120x^4 \\ &\quad +38,798,760x^6 - 116,396,280x^8 + 178,474,296x^{10} \\ &\quad -135,207,800x^{12} + 40,116,600x^{14}) \qquad (A.5)\end{aligned}$$

$$\begin{aligned}P_{15}(x) &= \frac{x}{2^{15}}(-102,960 + 4,084,080x^2 - 46,558,512x^4 \\ &\quad +232,792,560x^6 - 594,914,320x^8 + 811,246,800x^{10} \\ &\quad -561,632,400x^{12} + 155,117,520x^{14}) \qquad (A.6)\end{aligned}$$

$$\begin{aligned}P_{16}(x) &= \frac{1}{2^{16}}(12,870 - 1,750,320x^2 + 38,798,760x^4 - 325,909,584x^6 \\ &\quad +1,338,557,220x^8 - 2,974,571,600x^{10} + 3,650,610,600x^{12} \\ &\quad -2,326,762,800x^{14} + 601,080,390x^{16})\end{aligned} \tag{A.7}$$

$$\begin{aligned}P_{17}(x) &= \frac{x}{2^{17}}(437,580 - 22,170,720x^2 + 325,909,584x^4 \\ &\quad -2,141,691,552x^6 + 7,436,429,000x^8 \\ &\quad -14,602,442,400x^{10} + 16,287,339,600x^{12} \\ &\quad -9,617,286,240x^{14} + 2,333,606,220x^{16}\end{aligned} \tag{A.8}$$

$$\begin{aligned}P_{18}(x) &= \frac{1}{2^{18}}(-48,620 + 8,314,020x^2 - 232,792,560x^4 \\ &\quad +2,498,640,144x^6 - 13,385,572,200x^8 \\ &\quad +40,156,716,600x^{10} - 70,578,471,600x^{12} \\ &\quad +72,129,646,800x^{14} - 39,671,305,740x^{16} \\ &\quad +9,075,135,300x^{18})\end{aligned} \tag{A.9}$$

$$\begin{aligned}P_{19}(x) &= \frac{x}{2^{19}}(-1,847,560 + 116,396,280x^2 - 2,141,691,552x^4 \\ &\quad +17,847,429,600x^6 - 80,313,433,200x^8 \\ &\quad +211,735,414,800x^{10} - 336,605,018,400x^{12} \\ &\quad +317,370,445,920x^{14} - 163,352,435,400x^{16} \\ &\quad +35,345,263,800x^{18})\end{aligned} \tag{A.10}$$

$$\begin{aligned}P_{20}(x) &= \frac{1}{2^{20}}(184,756 - 38,798,760x^2 + 1,338,557,220x^4 \\ &\quad - 17,847,429,600x^6 + 120,470,149,800x^8 \\ &\quad - 465,817,912,560x^{10} + 1,093,966,309,800x^{12} \\ &\quad - 1,586,852,229,600x^{14} + 1,388,495,700,900x^{16} \\ &\quad - 671,560,012,200x^{18} + 137,846,528,820x^{20})\end{aligned} \tag{A.11}$$

$$
\begin{aligned}
P_{21}(x) \;=\; & \frac{x}{2^{21}}(7,759,752-594,914,320x^2+13,385,572,200x^4 \\
& -137,680,171,200x^6+776,363,187,600x^8 \\
& -2,625,519,143,520x^{10}+5,553,982,803,600x^{12} \\
& -7,405,310,404,800x^{14}+6,044,040,109,800x^{16} \\
& -2,756,930,576,400x^{18}+538,257,874,440x^{20}
\end{aligned} \tag{A.12}
$$

$$
\begin{aligned}
P_{22}(x) \;=\; & \frac{1}{2^{22}}(-705,432+178,474,296x^2-7,436,429,000x^4 \\
& +120,470,149,800x^6-998,181,241,200x^8 \\
& +4,813,451,763,120x^{10}-14,440,355,289,360x^{12} \\
& +27,769,914,018,000x^{14}-34,249,560,622,200x^{16} \\
& +26,190,840,475,800x^{18}-11,303,415,363,240x^{20} \\
& +2,104,098,963,720x^{22})
\end{aligned} \tag{A.13}
$$

$$
\begin{aligned}
P_{23}(x) \;=\; & \frac{x}{2^{23}}(-32,449,872+2,974,571,600x^2-80,313,433,200x^4 \\
& +998,181,241,200x^6-6,876,359,661,600x^8 \\
& +28,880,710,578,720x^{10}-77,755,759,250,400x^{12} \\
& +136,998,242,488,800x^{14}-157,145,042,854,800x^{16} \\
& +113,034,153,632,400x^{18}-46,290,177,201,840x^{20} \\
& +8,233,430,727,600x^{22})
\end{aligned} \tag{A.14}
$$

$$
\begin{aligned}
P_{24}(x) \;=\; & \frac{1}{2^{24}}(2,704,156-811,246,800x^2+40,156,716,600x^4 \\
& -776,363,187,600x^6+7,735,904,619,300x^8 \\
& -45,383,973,766,560x^{10}+168,470,811,709,200x^{12} \\
& -410,994,727,466,400x^{14}+667,866,432,132,900x^{16} \\
& -715,882,973,005,200x^{18}+486,046,860,619,320x^{20} \\
& -189,368,906,734,800x^{22}+32,247,603,683,100x^{24})
\end{aligned} \tag{A.15}
$$

$$\begin{aligned}P_{25}(x) \;=\;& \frac{x}{2^{25}}(135,207,800 - 14,602,442,400x^2 + 465,817,912,560x^4\\ & -6,876,359,661,600x^6 + 56,729,967,208,200x^8\\ & -288,807,105,787,200x^{10} + 958,987,697,208,200x^{12}\\ & -2,137,172,582,825,280x^{14} + 3,221,473,378,523,400x^{16}\\ & -3,240,312,404,128,800x^{18} + 2,083,057,974,082,800x^{20}\\ & -773,942,488,394,400x^{22} + 126,410,606,437,752x^{24})\end{aligned} \tag{A.16}$$

$$\begin{aligned}P_{26}(x) \;=\;& \frac{1}{2^{26}}(-10,400,600 + 3,650,610,600x^2 - 211,735,414,800x^4\\ & +4,813,451,763,120x^6 - 56,729,967,208,200x^8\\ & +397,109,770,457,400x^{10} - 1,780,977,152,354,400x^{12}\\ & +5,342,931,457,063,200x^{14} - 10,953,009,486,979,560x^{16}\\ & +15,391,483,919,611,800x^{18} - 14,581,405,818,579,600x^{20}\\ & +8,900,338,616,535,600x^{22} - 3,160,265,160,943,800x^{24}\\ & +495,918,532,948,104x^{26})\end{aligned} \tag{A.17}$$

$$\begin{aligned}P_{27}(x) \;=\;& \frac{x}{2^{27}}(-561,632,400 + 70,578,471,600x^2 - 2,625,519,143,520x^4\\ & +45,383,973,766,560x^6 - 441,233,078,286,000x^8\\ & +2,671,465,728,531,600x^{10} - 10,685,862,914,126,400x^{12}\\ & +29,208,025,298,612,160x^{14} - 55,409,342,110,602,480x^{16}\\ & +72,907,029,092,898,000x^{18} - 65,269,149,854,594,400x^{20}\\ & +37,923,181,931,325,600x^{22} - 12,893,881,856,650,704x^{24}\\ & +1,946,939,425,648,112x^{26})\end{aligned} \tag{A.18}$$

$$\begin{aligned}P_{28}(x) \;=\;& \frac{1}{2^{28}}(40,116,600 - 16,287,339,600x^2 + 1,093,966,309,800x^4\\ & -28,880,710,578,720x^6 + 397,109,770,457,400x^8\\ & -3,265,124,779,316,400x^{10} + 17,364,527,235,455,400x^{12}\\ & -62,588,625,639,883,200x^{14} + 156,993,135,980,040,352x^{16}\\ & -277,046,710,553,012,416x^{18} + 342,663,036,736,620,608x^{20}\\ & -290,744,394,806,829,568x^{22} + 161,173,523,208,133,792x^{24}\\ & -52,567,364,492,499,024x^{26}\\ & +7,648,690,600,760,440x^{28})\end{aligned} \tag{A.19}$$

$$\begin{aligned} P_{29}(x) &= \frac{x}{2^{29}}(2,326,762,800 - 336,605,018,400x^2 + 14,440,355,289,360x^4 \\ &\quad -288,807,105,787,200x^6 + 3,265,124,779,316,400x^8 \\ &\quad -23,152,702,980,607,200x^{10} + 109,530,094,869,795,600x^{12} \\ &\quad -358,841,453,668,663,680x^{14} + 831,140,131,659,037,184x^{16} \\ &\quad -1,370,652,146,946,482,430x^{18} + 1,599,094,171,437,562,880x^{20} \\ &\quad -1,289,388,185,665,070,080x^{22} + 683,375,738,402,487,296x^{24} \\ &\quad -214,163,336,821,292,320x^{26} + 30,067,266,499,541,040x^{28}) \end{aligned} \tag{A.20}$$

$$\begin{aligned} P_{30}(x) &= \frac{1}{2^{30}}(-155,117,520 + 72,129,646,800x^2 - 5,553,982,803,600x^4 \\ &\quad +168,470,811,709,200x^6 - 2,671,465,728,531,600x^8 \\ &\quad +25,467,973,278,667,920x^{10} - 158,210,137,034,149,216x^{12} \\ &\quad +672,827,725,628,744,320x^{14} - 2,018,483,176,886,233,340x^{16} \\ &\quad +4,340,398,465,330,527,740x^{18} - 6,716,195,520,037,764,100x^{20} \\ &\quad +7,413,982,067,574,155,260x^{22} - 5,694,797,820,020,725,760x^{24} \\ &\quad +2,891,205,047,087,446,530x^{26} - 871,950,728,486,690,176x^{28} \\ &\quad +118,264,581,564,861,424x^{30}) \end{aligned} \tag{A.21}$$

$$\begin{aligned} P_{31}(x) &= \frac{x}{2^{31}}(-9,617,286,240 + 1,586,852,229,600x^2 - 77,755,759,250,400x^4 \\ &\quad +1,780,977,152,354,400x^6 - 23,152,702,980,607,200x^8 \\ &\quad +189,852,164,440,979,040x^{10} - 1,046,620,906,533,602,430x^{12} \\ &\quad +4,036,966,353,772,466,180x^{14} - 11,161,024,625,135,642,600x^{16} \\ &\quad +22,387,318,400,125,882,400x^{18} - 32,621,521,097,326,284,800x^{20} \\ &\quad +34,168,786,920,124,362,800x^{22} - 25,057,110,408,091,193,300x^{24} \\ &\quad +12,207,310,198,813,663,200x^{26} - 3,547,937,446,945,843,200x^{28} \\ &\quad +465,428,353,255,261,056x^{30}) \end{aligned} \tag{A.22}$$

$$
\begin{aligned}
P_{32}(x) &= \frac{1}{2^{32}}(601,080,390 - 317,370,445,920x^2 \\
&\quad +27,769,914,018,000x^4 - 958,987,697,421,600x^6 \\
&\quad +17,364,527,235,455,400x^8 - 189,852,164,440,979,040x^{10} \\
&\quad +1,360,607,178,493,683,200x^{12} - 6,728,277,256,287,443,970x^{14} \\
&\quad +23,717,177,328,413,241,300x^{16} - 60,765,578,514,627,387,400x^{18} \\
&\quad +114,175,323,840,641,991,000x^{20} - 157,176,419,832,572,084,000x^{22} \\
&\quad +156,606,940,050,569,986,000x^{24} - 109,865,791,789,322,928,000x^{26} \\
&\quad +51,445,092,980,714,725,400x^{28} - 14,428,278,950,913,095,700x^{30} \\
&\quad +1,832,624,140,942,590,460x^{32}) \qquad (A.23)
\end{aligned}
$$

$$
\begin{aligned}
P_{33}(x) &= \frac{x}{2^{33}}(39,671,305,740 - 7,405,310,404,800x^2 \\
&\quad +410,994,727,466,400x^4 - 10,685,862,914,126,400x^6 \\
&\quad +158,210,137,034,149,216x^8 - 1,484,298,740,174,926,850x^{10} \\
&\quad +9,419,588,158,802,421,760x^{12} - 42,163,870,806,067,986,400x^{14} \\
&\quad +136,722,551,657,911,632,000x^{16} - 326,215,210,973,262,774,000x^{18} \\
&\quad +576,313,539,386,097,664,000x^{20} - 751,713,312,242,736,038,000x^{22} \\
&\quad +714,127,646,630,599,197,000x^{24} - 480,154,201,153,337,229,000x^{26} \\
&\quad +216,424,184,263,696,417,000x^{28} - 58,643,972,510,162,903,000x^{30} \\
&\quad +7,219,428,434,016,265,220x^{32}) \qquad (A.24)
\end{aligned}
$$

$$
\begin{aligned}
P_{34}(x) &= \frac{1}{2^{34}}(-2,333,606,220 + 1,388,495,700,900x^2 \\
&\quad -136,998,242,488,800x^4 + 5,342,931,457,063,200x^6 \\
&\quad -109,530,094,869,795,600x^8 + 1,360,607,178,493,683,200x^{10} \\
&\quad -11,132,240,551,311,951,900x^{12} + 63,245,806,209,101,971,500x^{14} \\
&\quad -258,253,708,687,166,407,000x^{16} + 774,761,126,061,499,220,000x^{18} \\
&\quad -1,728,940,618,158,292,860,000x^{20} \\
&\quad +2,881,567,696,930,488,710,000x^{22} \\
&\quad -3,570,638,233,152,996,250,000x^{24} \\
&\quad +3,241,040,857,785,026,740,000x^{26} \\
&\quad -2,092,100,447,882,397,880,000x^{28} \\
&\quad +908,981,573,907,525,009,000x^{30}
\end{aligned}
$$

$$-238,241,138,322,536,792,000x^{32}$$
$$+28,453,041,475,240,575,000x^{34}) \tag{A.25}$$

$$
\begin{aligned}
P_{35}(x) \;=\; & \frac{x}{2^{35}}(-163,352,435,400 + 34,249,560,622,200x^2 \\
& -2,137,172,582,825,280x^4 \\
& +62,588,625,639,883,200x^6 \\
& -1,046,620,906,533,602,430x^8 \\
& +11,132,240,551,311,953,900x^{10} \\
& -80,494,662,447,947,956,200x^{12} \\
& +413,205,933,899,466,277,000x^{14} \\
& -1,549,522,252,122,998,440,000x^{16} \\
& +4,322,351,545,395,732,020,000x^{18} \\
& -9,056,355,618,924,391,300,000x^{20} \\
& +14,282,552,932,611,987,100,000x^{22} \\
& -16,853,412,460,482,141,400,000x^{24} \\
& +14,644,703,135,176,788,500,000x^{26} \\
& -9,089,815,739,075,246,690,000x^{28} \\
& +3,811,858,213,160,589,200,000x^{30} \\
& -967,403,410,158,179,713,000x^{32} \\
& +112,186,277,816,662,835,000x^{34})
\end{aligned}
\tag{A.26}
$$

$$
\begin{aligned}
P_{36}(x) \;=\; & \frac{1}{2^{36}}(9,075,135,300 - 6,044,040,109,800x^2 \\
& +667,866,432,132,900x^4 \\
& -29,208,025,298,612,160x^6 \\
& +672,827,725,628,744,320x^8 \\
& -9,419,588,158,802,421,760x^{10} \\
& +87,202,550,985,276,964,900x^{12} \\
& -563,462,637,135,635,743,000x^{14} \\
& +2,634,187,828,609,097,400,000x^{16} \\
& -9,124,964,373,613,211,810,000x^{18} \\
& +23,772,933,499,676,526,600,000x^{20} \\
& -46,928,388,207,153,669,700,000x^{22} \\
& +70,222,551,918,675,599,800,000x^{24}
\end{aligned}
$$

$$-79,081,396,929,954,656,000,000x^{26}$$
$$+65,901,164,108,295,541,100,000x^{28}$$
$$-39,389,201,535,992,739,100,000x^{30}$$
$$+15,962,156,267,609,966,800,000x^{32}$$
$$-3,926,519,723,583,200,030,000x^{34}$$
$$+442,512,540,276,836,729,000x^{36}) \qquad \text{(A.27)}$$

$$P_{37}(x) \;=\; \frac{x}{2^{37}}(671,560,012,200 - 157,145,042,854,800x^2$$
$$+10,953,009,486,979,560x^4$$
$$-358,841,453,668,663,680x^6$$
$$+6,728,277,256,287,442,940x^8$$
$$-80,494,662,447,947,972,600x^{10}$$
$$+657,373,076,658,241,667,000x^{12}$$
$$-3,831,545,932,522,323,440,000x^{14}$$
$$+16,424,935,872,503,784,400,000x^{16}$$
$$-52,828,741,110,392,277,700,000x^{18}$$
$$+129,053,067,569,672,562,000,000x^{20}$$
$$-240,763,035,149,744,904,000,000x^{22}$$
$$+342,686,053,363,136,921,000,000x^{24}$$
$$-369,046,519,006,455,017,000,000x^{26}$$
$$+295,419,011,519,945,493,000,000x^{28}$$
$$-170,263,000,187,839,590,000,000x^{30}$$
$$+66,750,835,300,914,406,900,000x^{32}$$
$$-15,930,451,449,966,124,600,000x^{34}$$
$$+1,746,130,564,335,625,830,000x^{36}) \qquad \text{(A.28)}$$

$$P_{38}(x) \;=\; \frac{1}{2^{38}}(-35,345,263,800 + 26,190,840,475,800x^2$$
$$-3,221,473,378,523,400x^4$$
$$+156,993,135,980,040,352x^6$$
$$-4,036,966,353,772,466,180x^8$$
$$+63,245,806,209,101,971,500x^{10}$$
$$-657,373,076,658,241,798,000x^{12}$$
$$+4,789,432,415,652,904,170,000x^{14}$$

$$
\begin{aligned}
&-25,383,991,802,960,394,800,000x^{16}\\
&+100,374,608,109,745,339,000,000x^{18}\\
&-301,123,824,329,235,951,000,000x^{20}\\
&+692,193,726,055,516,515,000,000x^{22}\\
&-1,223,878,762,011,203,180,000,000x^{24}\\
&+1,660,709,335,529,048,210,000,000x^{26}\\
&-1,713,430,266,815,683,870,000,000x^{28}\\
&+1,319,538,251,455,756,680,000,000x^{30}\\
&-734,259,188,310,058,342,000,000x^{32}\\
&+278,782,900,374,407,213,000,000x^{34}\\
&-64,606,830,880,418,168,800,000x^{36}\\
&+6,892,620,648,693,259,830,000x^{38}) \qquad \text{(A.29)}
\end{aligned}
$$

$$
\begin{aligned}
P_{39}(x) \;=\; &\frac{x}{2^{39}}(-2,756,930,576,400 + 715,882,973,005,200x^2\\
&-55,409,342,110,602,480x^4\\
&+2,018,483,176,886,233,340x^6\\
&-42,163,870,806,067,978,200x^8\\
&+563,462,637,135,635,808,000x^{10}\\
&-5,157,850,293,780,051,130,000x^{12}\\
&+33,845,322,403,947,187,500,000x^{14}\\
&-164,249,358,725,037,852,000,000x^{16}\\
&+602,247,648,658,471,969,000,000x^{18}\\
&-1,692,029,108,135,706,680,000,000x^{20}\\
&+3,671,636,286,033,609,280,000,000x^{22}\\
&-6,168,348,960,536,463,640,000,000x^{24}\\
&+7,996,007,911,806,528,290,000,000x^{26}\\
&-7,917,229,508,734,538,990,000,000x^{28}\\
&+5,874,073,506,480,465,660,000,000x^{30}\\
&-3,159,539,537,576,614,990,000,000x^{32}\\
&+1,162,922,955,847,527,090,000,000x^{34}\\
&-261,919,584,650,343,915,000,000x^{36}\\
&+27,217,014,869,199,027,200,000x^{38}) \qquad \text{(A.30)}
\end{aligned}
$$

$$
\begin{aligned}
P_{40}(x) \;=\; & \frac{1}{2^{40}}(137,846,528,820 - 113,034,153,632,400x^2 \\
& +15,391,483,919,611,800x^4 \\
& -831,140,131,659,037,184x^6 \\
& +23,717,177,328,413,241,300x^8 \\
& -413,205,933,899,466,211,000x^{10} \\
& +4,789,432,415,652,904,170,000x^{12} \\
& -39,052,295,081,477,527,300,000x^{14} \\
& +232,686,591,527,136,918,000,000x^{16} \\
& -1,040,245,938,591,906,420,000,000x^{18} \\
& +3,553,261,127,084,984,750,000,000x^{20} \\
& -9,383,070,508,752,555,690,000,000x^{22} \\
& +19,276,090,501,676,449,700,000,000x^{24} \\
& -30,841,744,802,682,320,300,000,000x^{26} \\
& +38,266,609,292,216,954,400,000,000x^{28} \\
& -36,419,255,740,178,880,900,000,000x^{30} \\
& +26,066,201,185,007,068,400,000,000x^{32} \\
& -13,567,434,484,887,818,900,000,000x^{34} \\
& +4,845,512,316,031,362,800,000,000x^{36} \\
& -1,061,463,579,898,762,170,000,000x^{38} \\
& +107,507,208,733,336,168,000,000x^{40}) \qquad \text{(A.31)}
\end{aligned}
$$

$$
\begin{aligned}
P_{41}(x) \;=\; & \frac{x}{2^{41}}(11,303,415,363,240 - 3,240,312,404,128,800x^2 \\
& +277,046,710,553,012,384x^4 \\
& -11,161,024,625,135,642,600x^6 \\
& +258,253,708,687,166,374,000x^8 \\
& -3,831,545,932,522,323,440,000x^{10} \\
& +39,052,295,081,477,527,300,000x^{12} \\
& -286,383,497,264,168,528,000,000x^{14} \\
& +1,560,368,907,887,859,500,000,000x^{16} \\
& -6,460,474,776,518,154,680,000,000x^{18} \\
& +20,642,755,119,255,622,900,000,000x^{20} \\
& -51,402,908,004,470,526,800,000,000x^{22}
\end{aligned}
$$

$$\begin{aligned}
&+100,235,670,608,717,544,000,000,000x^{24}\\
&-153,066,437,168,867,818,000,000,000x^{26}\\
&+182,096,278,700,894,460,000,000,000x^{28}\\
&-166,823,687,584,045,204,000,000,000x^{30}\\
&+115,323,193,121,546,422,000,000,000x^{32}\\
&-58,146,147,792,376,366,500,000,000x^{34}\\
&+20,167,808,018,076,481,700,000,000x^{36}\\
&-4,300,288,349,333,446,860,000,000x^{38}\\
&+424,784,580,848,791,688,000,000x^{40}) \qquad \text{(A.32)}
\end{aligned}$$

$$\begin{aligned}
P_{42}(x) \;=\; & \frac{1}{2^{42}}(-538,257,874,440+486,046,860,619,320x^2\\
&-72,907,029,092,898,000x^4\\
&+4,340,398,465,330,527,230x^6\\
&-136,722,551,657,911,632,000x^8\\
&+2,634,187,828,609,096,880,000x^{10}\\
&-33,845,322,403,947,191,700,000x^{12}\\
&+306,839,461,354,466,311,000,000x^{14}\\
&-2,040,482,418,007,200,760,000,000x^{16}\\
&+10,229,085,062,820,412,100,000,000x^{18}\\
&-39,408,896,136,760,742,900,000,000x^{20}\\
&+118,226,688,410,282,212,000,000,000x^{22}\\
&-278,432,418,357,548,692,000,000,000x^{24}\\
&+516,599,225,444,928,880,000,000,000x^{26}\\
&-754,398,868,903,705,753,000,000,000x^{28}\\
&+861,922,385,850,900,465,000,000,000x^{30}\\
&-761,133,074,602,206,235,000,000,000x^{32}\\
&+508,778,793,183,293,040,000,000,000x^{34}\\
&-248,736,298,889,610,027,000,000,000x^{36}\\
&+83,855,622,812,002,211,500,000,000x^{38}\\
&-17,416,167,814,800,461,300,000,000x^{40}\\
&+1,678,910,486,211,891,090,000,000x^{42}) \qquad \text{(A.33)}
\end{aligned}$$

$$
\begin{aligned}
P_{43}(x) &= \frac{x}{2^{43}}(-46,290,177,201,840 + 14,581,405,818,579,600x^2 \\
&\quad -1,370,652,146,946,482,180x^4 + 60,765,578,514,627,387,400x^6 \\
&\quad -1,549,522,252,122,998,180,000x^8 \\
&\quad +25,383,991,802,960,390,700,000x^{10} \\
&\quad -286,383,497,264,168,528,000,000x^{12} \\
&\quad +2,331,979,906,293,943,720,000,000x^{14} \\
&\quad -14,163,348,548,520,570,100,000,000x^{16} \\
&\quad +65,681,493,561,267,904,800,000,000x^{18} \\
&\quad -236,453,376,820,564,423,000,000,000x^{20} \\
&\quad +668,237,804,058,116,779,000,000,000x^{22} \\
&\quad -1,492,397,762,396,461,040,000,000,000x^{24} \\
&\quad +2,640,396,041,162,969,720,000,000,000x^{26} \\
&\quad -3,693,953,082,218,145,160,000,000,000x^{28} \\
&\quad +4,059,376,397,878,433,990,000,000,000x^{30} \\
&\quad -3,459,695,793,646,391,930,000,000,000x^{32} \\
&\quad +2,238,626,690,006,489,350,000,000,000x^{34} \\
&\quad -1,062,171,222,285,361,750,000,000,000x^{36} \\
&\quad +348,323,356,296,009,200,000,000,000x^{38} \\
&\quad -70,514,240,420,899,425,100,000,000x^{40} \\
&\quad +6,637,553,085,023,755,350,000,000x^{42}) \qquad \text{(A.34)}
\end{aligned}
$$

$$
\begin{aligned}
P_{44}(x) &= \frac{1}{2^{44}}(2,104,098,963,720 - 2,083,057,974,082,800x^2 \\
&\quad +342,663,036,736,620,608x^4 - 22,387,318,400,125,878,300x^6 \\
&\quad +774,761,126,061,499,220,000x^8 \\
&\quad -16,424,935,872,503,780,200,000x^{10} \\
&\quad +232,686,591,527,136,918,000,000x^{12} \\
&\quad -2,331,979,906,293,943,990,000,000x^{14} \\
&\quad +17,198,351,808,917,834,700,000,000x^{16} \\
&\quad -95,996,029,051,083,864,900,000,000x^{18} \\
&\quad +413,793,409,435,987,826,000,000,000x^{20} \\
&\quad -1,397,224,499,394,244,390,000,000,000x^{22} \\
&\quad +3,730,994,405,991,152,190,000,000,000x^{24}
\end{aligned}
$$

$$-7,921,188,123,488,908,620,000,000,000x^{26}$$
$$+13,390,579,923,040,775,000,000,000,000x^{28}$$
$$-17,977,238,333,461,640,300,000,000,000x^{30}$$
$$+19,028,326,865,055,158,600,000,000,000x^{32}$$
$$-15,670,386,830,045,422,700,000,000,000x^{34}$$
$$+9,825,083,806,139,592,110,000,000,000x^{36}$$
$$-4,528,203,631,848,121,180,000,000,000x^{38}$$
$$+1,445,541,928,628,438,290,000,000,000x^{40}$$
$$-285,414,782,656,021,501,000,000,000x^{42}$$
$$+26,248,505,381,684,852,100,000,000x^{44}) \qquad (A.35)$$

$$P_{45}(x) = \frac{x}{2^{45}}(189,368,906,734,800 - 65,269,149,854,594,400x^2$$
$$+6,716,195,520,037,763,070x^4$$
$$-326,215,210,973,262,840,000x^6$$
$$+9,124,964,373,613,211,810,000x^8$$
$$-164,249,358,725,037,819,000,000x^{10}$$
$$+2,040,482,418,007,200,760,000,000x^{12}$$
$$-18,344,908,596,179,025,800,000,000x^{14}$$
$$+123,423,465,922,822,110,000,000,000x^{16}$$
$$-636,605,245,286,135,075,000,000,000x^{18}$$
$$+2,561,578,248,889,448,270,000,000,000x^{20}$$
$$-8,140,351,431,253,423,560,000,000,000x^{22}$$
$$+20,595,089,121,071,162,800,000,000,000x^{24}$$
$$-41,659,581,982,793,521,200,000,000,000x^{26}$$
$$+67,414,643,750,481,143,500,000,000,000x^{28}$$
$$-86,986,637,097,395,033,500,000,000,000x^{30}$$
$$+88,798,858,703,590,743,200,000,000,000x^{32}$$
$$-70,740,603,404,205,058,800,000,000,000x^{34}$$
$$+43,017,934,502,557,133,300,000,000,000x^{36}$$
$$-19,273,892,381,712,515,700,000,000,000x^{38}$$
$$+5,993,710,435,776,452,040,000,000,000x^{40}$$
$$-1,154,934,236,794,133,580,000,000,000x^{42}$$
$$+103,827,421,287,553,420,000,000,000x^{44}) \qquad (A.36)$$

$$
\begin{aligned}
P_{46}(x) \;=\; & \frac{1}{2^{46}}(-8,233,430,727,600 + 8,900,338,616,535,600x^2 \\
& -1,599,094,171,437,562,880x^4 \\
& +114,175,323,840,641,974,000x^6 \\
& -4,322,351,545,395,732,550,000x^8 \\
& +100,374,608,109,745,323,000,000x^{10} \\
& -1,560,368,907,887,859,230,000,000x^{12} \\
& +17,198,351,808,917,834,700,000,000x^{14} \\
& -139,879,928,045,865,060,000,000,000x^{16} \\
& +863,964,261,459,754,784,000,000,000x^{18} \\
& -4,137,934,094,359,877,850,000,000,000x^{20} \\
& +15,602,340,243,235,729,000,000,000,000x^{22} \\
& -46,807,020,729,707,182,700,000,000,000x^{24} \\
& +112,480,871,353,542,505,000,000,000,000x^{26} \\
& -217,224,963,195,994,770,000,000,000,000x^{28} \\
& +337,073,218,752,405,700,000,000,000,000x^{30} \\
& -418,623,191,031,213,534,000,000,000,000x^{32} \\
& +412,653,519,857,862,853,000,000,000,000x^{34} \\
& -318,332,715,318,922,786,000,000,000,000x^{36} \\
& +187,920,450,721,696,958,000,000,000,000x^{38} \\
& -81,914,042,622,278,177,900,000,000,000x^{40} \\
& +24,831,086,091,073,870,200,000,000,000x^{42} \\
& -4,672,233,957,939,903,860,000,000,000x^{44} \\
& +410,795,449,442,059,171,000,000,000x^{46}) \qquad \text{(A.37)}
\end{aligned}
$$

$$
\begin{aligned}
P_{47}(x) \;=\; & \frac{x}{2^{47}}(-773,942,488,394,400 + 290,744,394,806,829,568x^2 \\
& -32,621,521,097,326,276,600x^4 \\
& +1,728,940,618,158,292,860,000x^6 \\
& -52,828,741,110,392,277,700,000x^8 \\
& +1,040,245,938,591,906,150,000,000x^{10} \\
& -14,163,348,548,520,570,100,000,000x^{12} \\
& +139,879,928,045,865,060,000,000,000x^{14} \\
& -1,036,757,113,751,705,800,000,000,000x^{16}
\end{aligned}
$$

$$
\begin{aligned}
&+5,911,334,420,514,112,160,000,000,000x^{18}\\
&-26,403,960,411,629,698,300,000,000,000x^{20}\\
&+93,614,041,459,414,383,000,000,000,000x^{22}\\
&-265,863,877,744,736,799,000,000,000,000x^{24}\\
&+608,229,896,948,785,413,000,000,000,000x^{26}\\
&-1,123,577,395,841,352,330,000,000,000,000x^{28}\\
&+1,674,492,764,124,854,140,000,000,000,000x^{30}\\
&-2,004,317,096,452,477,000,000,000,000,000x^{32}\\
&+1,909,996,291,913,536,580,000,000,000,000x^{34}\\
&-1,428,195,425,484,896,970,000,000,000,000x^{36}\\
&+819,140,426,222,781,674,000,000,000,000x^{38}\\
&-347,635,205,275,034,254,000,000,000,000x^{40}\\
&+102,789,147,074,677,887,000,000,000,000x^{42}\\
&-18,896,590,674,334,723,200,000,000,000x^{44}\\
&+1,625,701,140,345,170,280,000,000,000x^{46}) \qquad \text{(A.38)}
\end{aligned}
$$

$$
\begin{aligned}
P_{48}(x) \;=\; &\frac{1}{2^{48}}(32,247,603,683,100\\
&-37,923,181,931,325,600x^{2}\\
&+7,413,982,067,574,155,260x^{4}\\
&-576,313,539,386,097,533,000x^{6}\\
&+23,772,933,499,676,526,600,000x^{8}\\
&-602,247,648,658,471,969,000,000x^{10}\\
&+10,229,085,062,820,410,000,000,000x^{12}\\
&-123,423,465,922,822,110,000,000,000x^{14}\\
&+1,101,554,433,361,187,300,000,000,000x^{16}\\
&-7,487,690,265,984,541,190,000,000,000x^{18}\\
&+39,605,940,617,444,549,700,000,000,000x^{20}\\
&-165,624,842,582,040,827,000,000,000,000x^{22}\\
&+553,883,078,634,868,414,000,000,000,000x^{24}\\
&-1,492,927,928,874,291,290,000,000,000,000x^{26}\\
&+3,258,374,447,939,921,550,000,000,000,000x^{28}\\
&-5,767,697,298,652,275,170,000,000,000,000x^{30}\\
&+8,267,808,022,866,467,540,000,000,000,000x^{32}
\end{aligned}
$$

$$
\begin{aligned}
&-9,549,981,459,567,684,020,000,000,000,000x^{34}\\
&+8,807,205,123,823,530,400,000,000,000,000x^{36}\\
&-6,389,295,324,537,697,560,000,000,000,000x^{38}\\
&+3,563,260,854,069,100,240,000,000,000,000x^{40}\\
&-1,473,311,108,070,383,340,000,000,000,000x^{42}\\
&+425,173,290,172,531,288,000,000,000,000x^{44}\\
&-76,407,953,596,223,016,400,000,000,000x^{46}\\
&+6,435,067,013,866,299,580,000,000,000x^{48})
\end{aligned}
\tag{A.39}
$$

$$
\begin{aligned}
P_{49}(x) \;=\; & \frac{x}{2^{49}}(3,160,265,160,943,800\\
&-1,289,388,185,665,070,340x^{2}+157,176,419,832,572,051,000x^{4}\\
&-9,056,355,618,924,390,250,000x^{6}\\
&+301,123,824,329,235,985,000,000x^{8}\\
&-6,460,474,776,518,153,610,000,000x^{10}\\
&+95,996,029,051,083,847,700,000,000x^{12}\\
&-1,036,757,113,751,705,660,000,000,000x^{14}\\
&+8,423,651,549,232,609,380,000,000,000x^{16}\\
&-52,807,920,823,259,396,700,000,000,000x^{18}\\
&+260,267,609,771,778,458,000,000,000,000x^{20}\\
&-1,022,553,375,941,295,480,000,000,000,000x^{22}\\
&+3,234,677,179,227,631,500,000,000,000,000x^{24}\\
&-8,294,044,049,301,618,400,000,000,000,000x^{26}\\
&+17,303,091,895,956,824,400,000,000,000,000x^{28}\\
&-29,396,650,747,969,658,900,000,000,000,000x^{30}\\
&+40,587,421,203,162,661,300,000,000,000,000x^{32}\\
&-45,294,197,779,663,868,700,000,000,000,000x^{34}\\
&+40,465,537,055,405,410,000,000,000,000,000x^{36}\\
&-28,506,086,832,552,801,900,000,000,000,000x^{38}\\
&+15,469,766,634,739,020,900,000,000,000,000x^{40}\\
&-6,235,874,922,530,459,080,000,000,000,000x^{42}\\
&+1,757,382,932,713,129,360,000,000,000,000x^{44}\\
&-308,883,216,665,582,415,000,000,000,000x^{46}\\
&+25,477,612,258,980,858,000,000,000,000x^{48})
\end{aligned}
\tag{A.40}
$$

$$
\begin{aligned}
P_{50}(x) \;=\; & \frac{1}{2^{50}}(-126,410,606,437,752 \\
& +161,173,523,208,133,792x^{2} \\
& -34,168,786,920,124,366,800x^{4} \\
& +2,881,567,696,930,487,660,000x^{6} \\
& -129,053,067,569,672,562,000,000x^{8} \\
& +3,553,261,127,084,984,750,000,000x^{10} \\
& -65,681,493,561,267,896,200,000,000x^{12} \\
& +863,964,261,459,754,646,000,000,000x^{14} \\
& -8,423,651,549,232,608,280,000,000,000x^{16} \\
& +62,709,405,977,620,535,200,000,000,000x^{18} \\
& -364,374,653,680,489,827,000,000,000,000x^{20} \\
& +1,679,909,117,617,842,600,000,000,000,000x^{22} \\
& -6,220,533,036,976,214,640,000,000,000,000x^{24} \\
& +18,661,599,110,928,642,800,000,000,000,000x^{26} \\
& -45,617,242,271,158,900,600,000,000,000,000x^{28} \\
& +91,129,617,318,705,942,900,000,000,000,000x^{30} \\
& -148,820,544,411,596,407,000,000,000,000,000x^{32} \\
& +198,162,115,286,029,465,000,000,000,000,000x^{34} \\
& -213,889,267,292,857,156,000,000,000,000,000x^{36} \\
& +185,289,564,411,593,194,000,000,000,000,000x^{38} \\
& -126,852,086,404,859,974,000,000,000,000,000x^{40} \\
& +67,035,655,417,202,428,300,000,000,000,000x^{42} \\
& -26,360,743,990,696,939,000,000,000,000,000x^{44} \\
& +7,258,755,591,641,186,970,000,000,000,000x^{46} \\
& -1,248,403,000,690,062,290,000,000,000,000x^{48} \\
& +100,891,344,545,564,202,000,000,000,000x^{50}) \qquad \text{(A.41)}
\end{aligned}
$$

$$
\begin{aligned}
P_{51}(x) \;=\; & \frac{x}{2^{51}}(-12,893,881,856,650,704 \\
& +5,694,797,820,020,726,780x^{2} \\
& -751,713,312,242,735,907,000x^{4} \\
& +46,928,388,207,153,653,000,000x^{6} \\
& -1,692,029,108,135,706,950,000,000x^{8} \\
& +39,408,896,136,760,734,300,000,000x^{10} \\
& -636,605,245,286,134,938,000,000,000x^{12} \\
& +7,487,690,265,984,540,090,000,000,000x^{14} \\
& -66,398,194,564,539,388,600,000,000,000x^{16} \\
& +455,468,317,100,612,319,000,000,000,000x^{18} \\
& -2,463,866,705,839,502,610,000,000,000,000x^{20} \\
& +10,663,770,920,530,654,000,000,000,000,000x^{22} \\
& -37,323,198,221,857,285,600,000,000,000,000x^{24} \\
& +106,440,231,966,037,441,000,000,000,000,000x^{26} \\
& -248,535,319,960,107,114,000,000,000,000,000x^{28} \\
& +476,225,742,117,108,501,000,000,000,000,000x^{30} \\
& -748,612,435,525,000,209,000,000,000,000,000x^{32} \\
& +962,501,702,817,857,329,000,000,000,000,000x^{34} \\
& -1,005,857,635,377,220,210,000,000,000,000,000x^{36} \\
& +845,680,576,032,399,754,000,000,000,000,000x^{38} \\
& -563,099,505,504,500,383,000,000,000,000,000x^{40} \\
& +289,968,183,897,666,324,000,000,000,000,000x^{42} \\
& -111,300,919,071,831,519,000,000,000,000,000x^{44} \\
& +29,961,672,016,561,493,800,000,000,000,000x^{46} \\
& -5,044,567,227,278,211,090,000,000,000,000x^{48} \\
& +399,608,854,866,744,482,000,000,000,000x^{50}) \qquad \text{(A.42)}
\end{aligned}
$$

$$
\begin{aligned}
P_{52}(x) \;=\; & \frac{1}{2^{52}}(495,918,532,948,104 \\
& -683,375,738,402,487,296x^{2} \\
& +156,606,940,050,570,019,000x^{4} \\
& -14,282,552,932,611,982,900,000x^{6} \\
& +692,193,726,055,516,381,000,000x^{8} \\
& -20,642,755,119,255,622,900,000,000x^{10} \\
& +413,793,409,435,987,758,000,000,000x^{12} \\
& -5,911,334,420,514,109,960,000,000,000x^{14} \\
& +62,709,405,977,620,526,400,000,000,000x^{16} \\
& -509,052,824,994,801,971,000,000,000,000x^{18} \\
& +3,233,825,051,414,347,620,000,000,000,000x^{20} \\
& -16,351,115,411,480,336,300,000,000,000,000x^{22} \\
& +66,648,568,253,316,589,200,000,000,000,000x^{24} \\
& -221,068,174,083,308,531,000,000,000,000,000x^{26} \\
& +600,627,023,236,925,564,000,000,000,000,000x^{28} \\
& -1,342,090,727,784,578,320,000,000,000,000,000x^{30} \\
& +2,470,421,037,232,500,530,000,000,000,000,000x^{32} \\
& -3,743,062,177,625,001,190,000,000,000,000,000x^{34} \\
& +4,652,091,563,619,643,730,000,000,000,000,000x^{36} \\
& -4,711,648,923,609,084,100,000,000,000,000,000x^{38} \\
& +3,847,846,620,947,418,790,000,000,000,000,000x^{40} \\
& -2,493,726,381,519,930,360,000,000,000,000,000x^{42} \\
& +1,252,135,339,558,104,680,000,000,000,000,000x^{44} \\
& -469,399,528,259,463,348,000,000,000,000,000x^{46} \\
& +123,591,897,068,316,174,000,000,000,000,000x^{48} \\
& -20,380,051,598,203,972,600,000,000,000,000x^{50} \\
& +1,583,065,848,125,949,460,000,000,000,000x^{52})
\end{aligned}
\tag{A.43}
$$

$$
\begin{aligned}
P_{53}(x) \;=\; & \frac{x}{2^{53}}(52,567,364,492,499,024 \\
& -25,057,110,408,091,197,400x^{2} \\
& +3,570,638,233,152,995,720,000x^{4} \\
& -240,763,035,149,744,837,000,000x^{6} \\
& +9,383,070,508,752,556,760,000,000x^{8} \\
& -236,453,376,820,564,423,000,000,000x^{10} \\
& +4,137,934,094,359,877,300,000,000,000x^{12} \\
& -52,807,920,823,259,387,900,000,000,000x^{14} \\
& +509,052,824,994,801,971,000,000,000,000x^{16} \\
& -3,804,500,060,487,467,230,000,000,000,000x^{18} \\
& +22,482,783,690,785,461,300,000,000,000,000x^{20} \\
& -106,637,709,205,306,535,000,000,000,000,000x^{22} \\
& +410,555,180,440,430,160,000,000,000,000,000x^{24} \\
& -1,293,658,203,894,916,590,000,000,000,000,000x^{26} \\
& +3,355,226,819,461,446,230,000,000,000,000,000x^{28} \\
& -7,186,679,381,040,000,390,000,000,000,000,000x^{30} \\
& +12,726,411,403,925,003,700,000,000,000,000,000x^{32} \\
& -18,608,366,254,478,577,200,000,000,000,000,000x^{34} \\
& +22,380,332,387,143,151,200,000,000,000,000,000x^{36} \\
& -21,987,694,976,842,392,400,000,000,000,000,000x^{38} \\
& +17,456,084,670,639,509,600,000,000,000,000,000x^{40} \\
& -11,018,790,988,111,321,200,000,000,000,000,000x^{42} \\
& +5,398,094,574,983,829,580,000,000,000,000,000x^{44} \\
& -1,977,470,353,093,058,490,000,000,000,000,000x^{46} \\
& +509,501,289,955,099,342,000,000,000,000,000x^{48} \\
& -82,319,424,102,549,375,400,000,000,000,000x^{50} \\
& +6,272,525,058,612,252,700,000,000,000,000x^{52})
\end{aligned}
\tag{A.44}
$$

$$
\begin{aligned}
P_{54}(x) \;=\; & \frac{1}{2^{54}}(-1,946,939,425,648,112 \\
& +2,891,205,047,087,446,530x^{2} \\
& -714,127,646,630,599,328,000x^{4} \\
& +70,222,551,918,675,583,000,000x^{6} \\
& -3,671,636,286,033,608,750,000,000x^{8} \\
& +118,226,688,410,282,212,000,000,000x^{10} \\
& -2,561,578,248,889,447,720,000,000,000x^{12} \\
& +39,605,940,617,444,536,500,000,000,000x^{14} \\
& -455,468,317,100,612,249,000,000,000,000x^{16} \\
& +4,015,861,174,958,993,530,000,000,000,000x^{18} \\
& -27,772,850,441,558,511,500,000,000,000,000x^{20} \\
& +153,291,706,982,628,138,000,000,000,000,000x^{22} \\
& -684,258,634,067,383,600,000,000,000,000,000x^{24} \\
& +2,494,912,250,368,767,860,000,000,000,000,000x^{26} \\
& -7,484,736,751,106,303,000,000,000,000,000,000x^{28} \\
& +18,565,588,401,020,000,900,000,000,000,000,000x^{30} \\
& -38,179,234,211,775,004,200,000,000,000,000,000x^{32} \\
& +65,129,281,890,675,021,500,000,000,000,000,000x^{34} \\
& -92,008,033,147,144,066,000,000,000,000,000,000x^{36} \\
& +107,190,013,012,106,670,000,000,000,000,000,000x^{38} \\
& -102,242,781,642,317,127,000,000,000,000,000,000x^{40} \\
& +78,968,002,081,464,447,400,000,000,000,000,000x^{42} \\
& -48,582,851,174,854,462,800,000,000,000,000,000x^{44} \\
& +23,235,276,648,843,441,100,000,000,000,000,000x^{46} \\
& -8,321,854,402,599,954,200,000,000,000,000,000x^{48} \\
& +2,099,145,314,615,009,150,000,000,000,000,000x^{50} \\
& -332,443,828,106,449,364,000,000,000,000,000x^{52} \\
& +24,857,784,491,537,444,000,000,000,000,000x^{54})
\end{aligned}
\tag{A.45}
$$

$$
\begin{aligned}
P_{55}(x) \;=\; & \frac{x}{2^{55}}(-214,163,336,821,292,320 \\
& +109,865,791,789,322,945,000x^{2} \\
& -16,853,412,460,482,141,400,000x^{4} \\
& +1,223,878,762,011,202,920,000,000x^{6} \\
& -51,402,908,004,470,526,800,000,000x^{8} \\
& +1,397,224,499,394,244,390,000,000,000x^{10} \\
& -26,403,960,411,629,693,900,000,000,000x^{12} \\
& +364,374,653,680,489,757,000,000,000,000x^{14} \\
& -3,804,500,060,487,467,230,000,000,000,000x^{16} \\
& +30,858,722,712,842,791,500,000,000,000,000x^{18} \\
& -198,377,503,153,989,361,000,000,000,000,000x^{20} \\
& +1,026,387,951,101,075,470,000,000,000,000,000x^{22} \\
& -4,324,514,567,305,864,410,000,000,000,000,000x^{24} \\
& +14,969,473,502,212,608,300,000,000,000,000,000x^{26} \\
& -42,843,665,540,815,389,900,000,000,000,000,000x^{28} \\
& +101,811,291,231,400,011,000,000,000,000,000,000x^{30} \\
& -201,308,689,480,268,205,000,000,000,000,000,000x^{32} \\
& +331,228,919,329,718,674,000,000,000,000,000,000x^{34} \\
& -452,580,054,940,005,983,000,000,000,000,000,000x^{36} \\
& +511,213,908,211,585,635,000,000,000,000,000,000x^{38} \\
& -473,808,012,488,786,740,000,000,000,000,000,000x^{40} \\
& +356,274,241,948,932,672,000,000,000,000,000,000x^{42} \\
& -213,764,545,169,359,655,000,000,000,000,000,000x^{44} \\
& +99,862,252,831,199,468,800,000,000,000,000,000x^{46} \\
& -34,985,755,243,583,482,400,000,000,000,000,000x^{48} \\
& +8,643,539,530,767,684,610,000,000,000,000,000x^{50} \\
& -1,342,320,362,543,021,990,000,000,000,000,000x^{52} \\
& +98,527,218,530,093,882,200,000,000,000,000x^{54}) \qquad \text{(A.46)}
\end{aligned}
$$