Image Representations for Pattern Recognition

Thai V. Hoang

Nancy-Université
Université Nancy 2

École doctorale IAEM Lorraine

UFR mathématiques et informatique **Département de formation doctorale en informatique**

Image Representations for Pattern Recognition

THÈSE

présentée et soutenue publiquement le 14 décembre 2011

pour l'obtention du

Doctorat de l'université Nancy 2

(spécialité informatique)

par

Thai V. Hoang

(Hoàng Văn Thái)

Composition du jury

Président :	Jean-Marc Ogier	Professeur, Université de La Rochelle
Rapporteurs :	Jean-Philippe Domenger	Professeur, Université Bordeaux 1
	Nicole Vincent	Professeur, Université Paris Descartes
Examinateurs :	Atilla Baskurt	Professeur, INSA Lyon
	David W. Ritchie	Directeur de recherche, INRIA Nancy
	Djemel Ziou	Professeur, Université de Sherbrooke
Directeur de thèse :	Salvatore Tabbone	Professeur, Université Nancy 2

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Mis en page avec la classe thloria.

Dedicated to my parents,
to Mai, to Tom

Acknowledgments

This thesis is the outgrowth of my three-year research work that had been carried out at LORIA under the support of a CNRS's BDI-PED fellowship. In the course of writing this thesis, I had been accompanied and helped by several people, in one way or the other, and I would like to express my gratitude to all of them.

First of all, I would like to express my deep gratitude to my supervisor Salvatore-Antoine Tabbone for helping me to get the CNRS's fellowship and for his continuing encouragement and supervision during my stay in his team. I will always be indebted to him for having confidence in me and accepting me into his team, and for the valuable expertise he shared with me in the very beginning days. I particularly appreciate the great freedom I had in defining the research problems and in finding the solutions for them, leading to the numerous contributions presented in this thesis. I also owe him very special thanks for the help he gave me in settling in Nancy.

I would like to thank Jean-Philippe Domenger and Nicole Vincent for accepting to review my thesis and sharing interesting comments and discussions with me. I am also grateful to Atilla Baskurt, Jean-Marc Ogier, Dave Ritchie, and Djemel Ziou for accepting to be part of the jury. Thanks a lot to Dave Ritchie for commenting on my English and accepting me as a postdoctoral researcher in his team next year. Special thanks are owed to Djemel Ziou for inviting me to Sherbrooke for one month and sharing with me his expertise in statistical modeling, and for guiding and supporting me.

I am more thankful than I can say to Elisa H. Barney Smith for a remarkable collaboration from which I benefited a lot. I still remember the excitement of working with her on an image denoising problem and then turning it into a paper. I also would like to thank her for reading a part of the manuscript and helping correct my English.

I am also extremely grateful to Eric Castelli and Ngoc-Yen Pham for allowing me to work in the SEPIA project and for helping me to get the CNRS's fellowship. The project work brought me, an automatic control engineer by training, to the field of image analysis and recognition. I believe that this thesis would be impossible without that opportunity.

I would like to thank colleagues at LORIA and friends at Nancy, too numerous to name, for their interaction and friendly support during the last three years. In this context, I heartily thank Philippe Dosch for his outstanding technical support. I am also very grateful to Hervé Locteau for his help with my French and for his goodwill and humor.

And finally, I would like to thank Mai and Tom for giving me so much love and for their patience during the final period of my PhD. Special thanks also go to my parents for their spiritual care and protection, and for their endless love and support.

Abstract

One of the main requirements in many signal processing applications is to have a "meaningful representation" in which signal's characteristics are readily apparent. For example, for recognition, the representation should highlight salient features; for denoising, it should efficiently separate signal and noise; and for compression, it should capture a large part of signal using only a few coefficients. Interestingly, despite these seemingly different goals, good performance of signal processing applications generally has roots in the appropriateness of the adopted representations.

Representing a signal involves the design of a set of elementary generating signals, or a *dictionary* of *atoms*, which is used to decompose the signal. For many years, dictionary design has been pursued by many researchers for various fields of applications: Fourier transform was proposed to solve the heat equation; Radon transform was created for the reconstruction problem; wavelet transform was developed for piece-wise smooth, one-dimensional signals with a finite number of discontinuities; and contourlet transform was designed to efficiently represent two-dimensional signals made of smooth regions separated by smooth boundaries, etc.

For the developed dictionaries up to the present time, they can be roughly classified into two families: *mathematical models* of the data and *sets of realizations* of the data. Dictionaries of the first family are characterized by *analytical formulations*, which can sometimes be fast implemented. The representation coefficients of a signal in one dictionary are obtained by performing *signal transform*. Dictionaries of the second family, which are often *general overcomplete*, deliver greater flexibility and the ability to adapt to specific signal data. They are the results of much more recent dictionary designing approaches where dictionaries are learned from data for their representation.

The existence of many dictionaries naturally leads to the problem of selecting the most appropriate one for the representation of signals in a certain situation. The selected dictionary should have distinguished and beneficial properties which are preferable in the targeted applications. Speaking differently, it is the actual application that controls the selection of dictionary, not the reverse. In the framework of this thesis, three types of dictionaries, which correspond to three types of transforms/representations, will be studied for their applicability in some image analysis and pattern recognition tasks. They are the *Radon transform*, *unit disk-based moments*, and *sparse representation*. The Radon transform and unit disk-based moments are for invariant pattern recognition problems, whereas sparse representation for image denoising, separation, and classification problems.

This thesis contains a number of theoretical contributions which are accompanied by numerous validating experimental results. For the Radon transform, it discusses possible directions that can be followed to define invariant pattern descriptors, leading to the proposal of two descriptors that are totally invariant to rotation, scaling, and translation. For unit disk-based moments, it presents a unified view on strategies that have been used to define unit disk-based orthogonal moments, leading to the proposal of four generic polar harmonic moments and strategies for their fast computation. For sparse representation, it uses sparsity-based techniques for denoising and separation of graphical document images and proposes a representation framework that balances the three criteria sparsity, reconstruction error, and discrimination power for classification.

Keywords: image representation, Radon transform, unit disk-based moment, sparse representation, invariant pattern recognition, image denoising, image separation, classification.

Table of Contents

List of Figures

List of Tables

Chapter 1

General Introduction

The process of sampling a signal for its representation in digital systems leads to a sum of Kronecker delta functions which are convenient for display but mostly inefficient for analysis and recognition tasks. For this reason, one of the main requirements in many signal processing applications is to have a "meaningful representation" in which signal's characteristics are readily apparent. For example,

- for recognition: the representation should highlight salient features,
- for denoising: it should efficiently separate signal and noise,
- for compression: it should capture a large part of signal using only a few coefficients.

Interestingly, despite these seemingly different goals, good performance of signal processing applications generally has roots in the appropriateness of the adopted representation.

Representing a signal involves the design of a set of elementary generating signals, or a *dictionary* of *atoms*, which is used to decompose the signal. For many years, dictionary design has been pursued by many researchers for various fields of applications: Fourier transform was proposed to solve the heat equation; Radon transform was created for the reconstruction problem; wavelet transform was developed for piece-wise smooth, one-dimensional (1D) signals with a finite number of discontinuities; and contourlet transform was designed to efficiently represent two-dimensional (2D) signals made of smooth regions separated by smooth boundaries, etc. For the developed dictionaries up to the present time, they can be roughly classified into two families: *mathematical models* of the data and *sets of realizations* of the data.

Dictionaries of the first family are characterized by *analytical formulations*. The representation coefficients of a signal in one dictionary are obtained by performing *signal transform* [179], which can sometimes be fast implemented. When the dictionary forms a basis, these coefficients are unique and a signal is then represented as a linear combination of dictionary atoms. In this case, the synthesis operator of the transform is defined as the dictionary (for the orthogonal case) or as the dictionary inverse (for the bi-orthogonal case). However, in spite of the mathematical simplicity that explains their dominance for many years, orthogonal and bi-orthogonal dictionaries have weakness in their expressiveness due to the uniqueness of the representation coefficients. This main limitation has led to the recent development of *analytical overcomplete* dictionaries [123], which have more atoms than (bi-)orthogonal dictionaries and eventually promise to "better represent" a wider range of signal phenomena for various applications.

Dictionaries of the second family, which are often *general overcomplete* dictionaries, deliver greater flexibility and the ability to adapt to specific signal data. They are the results of

much more recent dictionary designing approaches where dictionaries are learned from data for their representation. Due to the unavoidable learning process, this line of approaches is strongly influenced by the latest advances in computational algebra and optimization algorithms. The main advantage of learned dictionaries is that they lead to results which are comparable and sometimes superior to the state-of-the-art in many practical signal processing applications. However, the cost of these approaches, as in the case of the Karhunen–Loève transform [171], is dictionaries with unknown inner structure or fast implementation that often prevents them from being used in time-critical applications.

The existence of many dictionaries naturally leads to the problem of selecting the most appropriate one for the representation of signals in a certain situation since there exists no dictionary that fits all purposes. The selected dictionary should have distinguished and beneficial properties which are preferable in the targeted applications. Speaking differently, it is the actual application that controls the selection of dictionary, not the reverse. In the framework of this thesis, three types of dictionaries, which correspond to three types of transforms/representations, will be studied for their applicability in some image analysis and pattern recognition tasks. They are the *Radon transform*, *unit disk-based moments*, and *sparse representation*. Radon transform and unit disk-based moments are for invariant pattern recognition problems, whereas sparse representation for image denoising, separation, and general classification problems.

1.1 Invariant representation

The problem of recognizing patterns that undergo *geometric transformations* like rotation, scaling, and translation (RST) is an important topic in pattern recognition and is the goal of many research works. A number of approaches were proposed for this problem and they can be classified into three main lines: brute force, normalization, and invariant features. *Brute force* approaches are the most trivial ones, using "complete" training datasets; for each pattern category, the training dataset contains all its RST-transformed versions. This line of approaches has inherent limitations in both storage requirement and time complexity that make it practically inapplicable. *Normalization* of patterns is a solution for the reduction of the size of training datasets. The burden of the encoded RST transformation parameters in input patterns is alleviated by normalizing them regarding their orientation, size, and position. However, despite its efficiency in the recognition stage, normalization involves difficult inverse problems that are often ill-conditioned or ill-posed, leading to unreliable normalization results. Approaches using *invariant features* are based on the idea of describing each pattern by a set of measurable quantities that are insensitive to RST transformations while providing enough discrimination power for recognition. Mathematically speaking, if f is a pattern and g is another pattern described as $g = \mathcal{O}(f)$, where $\mathcal{O}$ is an RST transformation operator, then the invariant $\mathcal{I}$ is a functional which satisfies $\mathcal{I}(f) = \mathcal{I}(\mathcal{O}(f))$.

Many pattern descriptors were proposed in the literature for the extraction of pattern's invariant features using techniques that allow operator $\mathcal{O}$ to be rotation, scaling, translation, or their combination [233, 244]. Translation and scaling invariance could be obtained by using the Fourier [98] and Mellin [19] transforms respectively; rotation invariance by computing the harmonic expansion [99] or performing the discrete Fourier transform on the circular slices of the pattern represented in the polar space [243], etc. However, the task of combining several techniques to make operator $\mathcal{O}$ a full RST transformation while guaranteeing the discrimination power of the extracted invariant features is challenging and has attracted attention of many researchers. Most of the existing methods do not allow operator $\mathcal{O}$ to be a full RST transformation, they usually require normalization for the unavailability of any of RST transformations in operator

$\mathcal{O}$. For example, methods based on the theory of moments [213] usually normalize an input pattern regarding its centroid position and size: the pattern's centroid is required to coincide with the origin of the coordinate system and the longest distance from this centroid to a pattern point is set to a fixed value. These normalizations usually introduce errors, are sensitive to noise, and thus induce inaccuracy in the later recognition/matching process [113].

However, the existence of many pattern descriptors, each may or may not be fully invariant to RST transformations, suggests that each descriptor is particularly more suitable for certain applications and not to the others. For example, in situations where invariance to full RST transformations is not a must, fully RST-invariant descriptors are usually not employed. This is because, in general, invariance to any geometric transformation has to be paid by a loss of information. A notable example of information loss is the removal of phase from Fourier transform data in order to have translation invariance [166]. In situations where patterns are acquired from a constrained environment (i.e., from a fixed location and of a fixed size), moments could thus be computed directly from these patterns without the need of aforementioned normalizations. The extracted moments, if used only for this type of patterns, are naturally invariant to scaling and translation. In the sequel of this thesis, several invariant descriptors will be proposed by exploiting patterns in the two domains: Radon transform and unit disk-based moments.

1.1.1 Radon transform

The Radon transform is named after Johann Radon (1887–1956), an Austrian mathematician who wrote a classic paper in 1917 (its English translation is [181]) on the problem of determining a 2D function from the knowledge of its line integrals. This problem, which is actually the inverse problem of Radon transform, arises in widely diverse fields which include medical imaging, astronomy, crystallography, electron microscopy, geophysics, optics, and material science where the general problem of unfolding the internal structure of an object by its projections is known as the problem of *reconstruction from projections*. Previously, Radon transform was known by very few engineers and scientists who worked directly on this reconstruction problem in one of the major areas of applications. Nowadays, it is widely known by working scientists in medicine, engineering, physical science, and mathematics. A good introduction of Radon transform and some of its applications could be found in the monograph by Stanley R. Deans [51] (or [52] for the more concise and recently updated version).

The Radon transform of a 2D function is a integral transform which consists of integrals over straight lines in $\mathbb{R}^2$. From the mathematical viewpoint, the Radon transform of a function is "projections" of that function onto a space formed by elementary functions, each is a straight line in $\mathbb{R}^2$. However, from the perspective of signal decomposition, the set of these straight lines can be viewed as an "analysis" dictionary in which the representation of a signal is equivalent to its Radon transform. Since line-based atoms have analytical form, it can be concluded that this *dictionary of lines* belongs to the first family of analytical dictionaries.

The distinct characteristic of Radon transform is that it is one of some rare transforms that have geometric interpretation in the spatial domain. The other transforms of this group are the generalized Hough transform [10], trace transform [112], and geometric transform [134] which replace line integrals by integrals over more complex domains like circles, squares, closed contours, or even replace integrals by other functionals. The simplicity of line integrals has led to a wide adoption of Radon transform in image analysis and computer vision communities, sometimes under the name of Hough transform [97]. Typical applications are line and curve detection [66], texture analysis [107], and deblurring [44], etc. In pattern recognition, several invariant descriptors were proposed based on Radon transform. These descriptors are different from the

others in the sense that Radon transform is used to create an intermediate representation upon which invariant features are extracted from for the purpose of indexing/matching. There are some reasons for the utilization of Radon transform:

- It is a rich transform with one-to-many mapping, each pattern point lies on a set of lines in the spatial domain and contributes a curve to the transform data.
- It is a lossless transform, patterns can be reconstructed accurately by the inverse Radon transform.
- It has reasonably low complexity, requiring only $O(N^2 \log N)$ operations for an input pattern image of size $N \times N$.
- It has useful properties concerning RST transformations applied on patterns.

Among the above reasons, the final one is of paramount importance for invariant pattern recognition problems. By applying Radon transform on an RST-transformed pattern, the transformation parameters are encoded in the radial (for translation and scaling) and angular (for rotation) slices of the obtained transform data respectively. The exploitation of this encoded information in order to define Radon transform-based descriptors that are totally invariant to RST transformations will be presented in Chapter 2.

1.1.2 Image moments

In mathematics, moments are, loosely speaking, scalar quantities that are used to characterize functions and to capture their significant features. They have been used for centuries in statistics to measure quantitatively the shape of a probability density function [171]. For example, the "second moment" is widely used to measure the "width" of a distribution in one dimension or the shape of a distribution in higher dimensions. Other moments describe other aspects of a distribution such as the mean, variance, skewness, peakiness, etc. The mathematical concept of moments has a close relationship with its physical counterparts which are, however, often represented somewhat differently. As an example, in classic rigid-body mechanics, the second moment is used to measure the body's mass distribution, which has no link with the aforementioned width of a distribution.

Moments were introduced to the image processing and pattern recognition communities almost 50 years ago by Ming-Kuei Hu [100]. Similar to Radon transform, from the mathematical viewpoint, moments of a function are "projections" of that function onto a space formed by elementary functions which are not necessarily orthogonal. From the perspective of signal decomposition, the set of all these elementary functions can also be viewed as an "analysis" dictionary which belongs to the first family of analytical dictionaries. It is well-known that image moments are useful to describe objects after segmentation and they play a very important role in defining invariant features in pattern recognition problems. An image moment is defined as a certain particular weighted average of the image pixels' intensities, where the weighting function is 2D and characterized by a kernel function. In certain situations, functions of such moments, usually chosen to have some attractive properties or interpretations, are also called moments.

Various sets of image moments which differ in the sets of kernel functions were proposed. The most common and simplest choice for the kernel functions is the standard separable polynomials $x^p y^q$ $(p, q \in \mathbb{Z}^+)$ of *geometric moments* m_{pq}. Similar to the concept of moments in mathematics and physics, geometric moments of low orders also have intuitive meaning: m_{00} is the pattern's "mass" (the total sum of pixels' intensities); $\frac{m_{10}}{m_{00}}$ and $\frac{m_{10}}{m_{00}}$ define the pattern's centroid position; m_{20} and m_{02} describe the pattern's distributions of mass with respect to the axes, etc. In

addition, by means of the Weierstrass approximation theorem [193, Theorem 7.26], the set of kernel functions $\{x^p y^q : p, q \in \mathbb{Z}^+\}$ is complete. Combining this fact with the uniqueness and existence theorems [158] of moments of a piece-wise continuous and bounded intensity function f:

- *uniqueness*: the moment sequence $\{m_{pq}\}$ is uniquely determined by f,
- *existence*: the moments of all orders exist and finite,

any pattern can be characterized by its geometric moments. However, the application of geometric moments in pattern recognition problems is limited because of the non-orthogonality in $\{x^p y^q : p, q \in \mathbb{Z}^+\}$ [213] and the complexity in deriving invariant features from m_{pq} (see [100] for an example). Non-orthogonality causes information redundancy in m_{pq}, which in turn leads to difficulty in image reconstruction and low accuracy in pattern recognition.

In theory, polynomial sequences of the same degree are equivalent since they generate the same functional space. For this reason, moments generated from a certain set of kernel functions can be expressed by moments generated from any other set of kernel functions. By means of the Taylor series, moments of any type are thus equivalent to geometric moments. Nevertheless, in practice, different moments have different issues on invariance properties, numerical stability, computational complexity, and robustness to noise, etc. The quest for moments that "partially" resolve these issues has led to the proposals of many moments to date. They include complex moments [1], Legendre & Zernike moments [213], rotational moments [215], Tchebichef moments [156], just to name a few. A recent comprehensive survey on image moments is available in [82].

Originating from statistical science, the dictionary, or set of kernel functions by convention, used to compute moments is traditionally composed of polynomials of various orders. This historical standpoint has prevented the Fourier transform from being classified as a moment-computing method since Fourier basis is composed of harmonic functions. This restriction on the definition of kernel functions, however, has been violated when moments are brought to image processing, pattern recognition, and related fields. Typical examples are unit disk-based moments, which are usually used to define features for rotation-invariant pattern recognition problems. As will be seen in Section 3.1.1, it is necessary for these moments to employ complex exponential functions to define the angular kernels. In this line, Chapter 3 will present four classes of image moments that are defined on the unit disk using harmonic functions. These unit disk-based moments inherit the simplicity in deriving rotation-invariant features and, on the other hand, have some distinct characteristics that could make them more suitable for certain applications.

1.2 Sparse representation

The problem of representing signals sparsely in a dictionary [147], or *sparse coding* of signals, has recently attracted attention of many researchers from various fields of applications due to the potential use of sparsity-based representation to solve many challenging scientific problems concerning signals, e.g., compression, restoration/denoising, separation, and classification. Concretely, sparse coding of a signal consists of representing it as a linear combination of a few atoms from a given dictionary [29]. From this viewpoint, the sparse representation of a complex signal is only feasible when the dictionary, whose atoms can be defined as signals which ensemble generate the signal space, is overcomplete. Besides overcompleteness, the dictionary usually has no other constraint, it could be derived from an analytical transform or learned from data. The flexibility in defining dictionaries makes sparse representation different from the more traditional representations, such as the aforementioned Radon transform and image moments

where dictionaries are pre-defined and deterministic. This flexibility in dictionary design leads to the ability to

- compactly represent a large class of signals for compression,
- adapt to signal's morphological content for restoration/denoising,
- decompose a complex signal into separate sources for separation,
- capture signal's salient features for classification.

Looking back in history, sparsity could be considered as another form of *Occam's razor* [216], a principle attributed to the logician and Franciscan friar William of Ockham (1288–1348), which states that "Entities should not be multiplied unnecessarily"[1]. This principle has been used to justify many theories in physics (uncertainty theory in quantum mechanics), biology (evolutionary mechanism), science (heuristic argument), statistics (complexity penalization), etc. This principle, in fact, has been adopted or reinvented by many scientists throughout history, as in Gottfried Wilhelm Leibniz's *identity of observables* or in Ernst Mach's *principle of economy*. Nowadays, the most common and useful statement of the principle for scientists is

> *When you have two competing theories that make exactly the same predictions, the simpler one is the better.*

In spite of a long history, the significance of sparsity in vision has only become clear gradually over the last half century. The work of Horace Barlow in 1950s [13] led to one of the most important principles in sensory coding efficiency, which pointed out that the visual cortex must be a massive sparsifying engine. More specifically, it was recognized that 10^6 or more bits per second which arrive at the cortex is reduced to only dozens of bits per second by the time the information flow reaches the innermost abstract representation of the visual field. David H. Hubel and Torsten N. Wiesel then showed in their work [102] that the visual cortex employs multiscale and multidirection basis functions, much like what are now called wavelets, and thresholding devices which allow to ignore small wavelet coefficients and thereby sparsify them. Effective computational tools of the above theories were only available by the mid 1980s following the work of Peter J. Burt and Edward H. Adelson on pyramidal filter banks [31] and Ingrid Daubechies on wavelets [47]. It was finally found by the early 1990s that wavelets and their derivatives X-lets lead to sparse representation of much multimedia content (e.g., images, videos, sounds).

Studies in mammalian vision system [164] also gave a strong support to sparse representation. The receptive fields of simple cells in mammalian primary visual cortex can be characterized as being spatially localized, oriented, and bandpass (i.e., selective to structure at different scales), comparable to the basis functions of wavelet transforms, and having a strategy for producing a sparse distribution of output activity in response to natural images. Bruno A. Olshausen and David J. Field [165] validated this theory by considering the problem of efficient coding of natural images. They showed that when the dictionary is overcomplete and non-orthogonal, a coding strategy that maximizes sparseness (i.e., a small number of code elements are non-zero) will select only atoms that are necessary for representing a given input.

Due to the recent advances in optimization theory and numerical computation, finding a sparse representation of a signal in a given overcomplete dictionary becomes better-behaved and much more practical than it was supposed just a decade ago. In parallel with this development, it has been found that many important tasks dealing with media content can now be viewed as

[1] Its original Latin form is "Pluralitas non est ponenda sine neccesitate".

finding sparse representations in given dictionaries. For example, the media encoding standard JPEG and its successor, JPEG-2000, both are based on the notion of transform encoding that leads to a sparse representation. The more feasibility in finding sparse solutions, combined with the new insight into existing tasks, has fostered the adoption of sparse representation to solve many difficult problems in signal and image processing as reported in recently published two monographs [70, 207] or a special issue in *Proceedings of the IEEE* [12]. Chapter 4 will briefly review the sparse modeling framework and then apply sparse representation for document image processing and image classification. More explicitly, sparse representation will be used for removing noise that concentrates along graphical contours and for extracting text components from graphical document images. In addition, the current sparsifying frameworks will be modified to make the representation more suitable for classification tasks.

1.3 Thesis contributions

This thesis presents the research works on image representations for some image analysis and pattern recognition problems. It pursues both invariant and sparse representations introduced in the previous sections and makes the following main contributions:

Chapter 2 – Radon transform-based representation: This chapter provides a unified view on possible directions that can be followed to define invariant pattern descriptors using the Radon transform. It proves theoretically that the Radon transform has the property of suppressing additive white/"salt & pepper" noise. It generalizes an existing Radon transform-based descriptor, the R-signature, to have the generic R-signature that is totally invariant to RST transformations. It proposes to apply the 1D Fourier–Mellin and Fourier transforms on the radial and angular slices of the Radon transform data respectively to have the RFM descriptor that is totally invariant to RST transformations. It shows that the two proposed invariant pattern descriptors lead to superior experimental results over comparison descriptors in terms of retrieval rate on grayscale and binary noisy pattern datasets.

Chapter 3 – Unit disk-based representation: This chapter presents a unified view on strategies that have been used to define unit disk-based orthogonal moments. It introduces four generic harmonic radial kernels which correspond to four sets of generic polar harmonic moments and take existing sets of polar harmonic moments as special cases. It proves theoretically that the sets of generic polar harmonic kernels are complete in the Hilbert space of all square-integrable continuous complex-valued functions on the unit disk. It proposes several strategies for fast computation of polar harmonic kernels/moments based on the recursive computation of complex exponential and trigonometric functions. It shows experimentally that, when compared with existing moments of similar nature, the proposed generic polar harmonic moments are superior in terms of computational complexity and comparable in terms of representation capability and discrimination power.

Chapter 4 – Sparse representation: This chapter proposes to use sparse representation of images for the three following main problems.

- *Denoising*: It uses the synthesis operator of curvelet transform as the dictionary in a sparse representation framework for directional denoising. It demonstrates both theoretically and experientially that the information about the level of edge noise has a linear relationship

with the only framework's parameter. It shows that the proposed sparsity-based denoising method leads to superior performance over comparison methods on edge noise removal in bilevel graphical document images.

- *Separation*: It applies an existing sparsity-based separation technique using two appropriately chosen discriminative overcomplete dictionaries, each one gives sparse representation over one type of images and non-sparse representation over the other, for the classical problem of extracting text components from graphical document images. It proposes some heuristic rules to group text components into text strings in post-processing steps. It shows experimentally that the proposed sparsity-based text extraction method leads to better performance than the current benchmark.

- *Classification*: It proposes a new discriminative sparse coding method by adding a discriminative term to the conventional sparse representation framework, resulting in a model that is a controlled trade-off between sparsity, fidelity to the data, and discrimination power. It uses an information theoretic-based criterion, called minimum message length, to select the optimal statistical model. It shows that the proposed method leads to superior classification performance over comparison methods on the two common handwritten and face datasets.

Chapter 2

Radon Transform-based Invariant Pattern Representation

Contents

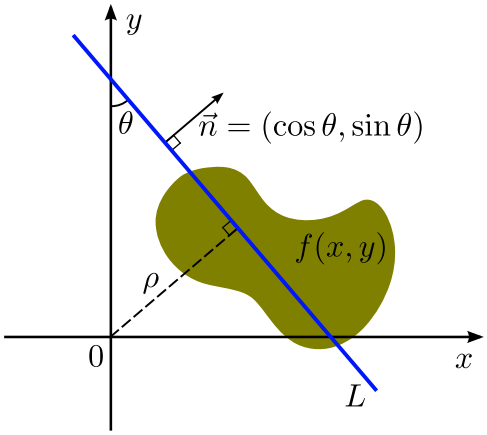


Figure 2.1: Geometric illustration of the Radon transform of a 2D function f. The Radon transform is a mapping from the spatial space (x, y) to the parameter space (θ, ρ) and can be mathematically represented by a line integral of f along all the lines $L(\theta, \rho)$ parameterized by (θ, ρ) represented in the spatial space (x, y).

2.1 The Radon transform

This section provides some basics of the Radon transform, started with its definition and its derived beneficial properties. A discussion on the robustness of the Radon transform to additive white/"salt & pepper" noise and on its efficient implementation strategies is also given. The inspiration for the derivation of RST invariants from the Radon transform of a pattern is provided along with a detailed review on related works. All these aspects are followed by a sketch of contributions that will be presented in this chapter.

2.1.1 Definition

Let f be a 2D function and $L(\theta, \rho)$ be a straight line in $\mathbb{R}^2$ represented by

$$L(\theta, \rho) = \{(x, y) \in \mathbb{R}^2 : x\cos\theta + y\sin\theta = \rho\},$$

where θ is the angle $L(\theta, \rho)$ makes with the y axis and ρ is the distance from the origin to $L(\theta, \rho)$. The Radon transform [51] of f, denoted by $\mathcal{R}_f$, is a function defined on the space of lines $L(\theta, \rho)$ $(\theta, \rho \in \mathbb{R})$ by the *line integral along each line*:

$$\mathcal{R}_f(\theta, \rho) = \int_{L(\theta,\rho)} f(x, y)\,\mathrm{d}x\mathrm{d}y = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,\mathrm{d}x\mathrm{d}y. \tag{2.1}$$

In the field of shape analysis and recognition, the function f is constrained to the following particular definition:

$$f(x, y) = \begin{cases} 1 & \text{if } x \in D \\ 0 & \text{otherwise,} \end{cases}$$

where D is the domain of the binary shape represented by f. In the illustration of the Radon transform in Fig. 2.1, the shaded region represents the region D. The value of the line integral in Eq. (2.1) is equal to the length of the intersection between the line L and the shaded region.

2.1.2 Properties

The Radon transform has some properties that are beneficial for invariant pattern recognition problems as outlined below:

*P*1 *linearity*: The Radon transform is linear.

$$\mathcal{R}_{(f+g)}(\theta,\rho) = \mathcal{R}_f(\theta,\rho) + \mathcal{R}_g(\theta,\rho).$$

*P*2 *periodicity*: The Radon transform of f is periodic in the variable θ with period 2π.

$$\mathcal{R}_f(\theta,\rho) = \mathcal{R}_f(\theta + 2k\pi,\rho), \qquad \forall k \in \mathbb{Z}.$$

*P*3 *semi-symmetry*: The Radon transform of f is semi-symmetric.

$$\mathcal{R}_f(\theta,\rho) = \mathcal{R}_f(\theta \pm \pi,-\rho).$$

*P*4 *translation*: A translation of f by a vector $\vec{u} = (x_0, y_0)$ results in a shift in the variable ρ of $\mathcal{R}_f$ by a distance $d = x_0\cos\theta + y_0\sin\theta$ that is equal to the length of the projection of $\vec{u}$ onto the line $x\cos\theta + y\sin\theta = \rho$.

$$\mathcal{R}_f(\theta,\rho) \rightarrow \mathcal{R}_f(\theta,\rho - x_0\cos\theta - y_0\sin\theta).$$

*P*5 *rotation*: A rotation of f by an angle θ_0 implies a circular shift in the variable θ of $\mathcal{R}_f$ by a distance θ_0.

$$\mathcal{R}_f(\theta,\rho) \rightarrow \mathcal{R}_f(\theta + \theta_0,\rho).$$

*P*6 *scaling*: A scaling of f by a factor α results in scalings in the variable ρ and the amplitude of $\mathcal{R}_f(\theta,\rho)$ by the factors α and $\frac{1}{\alpha}$ respectively.

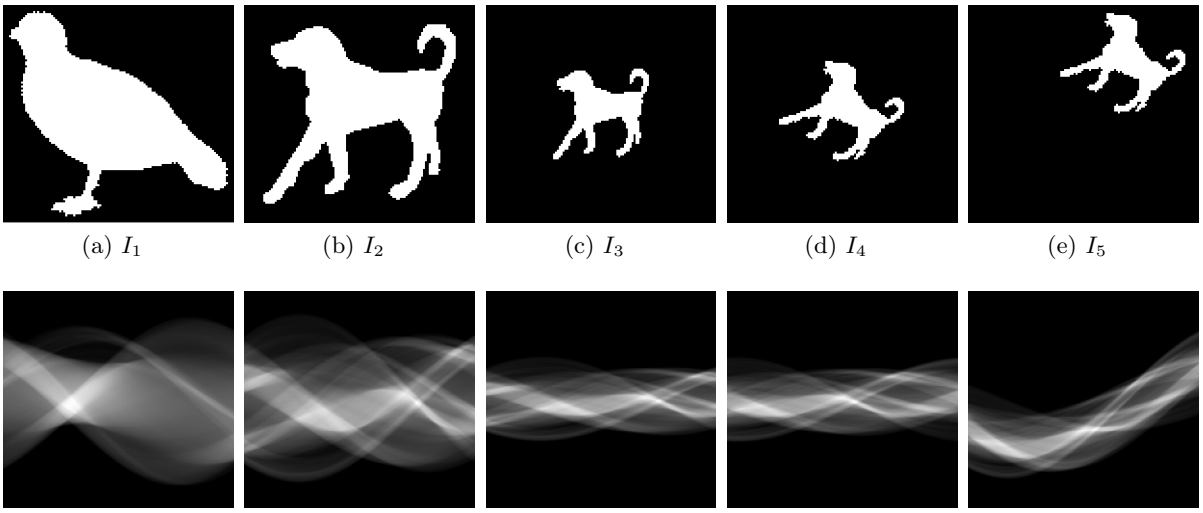
$$\mathcal{R}_f(\theta,\rho) \rightarrow \frac{1}{\alpha}\mathcal{R}_f(\theta,\alpha\rho).$$

Thus, by applying the Radon transform on an RST-transformed pattern, the RST transformation parameters are encoded in the slices of the obtained transform data [94]:

- radial slices (i.e., constant-θ slices) encode the translation and scaling parameters,
- angular slices (i.e., constant-ρ slices) encode the rotation parameter.

Current techniques usually exploit this encoded information to define invariant pattern descriptors. Fig. 2.2 illustrates the invariance properties of the Radon transform. The top row contains two original pattern images I_1 and I_2 (Figs. 2.2a and 2.2b) and the RST-transformed versions I_3, I_4, I_5 (Figs. 2.2c–2.2e) of I_2. The second row shows the Radon transforms of these five pattern images. It is observed that the Radon transforms of I_1 and I_2 are totally different while there exists resemblance between the Radon transforms of I_2, I_3, I_4, and I_5 due to the aforementioned properties *P*4–*P*6. It is observed that

- *scaling* ($I_2 \rightarrow I_3$) becomes a homogeneous compression in the radial slices,
- *rotation* ($I_3 \rightarrow I_4$) becomes a constant shift in the angular slices,
- and *translation* ($I_4 \rightarrow I_5$) becomes a sinusoidal shift in the radial slices.



(a) I_1 (b) I_2 (c) I_3 (d) I_4 (e) I_5

(f) $\mathcal{R}_{I_1}(\theta,\rho)$ (g) $\mathcal{R}_{I_2}(\theta,\rho)$ (h) $\mathcal{R}_{I_3}(\theta,\rho)$ (i) $\mathcal{R}_{I_4}(\theta,\rho)$ (j) $\mathcal{R}_{I_5}(\theta,\rho)$

Figure 2.2: Illustration of the invariance properties of the Radon transform. The first row contains two original pattern images I_1 and I_2 and the RST-transformed versions I_3, I_4, and I_5 of I_2. The second row shows the Radon transforms of these five pattern images. The intensity of these images has been rescaled to fit the display range.

2.1.3 Robustness to noise

The Radon transform over a circular domain was proven to be robust to additive white noise [107]. In this subsection, similar results on additive white/"salt & pepper" noise are developed for a rectangular domain. This naturally leads to the robustness of pattern descriptors defined based on the Radon transform to additive noise.

Additive white noise: Suppose the pattern f is corrupted by additive white noise η with zero mean and variance σ^2 to be $\hat{f}(x,y) = f(x,y) + \eta(x,y)$, the Radon transform of the noisy pattern $\hat{f}$ is obtained by applying the linearity property $(P1)$ of the Radon transform:

$$\mathcal{R}_{\hat{f}}(\theta,\rho) = \mathcal{R}_f(\theta,\rho) + \mathcal{R}_\eta(\theta,\rho).$$

Recall that the Radon transform, as illustrated in Fig. 2.1, is defined as line integrals of f along all the lines in the spatial domain. In the continuous domain, the Radon transform of additive white noise, $\mathcal{R}_\eta$, is proportional to the mean value of the noise, which means $\mathcal{R}_\eta(\theta,\rho) = 0$, or

$$\mathcal{R}_{\hat{f}}(\theta,\rho) = \mathcal{R}_f(\theta,\rho). \tag{2.2}$$

The ideal additive white noise, therefore, has no effect on the Radon transform in the continuous domain. However, in practice, the patterns represented and processed in digital systems are not continuous functions but their sampled and quantized versions; Eq. (2.2) therefore does not hold.

Suppose f is in the form of a sampled 2D signal of size $m \times n$ $(0 \le x \le m,\ 0 \le y \le n)$ whose pixels' values are random variables with mean μ and variance σ^2. The computation of the Radon transform of f is assumed to follow the definition, that is the values of f along each line $L(\theta,\rho)$ are summed up, as shown in Fig. 2.3. The contribution of each pixel i to the sum is proportional to the length of its intersection with $L(\theta,\rho)$. The sum of f along all the lines $L(\theta,\cdot)$ having the

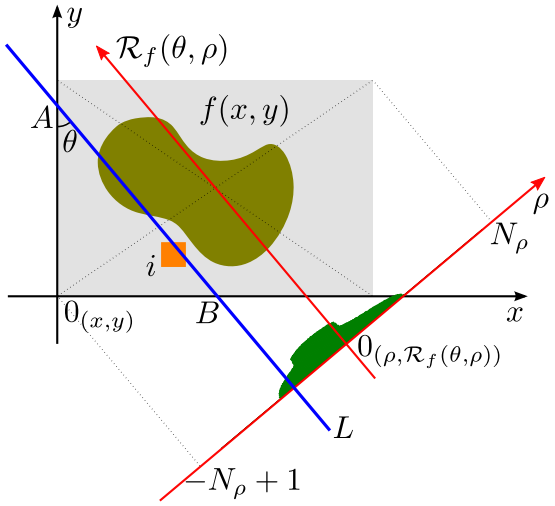


Figure 2.3: Illustration of the computation of the Radon transform by definition: for each value of θ, the function f is projected onto an axis ρ which makes an angle θ with the x axis. The projection makes itself a radial slice, $\mathcal{R}_f(\theta,\cdot)$, in the Radon transform of f.

same direction θ can also be interpreted as the projection of f onto an axis ρ that makes an angle θ with the x axis. This projection is the radial slice $\mathcal{R}_f(\theta,\cdot)$ in the Radon transform of f.

To study this projection, let $\theta = \text{const}$ and denoting $n_\rho = \overline{AB}$, then the sum of the pixel values $p_\rho = \mathcal{R}_f(\theta,\rho)$ for each line $L(\theta,\rho)$ has mean $n_\rho\mu$ and variance $n_\rho\sigma^2$. The average of the expected value of p_ρ^2 is

$$E_p = \frac{1}{2N_\rho}\int_{-N_\rho+1}^{N_\rho} E\{p_\rho^2\}\,\mathrm{d}\rho = \frac{1}{2N_\rho}\int_{-N_\rho+1}^{N_\rho} n_\rho\sigma^2\,\mathrm{d}\rho + \frac{1}{2N_\rho}\int_{-N_\rho+1}^{N_\rho} n_\rho^2\mu^2\,\mathrm{d}\rho. \tag{2.3}$$

In the above equation, the integral $\int_{-N_\rho+1}^{N_\rho} n_\rho\,\mathrm{d}\rho$ represents the area of f and is equal to the number of pixels in f, which is mn. Then, by denoting $A(\theta) = \int_{-N_\rho+1}^{N_\rho} n_\rho^2\,\mathrm{d}\rho$, Eq. (2.3) is simplified as

$$E_p = \frac{mn\sigma^2}{2N_\rho} + \frac{A(\theta)\mu^2}{2N_\rho}. \tag{2.4}$$

For the pattern image f corrupted by additive white noise η, assuming that f has mean μ_s and variance σ_s^2 and that $\eta(x,y)$ has mean $\mu_n = 0$ and variance σ_n^2, then $E_s = \frac{mn\sigma_s^2}{2N_\rho} + \frac{A(\theta)\mu_s^2}{2N_\rho}$ and $E_n = \frac{mn\sigma_n^2}{2N_\rho}$. The signal-to-noise ratios (SNR) of $\hat{f}$ and its projection along the direction θ, $\mathcal{R}_{\hat{f}}(\theta,\cdot)$, are

$$\mathrm{SNR}_{\text{image}} = \frac{\sigma_s^2+\mu_s^2}{\sigma_n^2},$$

$$\mathrm{SNR}_{\text{proj}(\theta)} = \frac{E_s}{E_n} = \frac{mn\sigma_s^2 + A(\theta)\mu_s^2}{mn\sigma_n^2} = \frac{\sigma_s^2 + \frac{A(\theta)}{mn}\mu_s^2}{\sigma_n^2},$$

or

$$\mathrm{SNR}_{\text{proj}(\theta)} = \mathrm{SNR}_{\text{image}} + \left(\frac{A(\theta)}{mn} - 1\right)\frac{\mu_s^2}{\sigma_n^2}. \tag{2.5}$$

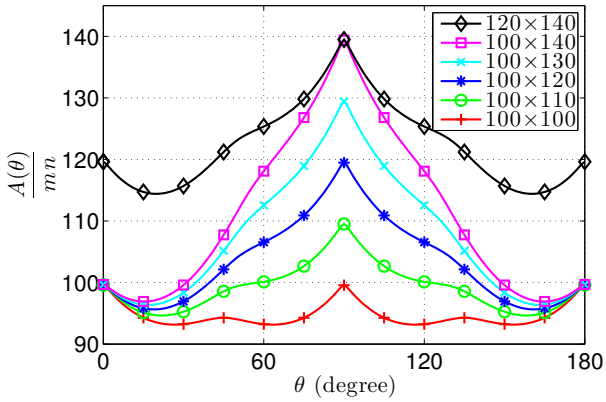


Figure 2.4: The values of the multiplicative factor, $\frac{A(\theta)}{mn}$, at different projection directions and different pattern sizes. $\frac{A(\theta)}{mn}$ gets its maximum at $\theta = 90$ (degree), which corresponds to the direction of the longer side, and its minimum near $\theta = 0$ (degree). The values of $\frac{A(\theta)}{mn}$ relate directly to the noise-suppressing ability of the Radon transform through Eqs. (2.6) and (2.7).

The SNR is increased by a quantity $\left(\frac{A(\theta)}{mn} - 1\right) \frac{\mu_s^2}{\sigma_n^2}$ after projecting $\hat{f}$ along the direction θ. As the value of $A(\theta)$ depends on both θ, m and n, the multiplicative factor $\left(\frac{A(\theta)}{mn} - 1\right)$ is not constant. Moreover, the value of $\frac{A(\theta)}{mn}$ is relatively "large" because $A(\theta) = \int_{-N_\rho+1}^{N_\rho} n_\rho^2 \, \mathrm{d}\rho$ is one-order larger than $mn = \int_{-N_\rho+1}^{N_\rho} n_\rho \, \mathrm{d}\rho$. Eq. (2.5) can then be rewritten as

$$\mathrm{SNR}_{\mathrm{proj}(\theta)} \simeq \mathrm{SNR}_{\mathrm{image}} + \frac{A(\theta)}{mn} \times \frac{\mu_s^2}{\sigma_n^2}. \tag{2.6}$$

Since $\frac{A(\theta)}{mn} \times \frac{\mu_s^2}{\sigma_n^2}$ is positive and has a large value, which corresponds to a larger increase in the value of SNR after projection, the above equation means that the Radon transform is very robust to additive white noise. Fig. 2.4 depicts the values of $\frac{A(\theta)}{mn}$ for a range of θ from 0 to 180 (degree) using input pattern images of different sizes. Notice from the figure that the value of $\frac{A(\theta)}{mn}$ depends on both the projection direction θ and the actual size of f. It gets its maximum in the direction of the longer side and its minimum near the direction of the shorter side of f:

$$\min_\theta \frac{A(\theta)}{mn} \simeq \min(m, n),$$
$$\max_\theta \frac{A(\theta)}{mn} = \max(m, n).$$

Additive "salt & pepper" noise: In the field of shape analysis and recognition, f is constrained to have binary values of 0 or 1 and the additive noise to f is in the form of "salt & pepper" noise, instead of white noise. To model this type of noise, let D and d be the percentage of pixels in $\hat{f}$ occupied by the shape region and flipped by the noise respectively. Then

$$\mu_s = D, \qquad \sigma_s^2 = D - D^2,$$
$$\mu_n = d(1 - 2D), \qquad \sigma_n^2 = d - d^2(1 - 2D)^2.$$

Using Eq. (2.4), the SNRs of $\hat{f}$ and its projection along the direction θ, $\mathcal{R}_{\hat{f}}(\theta,\cdot)$, are

$$\begin{aligned}
\mathrm{SNR}_{\mathrm{image}} &= \frac{\sigma_s^2+\mu_s^2}{\sigma_n^2+\mu_n^2} = \frac{D}{d},\\
\mathrm{SNR}_{\mathrm{proj}(\theta)} &= \frac{mn\sigma_s^2 + A(\theta)\mu_s^2}{mn\sigma_n^2 + A(\theta)\mu_n^2} = \frac{D - D^2 + \frac{A(\theta)}{mn}D^2}{d - d^2(1-2D)^2 + \frac{A(\theta)}{mn}d^2(1-2D)^2}\\
&= \frac{D}{d} \times \frac{1 + D\left(\frac{A(\theta)}{mn} - 1\right)}{1 + d(1-2D)^2\left(\frac{A(\theta)}{mn} - 1\right)},
\end{aligned}$$

or

$$\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}} = \frac{1 + D\left(\frac{A(\theta)}{mn} - 1\right)}{1 + d(1-2D)^2\left(\frac{A(\theta)}{mn} - 1\right)}. \tag{2.7}$$

It is clear that $\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}}$ depends on the size of the input noisy pattern $\hat{f}$, the projection direction θ, the percentage of shape region D, and the level of noise d. In order to estimate an explicit minimum value of $\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}}$, assuming that $D \in [0.3, 0.7]$ and $d \in [0, 0.2]$. These are practically reasonable assumptions since the binary shape usually occupies around half of the pattern area ($D = 0.5$) and the pattern is not too noisy[2]. Due to the inverse proportion of $\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}}$ to d, $\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}}$ gets its minimum value at $\max d = 0.2$. Moreover at $d = 0.2$, since $\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}}$ decreases as D goes away from the point $D = 0.5$, the minimum value of $\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}}$, at a specific value of $\frac{A(\theta)}{mn}$, is reached at $\min D = 0.3$. The depiction of the values of $\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}}$ for the case $\frac{A(\theta)}{mn} = 100$ over the domain $D \in [0.3, 0.7]$ and $d \in [0, 0.2]$ is given in Fig. 2.5a.

Fixing $D = 0.3$ and $d = 0.2$, the dependance of $\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}}$ on $\frac{A(\theta)}{mn}$ is further given in Fig. 2.5b. It is evident that $\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}} > 1$, meaning the projection in the Radon transform has the property of suppressing additive "salt & pepper" noise. Additionally, $\frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}}$ increases with the increase in $\frac{A(\theta)}{mn}$ from 4.1667 at $\frac{A(\theta)}{mn} = 20$ (a very small pattern) to its maximum value

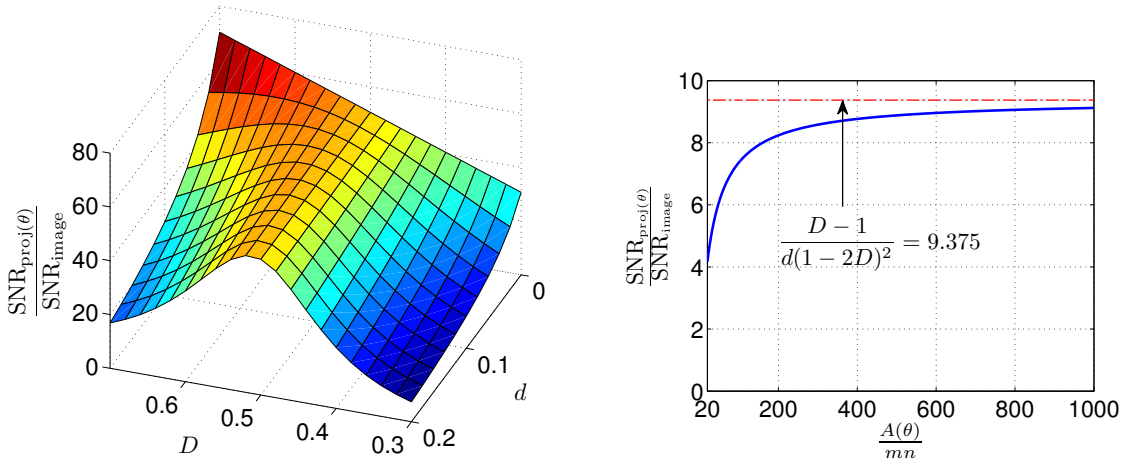
$$\lim_{\frac{A(\theta)}{mn}\to\infty} \frac{\mathrm{SNR}_{\mathrm{proj}(\theta)}}{\mathrm{SNR}_{\mathrm{image}}} = \frac{D}{d(1-2D)^2} = 9.375$$

at $\frac{A(\theta)}{mn} = \infty$. This observation implies a better suppression of additive "salt & pepper" noise in the projections of larger-sized patterns.

2.1.4 Implementation

The Radon transform as defined in Eq. (2.1) is continuous by nature; it should be adapted to discrete data in order to be used in digital systems. A seminal algorithm for the discrete Radon transform was proposed [20] utilizing projections along straight lines to compute an approximation to the Radon transform, requiring $O(N^4)$ operations for a pattern image of size

[2]The highest level of "salt & pepper" noise used in experiments is $d = 0.1$, meaning 10% of the pixels is flipped (the dataset in the rightmost column of Fig. 2.26b).



(a) $\frac{A(\theta)}{mn} = 100$ (b) $(D, d) = (0.3, 0.2)$

Figure 2.5: (a) The values of $\frac{\mathrm{SNR}_{\mathrm{proj}}}{\mathrm{SNR}_{\mathrm{image}}}$ over the domain $\{(D, d) : 0.3 \leq D \leq 0.7; 0 \leq d \leq 0.2\}$ for the case $\frac{A(\theta)}{mn} = 100$. $\frac{\mathrm{SNR}_{\mathrm{proj}}}{\mathrm{SNR}_{\mathrm{image}}}$ reaches its minimum value at one of the four corners of the plotting range. (b) The dependance of the values of $\frac{\mathrm{SNR}_{\mathrm{proj}}}{\mathrm{SNR}_{\mathrm{image}}}$ on $\frac{A(\theta)}{mn}$ for the minimum case, i.e., $(D, d) = (0.3, 0.2)$.

$N \times N$. This approach was extended [115] by constructing a discrete Radon transform that has an exact relationship with the continuous Radon transform. This algorithm, however, still has an unfavorable computational complexity of $O(N^3)$. A reduction in the computational complexity could be obtained by summing pixels' values along a set of aptly chosen discrete lines that are complete in slope and intercept [28, 89]. These approaches require only $O(N^2 \log N)$ operations and, in addition, an iterative algorithm has been developed from them to recover the original pattern with desired accuracy [180].

The same complexity of $O(N^2 \log N)$ could also be achieved by interpreting the Radon transform through the 2D Fourier transform by means of the projection-slice theorem, which states that the 1D radial slice of the Radon transform data and the 1D radial slice of the 2D Fourier transform data make a 1D Fourier transform pair:

$$\begin{aligned}
\mathcal{F}_{\mathcal{R}_f(\theta,\cdot)}(\xi) &= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)\, \delta(\rho - x\cos\theta - y\sin\theta)\, \mathrm{d}x\mathrm{d}y \right) \mathrm{e}^{-i\rho\xi}\, \mathrm{d}\rho \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)\, \mathrm{e}^{-i(x\cos\theta + y\sin\theta)\xi}\, \mathrm{d}x\mathrm{d}y \\
&= \mathcal{F}_f(\xi\cos\theta, \xi\sin\theta). \qquad (2.8)
\end{aligned}$$

Naturally, the discrete Radon transform [7] relies on the discrete version of the projection-slice theorem, which associates it with the pseudo-polar Fourier transform [6].

A more comprehensive survey on discrete Radon transform approaches could be found in [38] with $O(N^2 \log N)$ is the lowest complexity to date. Thus, whenever only the Radon transform is concerned, any implementation requiring $O(N^2 \log N)$ operations should be applicable. However, when there is a possibility of fusing the Radon transform with another transform as in the case of the RFM descriptor in Section 2.3, a lucid choice of implementation strategy may lead to some computational benefit.

Table 2.1: The influence of geometric transformations (rotation, scaling, and translation) on the Radon transform of a pattern f, summarized from properties $P4 - P6$ of the Radon transform in Subsection 2.1.2.

Geometric transformation	Influenced slice	Change in position	Change in magnitude
Rotation	θ	Circular shift	$\times$
Scaling	ρ	Scaling	$\surd$
Translation	ρ	Shift	$\times$

2.1.5 Related works

Pattern descriptors defined based on the Radon transform usually exploit its beneficial properties to be invariant to rotation, scaling, translation, or their combinations. It is pertinent to mention that the influences of each transformation on the Radon transform data are quite separated from those of other transformations as given in Table 2.1: rotation on the angular slices and scaling/translation on the radial slices. For this reason, strategies used to define invariant descriptors of a pattern from its Radon transform thus need to overcome these residual influences. Different strategies have led to different descriptors, each may or may not be totally invariant to RST transformations; and normalization comes as a natural solution to fulfill the lack of invariance to any geometric transformation.

R-transform and R-signature: A pioneer work in this direction is the R-transform, which gives rise to the R-signature [211], obtained by using an integral function and then the discrete Fourier transform on the radial and angular slices of the Radon transform data respectively. The R-transform of a 2D function f has the following definition:

$$R_{f2}(\theta) = \int_{-\infty}^{\infty} \mathcal{R}_f^2(\theta, \rho)\, \mathrm{d}\rho. \tag{2.9}$$

The integration computed on the radial slices of the Radon transform data of f makes R_{f2} invariant to translation and scaling, except for a multiplicative factor $\frac{1}{\alpha^3}$ resulting from the scaling factor α in f, and periodic with period π. Furthermore, in order to have a representation that is totally invariant to RST transformations, the magnitude of the discrete Fourier transform of the discretized R_{f2} normalized by the DC component has been used:

$$FR_{f2}(k) = \left| \frac{\sum_{n=0}^{N-1} R_{f2}(\theta_n)\, \mathrm{e}^{-\frac{2\pi i}{N} kn}}{\sum_{n=0}^{N-1} R_{f2}(\theta_n)} \right|, \qquad k = 0, 1, \ldots, N-1.$$

In this way, the conventional R-signature of f is originally defined as

$$\left[FR_{f2}(1), FR_{f2}(2), \ldots, FR_{f2}(N-1)\right]. \tag{2.10}$$

Φ-signature: Similar to the R-signature, the Φ-signature [159] is computed by using an integral function on the angular slices of the Radon transform data to get rotation invariance. The Φ-signature of a 2D function f has the following definition:

$$\Phi_f(\rho) = \int_0^{2\pi} \mathcal{R}_f(\theta, \rho)\, \mathrm{d}\theta.$$

The integration computed on the angular slices of the Radon transform data of f makes Φ_f invariant to rotation. Invariance to translation and scaling is made possible by normalizations. However, the required normalizations concerning pattern's position and size prevent the Φ-signature from being applied to noisy patterns.

HRT descriptor: A histogram of the Radon transform was also proposed in [210]. In this work, the intensity values over each radial slice of the Radon transform data are put into bins, regardless of their radial positions, in order to have a representation that is invariant to scaling and translation. The HRT descriptor of a 2D function f has the following definition:

$$\mathrm{HRT}_f(\theta,\gamma) = \frac{|\{\rho : \gamma = \mathcal{R}_f(\theta,\rho)\}|}{|\{\rho\}|},$$

where $|X|$ denotes the cardinality of a set X. The main weakness of this approach is the need to compute the "rotational distance" between the 2D HRT descriptors of f and g as

$$\mathrm{dist}(f,g) = \min_{\alpha\in[0,\pi)} \|\mathrm{HRT}_f - \mathrm{HRT}_g^{\alpha}\|_2$$

where

$$\mathrm{HRT}_g^{\alpha}(\theta,\gamma) = \mathrm{HRT}_g(\theta+\alpha,\gamma)$$

in order to overcome the problem of rotation. The computation of this distance requires circular-shifting of a 2D matrix along the angular axis for all possible values of α. The resulting process is then prohibitively slow.

R2DFM descriptor: There was an effort [227] to apply the 2D Fourier–Mellin transform [199] on the Radon transform data to get invariance to rotation and scaling. In this approach, Mellin transform and harmonic expansion are applied on the radial and angular slices of the Radon transform data respectively as

$$\mathrm{R2DFM}_f(s,k) = \int_0^{\infty}\int_0^{2\pi} \mathcal{R}_f(\theta,\rho)\,\rho^{s-1}\,\mathrm{e}^{-ik\theta}\,\mathrm{d}\theta\mathrm{d}\rho,$$

where $s=\sigma+i\tau$ with $\sigma=\mathrm{const}$. The magnitude of $\mathrm{R2DFM}_f$ is then invariant to rotation and scaling due to the invariance property of the Mellin transform and harmonic expansion. The required normalization to have translation invariance is the main weakness of this approach and hence prevents it from being applied to noisy patterns.

RCF descriptor: A set of spectral and structural features, called Radon composite features, has also been extracted from the Radon transform data for pattern description [42]. The features in this set are extracted in the ways that makes them invariant to translation. In this set, the "degree of uniformity" is essentially the R-transform that was proposed [211] and the "longest line" defined as

$$\xi_f(\theta) = \max_{\rho} \mathcal{R}_f(\theta,\rho)$$

is the information encoded in the generic R-signature described in this chapter. Normalization is used to make this set of features invariant to scaling. However, and more importantly, this set of features is not invariant to rotation and consequently, in the matching step, these features need to be rotated to all possible angles corresponding to potential pattern's orientations in order to compute patterns' similarity. Long matching time may prevent the application of this approach in real systems.

Table 2.2: Strategies used by each approach to overcome the residual influences of RST transformations on the radial and angular slices of the Radon transform of a pattern f. The symbol "×" denotes the lack of invariance to the corresponding geometric transformation and normalization should be used when necessary.

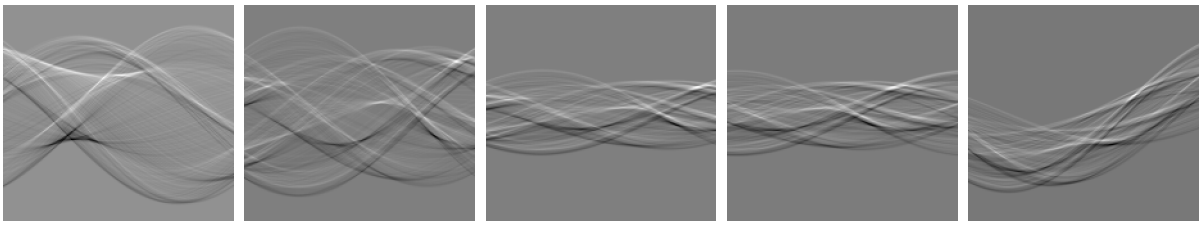
Descriptor	Rotation	Scaling	Translation
R-signature [211]	DFT	Integration	Integration
Φ-signature [159]	Integration	×	×
HRT [210]	×	Histogram	Histogram
R2DFM [227]	Fourier series	Mellin trans.	×
RCF [42]	×	×	Max, Integration, Fourier trans.
RWF [40]	DFT	×	×

RWF descriptor: Recently, a rotation-invariant descriptor was proposed [40] by using the dual-tree complex wavelet and discrete Fourier transforms on the radial and angular slices of the Radon transform data respectively. Invariance to rotation is due to the discrete Fourier transform and invariance to translation and scaling is obtained from normalizations. The dual-tree complex wavelet transform selects shift-invariant features in a multi-resolution way. Again, being invariant only to rotation limits the applicability of this approach since it cannot be used, for example, for noisy patterns.

The others: Another direction in using the Radon transform for pattern description is to extract features directly from the Radon transform data, similar to the way the Hough transform [66] is used. For example, pattern primitives in edge form are detected from the Radon transform data and represented analytically [129]. Moreover, their spatial relations can be made explicit [127] and these lead to a taxonomy of primitives for their characterization [128]. This approach, however, is quite limited since it requires that the edge primitives have analytical form. Generalizations of the Radon transform, called the trace and geometric transforms, were also proposed and used for pattern description [112, 134] by using functionals other than integral and by extending the functional domain from lines to regions delimited by closed contours. However, the application of these generalizations is restricted due to high computational complexity.

Table 2.2 summarizes the strategies used by each approach to overcome the residual influences of RST transformations on the radial and angular slices of the Radon transform of a pattern f. Among existing Radon transform-based approaches, only the R-signature is totally invariant to RST transformations by definition, all other approaches need to resort to normalizations for the lack of invariance to any geometric transformation. Additionally, even though R-signature has a low discrimination power because of the information loss in the compression process from the Radon transform data to a 1D signature, among the Radon transform-based pattern descriptors, R-signature is the most popular because of its simplicity and has been successfully applied to several applications (e.g., symbol recognition [184], activity recognition [204, 228], and orientation estimation [95]).

It is not difficult to see that the basic idea of the R-transform is the use of an integration to overcome the residual influences that remain in the radial slices of the Radon transform data caused by scaling and translation (properties $P4$ and $P6$). For any 1D function g and its scaled and then shifted version h defined as $h(x) = g(\alpha x - x_0)$, their integrations:



(a) $\frac{\partial}{\partial\rho}\mathcal{R}_{I_1}(\theta,\rho)$ (b) $\frac{\partial}{\partial\rho}\mathcal{R}_{I_2}(\theta,\rho)$ (c) $\frac{\partial}{\partial\rho}\mathcal{R}_{I_3}(\theta,\rho)$ (d) $\frac{\partial}{\partial\rho}\mathcal{R}_{I_4}(\theta,\rho)$ (e) $\frac{\partial}{\partial\rho}\mathcal{R}_{I_5}(\theta,\rho)$

Figure 2.6: The partial derivatives of the Radon transform data in the second row of Fig. 2.2 with respect to the variable ρ. For differentiation, the coefficient $\kappa_1(\alpha)$ in Eq. (2.12) takes the form $\frac{1}{\alpha}$, which depends solely on α. The intensity of these images has been rescaled to fit the display range.

$$\int_{-\infty}^{\infty} h(x)\,\mathrm{d}x = \frac{1}{\alpha}\int_{-\infty}^{\infty} g(x)\,\mathrm{d}x$$

differ by a multiplicative factor that depends solely on the scaling parameter α. This factor could be easily removed by normalization in a later processing step. This strategy could actually be extended to any operator $\mathcal{O}_2$ satisfying:

$$\mathcal{O}_2(h(\cdot)) = \kappa_2(\alpha)\,\mathcal{O}_2(g(\cdot)), \tag{2.11}$$

where $\kappa_2(\alpha)$ is a coefficient depending only on α. Possible other choices for $\mathcal{O}_2$ could be the 1D Fourier–Mellin transform (to be discussed in Section 2.3), histogram, and some measures like maxima, median, etc.

In addition to the integral operator, a square operator is also employed in the definition of the R-transform to avoid the singularities (to be discussed in Subsection 2.2.4) while preserving properties $P4$ and $P6$ of the Radon transform. This operator in turn could also be replaced by any operator $\mathcal{O}_1$ satisfying:

$$\mathcal{O}_1(h(x)) = \kappa_1(\alpha)\,\mathcal{O}_1\left(\frac{1}{\alpha}g(\alpha x - x_0)\right), \tag{2.12}$$

where $\kappa_1(\alpha)$ is a coefficient depending solely on α. Some other operators like exponentiation, differentiation could also be used for $\mathcal{O}_1$. As an example, Fig. 2.6 shows the results obtained by using differentiation of the Radon transform data in the second row of Fig. 2.2 with respect to the variable ρ. It is clear that differentiation retains properties $P4$ and $P6$ of the Radon transform and, at the same time, accentuates small variation in the Radon transform data due to sampling/quantization and additive noise.

Coming from R-transform to R-signature requires the use of the discrete Fourier transform, of which the main purpose is to get invariance to rotation. As a rotation in the spatial domain corresponds to a circular shift in the Radon transform data along the angular axis, discrete Fourier transform could be replaced by any operator $\mathcal{O}_3$ that is invariant to circular shift. Denoting $\{x_n\}$ and $\{y_n\}$ $(n = 0, 1, \ldots, N-1)$ are two sequences of complex numbers with $\{y_n\}$ is obtained by circular-shifting $\{x_n\}$ by m indices, then $\mathcal{O}_3$ should satisfy:

$$\mathcal{O}_3\big(\{x_n\}\big) = \kappa_3(m)\,\mathcal{O}_3\big(\{y_n\}\big), \tag{2.13}$$

Table 2.3: Operators employed for the proposed generic R-signature and RFM descriptor. The combined operator $\mathcal{O}_{12} = \mathcal{O}_2 \circ \mathcal{O}_1$ is applied on the radial slices whereas the operator $\mathcal{O}_3$ is applied on the angular slices of the Radon transform data.

Operator	The generic R-signature	The RFM descriptor
$\mathcal{O}_1$	Exponentiation	Identity function
$\mathcal{O}_2$	Integration	1D Fourier–Mellin transform
$\mathcal{O}_3$	Discrete Fourier transform	Discrete Fourier transform

where $\kappa_3(m)$ is a function depending only on the shifting distance m. Besides discrete Fourier transform, some other operators like Fourier series or inverse discrete-time Fourier transform could also be used for $\mathcal{O}_3$.

If there exists two operators $\mathcal{O}_1$ and $\mathcal{O}_2$ that satisfy Eqs. (2.12) and (2.11) respectively, the combined operator $\mathcal{O}_{12} = \mathcal{O}_2 \circ \mathcal{O}_1$, when applied on the radial slices of the Radon transform data, will overcome the residual influences caused by scaling and translation. The operator $\mathcal{O}_{12}$, when used in combination with the operator $\mathcal{O}_3$ that satisfies Eq. (2.13) as $\mathcal{O}_{123} = \mathcal{O}_3 \circ \mathcal{O}_2 \circ \mathcal{O}_1$, will result in a pattern descriptor that is totally invariant to RST transformations. In this manner, finding an appropriate set of operators $\{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3\}$ is the main challenge for any attempt to define an invariant pattern descriptor using the Radon transform. Despite the existence of several choices for $\mathcal{O}_1$, $\mathcal{O}_2$, and $\mathcal{O}_3$, the remaining of this chapter is devoted to the two invariant pattern descriptors, the generic R-signature and the RFM descriptor, that correspond to the two specific choices of $\{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3\}$ given in Table 2.3.

2.1.6 Contributions

In pursuing the formulation of invariant pattern descriptors defined based on the Radon transform, this chapter makes the following main contributions:

- It provides a unified view on possible directions that can be followed to define invariant pattern descriptors using the Radon transform.
- It proves theoretically that the Radon transform has the property of suppressing additive white/"salt & pepper" noise.
- It generalizes an existing Radon transform-based descriptor, the R-signature, to have the generic R-signature that is totally invariant to RST transformations.
- It proposes to apply the 1D Fourier–Mellin and Fourier transforms on the radial and angular slices of the Radon image respectively to have the RFM descriptor that is totally invariant to RST transformations.
- It shows that the two proposed invariant pattern descriptors lead to superior experimental results over comparison descriptors in terms of retrieval rate on grayscale and binary noisy pattern datasets.

The remainder of this chapter is organized as follows. The definition and theoretical analysis of the generic R-signature and RFM descriptor are presented in Sections 2.2 and 2.3 respectively. Experimental results are given in Section 2.4 and finally conclusions are drawn in Section 2.5.

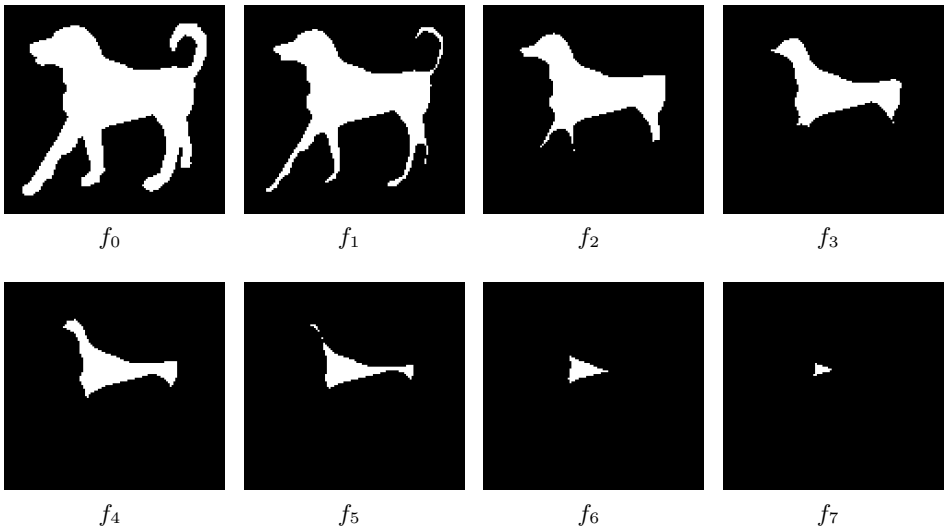


Figure 2.7: Eight shape images obtained by segmenting the distance transform of the image I_2 in Fig. 2.2b at eight equi-distant levels. The conventional R-transforms of these shape images are computed and combined in order to increase the discrimination power for shape recognition/matching.

2.2 The generic R-signature

The R-signature defined in Eq. (2.10), originally proposed for invariant shape representation, has been extended in [211] by computing $FR_{f_i 2}$, where f_i $(i = 0, 1, \ldots, 7)$ are derived from a shape f by segmenting its distance transform [24] at eight equi-distant levels. This extension leads to an increase in the discrimination power of the R-signature because the derived shapes f_i preserve the topology of f and, when i increases, the level of deformation decreases. As an example, the shape images obtained by segmenting the distance transform of the image I_2 in Fig. 2.2b are illustrated in Fig. 2.7. This extension, however, works only with silhouette shapes and it is difficult to use it with noisy patterns.

Another extension, which is orthogonal to the extension described above, is proposed in this section by generalizing the R-transform in Eq. (2.9) to further increase the discrimination power. It will be shown that the R-transform is just a special case of a class of transforms, members of which share beneficial properties for pattern representation and matching. This section provides the definition of this class of transforms, the geometric interpretation in the spatial domain, and the beneficial properties of the signatures defined based on these transforms. Furthermore, a discussion on the meaningful domain of these generic transform/signature is also carried out, along with theoretical arguments on their robustness to additive noise and their ability to encode dominant directions of patterns.

2.2.1 Definition

The generalization of the R-transform described below uses an exponentiation for $\mathcal{O}_1$ and an integration for $\mathcal{O}_2$. These choices of operators result in a generic transform that has many beneficial properties and superior performance over existing methods. For a 2D function f and $m \in \mathbb{R}$, the generic R-transform of f, denoted as R_{fm}, is defined as

$$R_{fm}(\theta) = \int_{-\infty}^{\infty} \mathcal{R}_f^m(\theta, \rho)\, \mathrm{d}\rho. \tag{2.14}$$

Evidently, by setting $m = 2$, R_{fm} in the above equation becomes R_{f2} in Eq. (2.9). The utilization of the exponent m as a parameter makes R_{fm} a generic version of R_{f2}. Furthermore, by varying the value of m, a class of transforms is obtained and this in turn results in a class of signatures. The derivation of the generic R-signature, FR_{fm}, from R_{fm} follows strictly Eq. (2.10) in Subsection 2.1.5.

2.2.2 Geometric interpretation

Recall that the value of $\mathcal{R}_f(\theta, \rho)$ is the result of a line integral of f along the line $L(\theta, \rho)$ parameterized by (θ, ρ). Consequently, the generic R-transform defined based on $\mathcal{R}_f$ by computing an integral over the variable ρ has some geometric interpretations as follows.

The generic R-transform of f, R_{fm}, in Eq. (2.14) is basically an integral of $\mathcal{R}_f^m$ computed over the variable ρ of the Radon transform data. In other words, this integral is computed by using the results of line integrals along all the lines parameterized by a fixed value of θ and different values of ρ. Sharing the same value of θ means that these lines are parallel in the spatial domain (as depicted in Fig. 2.8) and $R_{fm}(\theta)$ encodes the spatial information of the pattern f in the direction making an angle θ with the y axis. Encoding f at different directions is possible by varying θ to have R_{fm} and R_{fm} could then be interpreted as containing the encoded spatial information of f at all directions.

The role of the exponent m in Eq. (2.14), besides setting up a class of transforms, is to make R_{fm} discriminative at different values of m by exploiting the variation in the constant-θ slices of $\mathcal{R}_f$, which in turn is the variation in the accumulations of f along all parallel lines L_i making an angle θ with the y axis (for a binary shape f, it is the variation in the lengths of the intersections of f with all parallel lines L_i). Evidently, at $m = 2$, R_{fm} has the same interpretation and discrimination power as those of the conventional R-transform. The interest here lies in large m at which R_{fm} has the capability to encode the dominant direction or the "longest line" that will be demonstrated in Subsection 2.2.5. In addition, due to the singularity at $m = 1$ (to be discussed Subsection 2.2.4), it is anticipated that the generic R-transform will have a higher discrimination power when the value of m goes farther from 1. Inversely, when $m < 1$, R_{fm} weights more on shorter lines.

2.2.3 Properties

The generic R-transform defined in Eq. (2.14) has some beneficial properties as follows:

- *Periodicity*: The generic R-transform of f is periodic in the θ coordinate with period π. Using property $P3$ of the Radon transform:

$$\begin{aligned} R_{fm}(\theta) &= \int_{-\infty}^{\infty} \mathcal{R}_f^m(\theta, \rho)\, d\rho = \int_{-\infty}^{\infty} \mathcal{R}_f^m(\theta \pm \pi, -\rho)\, d\rho = -\int_{\infty}^{-\infty} \mathcal{R}_f^m(\theta \pm \pi, \upsilon)\, d\upsilon \\ &= \int_{-\infty}^{\infty} \mathcal{R}_f^m(\theta \pm \pi, \upsilon)\, d\upsilon = R_{fm}(\theta \pm \pi), \end{aligned}$$

  where $\upsilon = -\rho$.

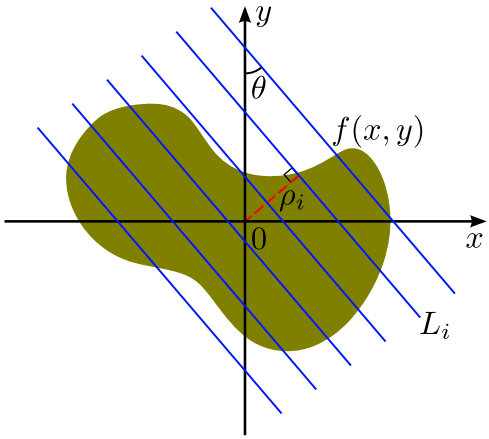


Figure 2.8: Geometric illustration of the generic R-transform of a function f. Depicted in the figure is the set of all parallel lines $L(\theta, \cdot)$ used to compute the value of $R_{fm}(\theta)$ in Eq. (2.14). $R_{fm}(\theta)$ contains the encoded spatial information of the pattern in the direction making an angle θ with the y axis.

- *Translation*: The generic R-transform of f is invariant to translation. Assuming that f is translated by a vector $\vec{u} = (x_0, y_0)$, using property $P4$ of the Radon transform:

$$\begin{aligned} R_{f'm}(\theta) &= \int_{-\infty}^{\infty} \mathcal{R}^m_{f'}(\theta, \rho)\, d\rho = \int_{-\infty}^{\infty} \mathcal{R}^m_{f}(\theta, \rho - x_0 \cos\theta - y_0 \sin\theta)\, d\rho \\ &= \int_{-\infty}^{\infty} \mathcal{R}^m_{f}(\theta, \upsilon)\, d\upsilon = R_{fm}(\theta), \end{aligned}$$

  where $\upsilon = \rho - x_0 \cos\theta - y_0 \sin\theta$.

- *Rotation*: A rotation of f by an angle θ_0 implies a circular shift of R_{fm} by a distance θ_0. Assuming that f is rotated by an angle θ_0, using property $P5$ of the Radon transform:

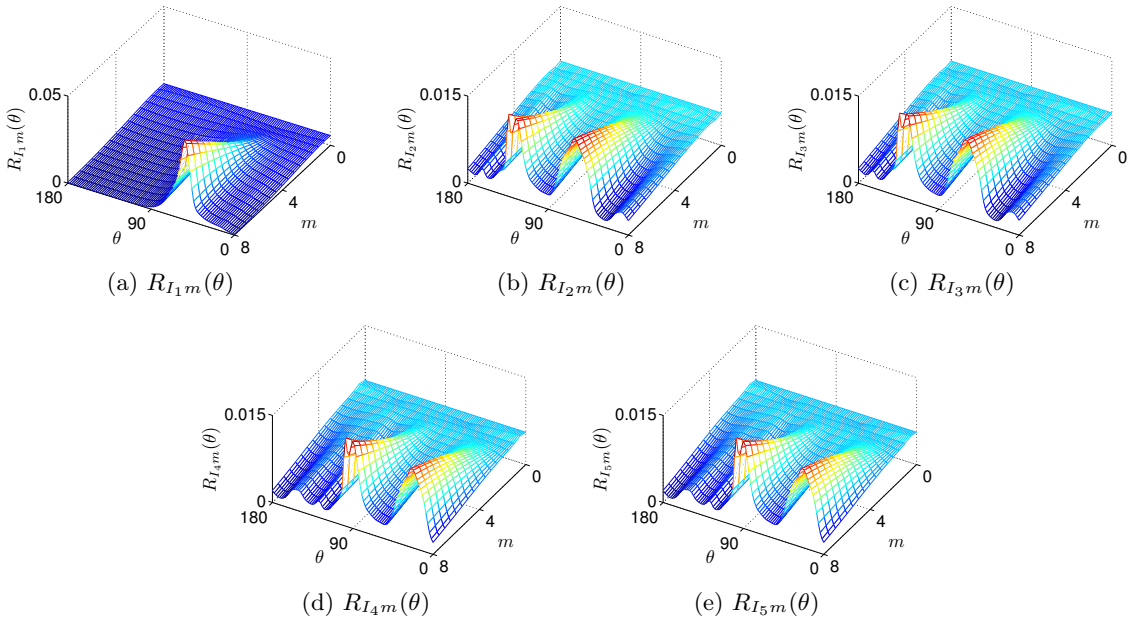
$$R_{f'm}(\theta) = \int_{-\infty}^{\infty} \mathcal{R}^m_{f'}(\theta, \rho)\, d\rho = \int_{-\infty}^{\infty} \mathcal{R}^m_{f}(\theta + \theta_0, \rho)\, d\rho = R_{fm}(\theta + \theta_0).$$

- *Scaling*: A scaling of f by a factor α results in a scaling in the amplitude of R_{fm} by a factor $\frac{1}{\alpha^{m+1}}$. Assuming that f is scaled by a factor α, using property $P6$ of the Radon transform:

$$\begin{aligned} R_{f'm}(\theta) &= \int_{-\infty}^{\infty} \mathcal{R}^m_{f'}(\theta, \rho)\, d\rho = \int_{-\infty}^{\infty} \frac{1}{\alpha^m} \mathcal{R}^m_{f}(\theta, \alpha\rho)\, d\rho \\ &= \frac{1}{\alpha^{m+1}} \int_{-\infty}^{\infty} \mathcal{R}^m_{f}(\theta, \upsilon)\, d\upsilon = \frac{1}{\alpha^{m+1}} R_{fm}(\theta), \end{aligned} \tag{2.15}$$

  where $\upsilon = \alpha\rho$.

From these properties, it is straightforward that the generic R-signature FR_{fm} of f defined based on the generic R-transform R_{fm} of f as in Eq. (2.10) in Subsection 2.1.5 is totally invariant to RST transformations. Illustration of the properties concerning RST transformations of the generic R-transform is given in Fig. 2.9 using patterns in the top row of Fig. 2.2. The value of R_{I_km} with $k = 1 \to 5$ has been normalized by the area they make with the θ axis, $\int_0^\pi R_{I_km}(\theta)\, d\theta$, for better viewing. The two patterns I_1 and I_2 in Figs. 2.2a, 2.2b are not similar and as a



(a) $R_{I_1m}(\theta)$ (b) $R_{I_2m}(\theta)$ (c) $R_{I_3m}(\theta)$

(d) $R_{I_4m}(\theta)$ (e) $R_{I_5m}(\theta)$

Figure 2.9: Illustration of the invariance properties of the generic R-transform using patterns in the top row of Fig. 2.2 with the exponent $m = 0 \rightarrow 6$. The generic R-transform is invariant to translation and scaling, and converts a rotation in the spatial domain into a circular shift in the generic R-transform by a distance equal to the rotation angle.

consequence R_{I_1m} and R_{I_2m} in Figs. 2.9a and 2.9b have different surfaces. The patterns I_3, I_4, and I_5 in Figs. 2.2c–2.2e are transformed versions of the pattern I_2 in Fig. 2.2b then R_{I_3m}, R_{I_4m}, and R_{I_5m} in Figs. 2.9c–2.9e have surfaces that are similar to that of R_{I_2m} in Fig. 2.9b. In addition, it is evident from the figure that the generic R-transform is invariant to scaling ($R_{I_2m} \simeq R_{I_3m}$) and translation ($R_{I_4m} \simeq R_{I_5m}$). It converts a rotation in the spatial domain into a circular shift in the generic R-transform by a distance equal to the rotation angle ($R_{I_3m} \rightarrow R_{I_4m}$).

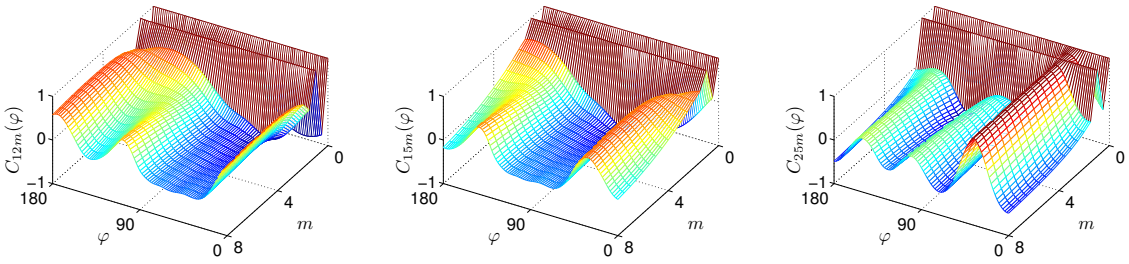
A quantitative evaluation of the invariance properties of the generic R-transform is given in Fig. 2.10 using the normalized cross-correlation between three possible pairs of the generic R-transforms R_{I_1m}, R_{I_2m}, and R_{I_5m} from Fig. 2.9. Normalized cross-correlation is selected for the sake of overcoming the constant multiplicative factor $\frac{1}{\alpha^{m+1}}$ in Eq. (2.15) and the remaining rotation. To overcome the rotation parameter, at a specific value of m, the correlation is calculated for all possible rotation angles, meaning that one of the two generic R-transforms is circular-shifted by 180 possible distances from 0 to 179 with increment of 1 before computing the correlation. Denoting φ as the shifting distance, the correlation between R_{I_im} and R_{I_jm} at φ is defined as

$$C_{ijm}(\varphi) = \text{corr}\left(R_{I_im}, R^{\varphi}_{I_jm}\right),$$

where $R^{\varphi}_{I_jm}(\theta) = R_{I_jm}(\theta + \varphi)$ and $\text{corr}(A, B)$ is the normalized cross-correlation function between two input vectors A and B of length n calculated using the following formula:

$$\text{corr}(A, B) = \frac{\sum_{i=1}^{n}(A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\left(\sum_{i=1}^{n}(A_i - \bar{A})^2\right)\left(\sum_{i=1}^{n}(B_i - \bar{B})^2\right)}},$$

where $\bar{A}$ and $\bar{B}$ are the mean values of A and B respectively.



(a) $C_{12m}(\varphi)$ (b) $C_{15m}(\varphi)$ (c) $C_{25m}(\varphi)$

Figure 2.10: The normalized cross-correlation between the three possible pairs of the generic R-transforms R_{I_1m}, R_{I_2m}, and R_{I_5m} from Fig. 2.9. For each pair of generic R-transforms and at a specific value of m, 180 correlation values are calculated after circular-shifting one of the two generic R-transforms by 180 possible distances from 0 to 179 with increment of 1.

The three surfaces C_{ijm} that correspond to the three possible pairs of three generic R-transforms have some distinct characteristics. Firstly, they all have two constant bars at $m = 0, 1$ due to the singularities of the generic R-transform at $m = 0, 1$ (to be discussed in Subsection 2.2.4). Secondly, at a specific value of m, the maximum value of C_{25m} is almost 1 while those of C_{12m} and C_{15m} are always less than 0.67. It thus can be concluded that C_{ijm} is peaky only when the two patterns I_i and I_j are similar. The non-peaky and peaky maxima exhibit the discrimination power of the proposed descriptor and the maximum of nearly 1 means, in this case, that the generic R-transform is invariant to translation and scaling. Moreover, the value of φ that corresponds to the peak in C_{ijm} denotes the difference in orientation (in degree) between the two patterns I_i and I_j.

2.2.4 The domain of m

The generic R-transform defined in Eq. (2.14) theoretically produces a class of transforms having an infinite number of members obtained by varying the value of the exponent m. However, in reality, the domain of reasonable values of m is limited, not all the space $\mathbb{R}$, due to the existence of singularities in the generic R-transform and the sensitivity of the generic R-transform to sampling/quantization and additive noise.

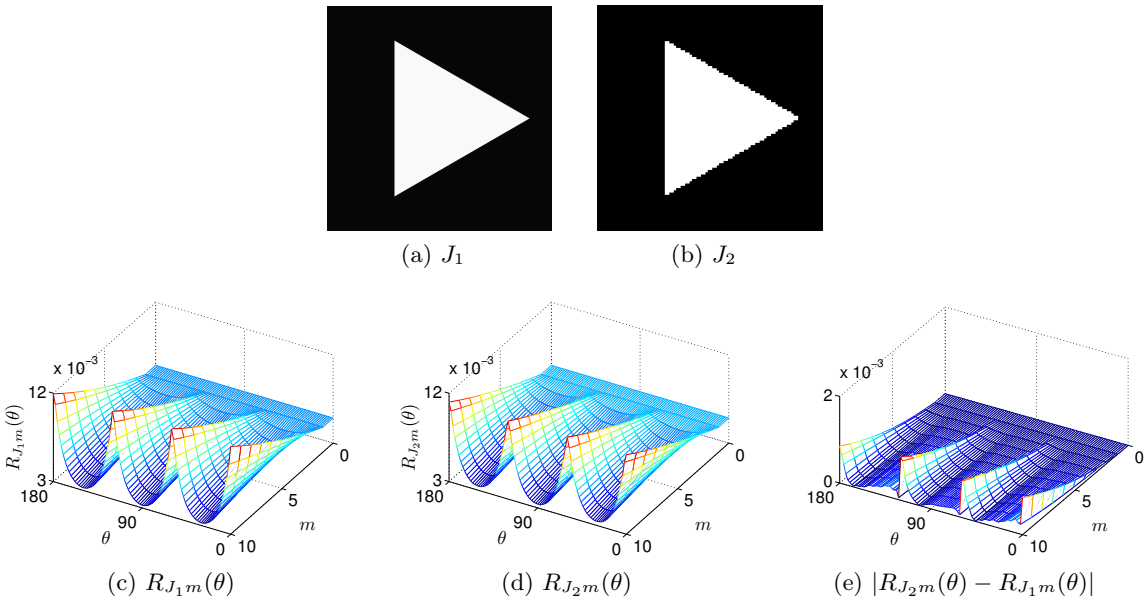
Singularities

The generic R-transform has two singularities at $m = 0$ and $m = 1$:

$$R_{f0}(\theta) = \int_{-\infty}^{\infty} \mathcal{R}_f^0(\theta, \rho)\, \mathrm{d}\rho = \rho_{\max} - \rho_{\min} = \text{const},$$

$$R_{f1}(\theta) = \int_{-\infty}^{\infty} \mathcal{R}_f^1(\theta, \rho)\, \mathrm{d}\rho = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\, \mathrm{d}x\mathrm{d}y = m_{00} = \text{const},$$

where m_{00} is the zeroth-order moment of f. R_{f0} and R_{f1} hence contain no discriminative information about f, except for scaling when $m = 1$, and thus should not be used to represent patterns for the purpose of recognition/matching. Additionally, when m reaches $+\infty$, as the cumulative sum of f along a line $L(\theta, \rho)$ is most of the time greater than 1, then



(a) J_1 (b) J_2

(c) $R_{J_1m}(\theta)$ (d) $R_{J_2m}(\theta)$ (e) $|R_{J_2m}(\theta) - R_{J_1m}(\theta)|$

Figure 2.11: Illustration of the sensitivity of the generic R-transform to sampling and quantization using an analytical triangle J_1 and its sampled and quantized version J_2 of size 100×100. A larger value of m will result in a larger difference between R_{J_1m} and R_{J_2m}, meaning a more severe deformation in R_{J_2m}.

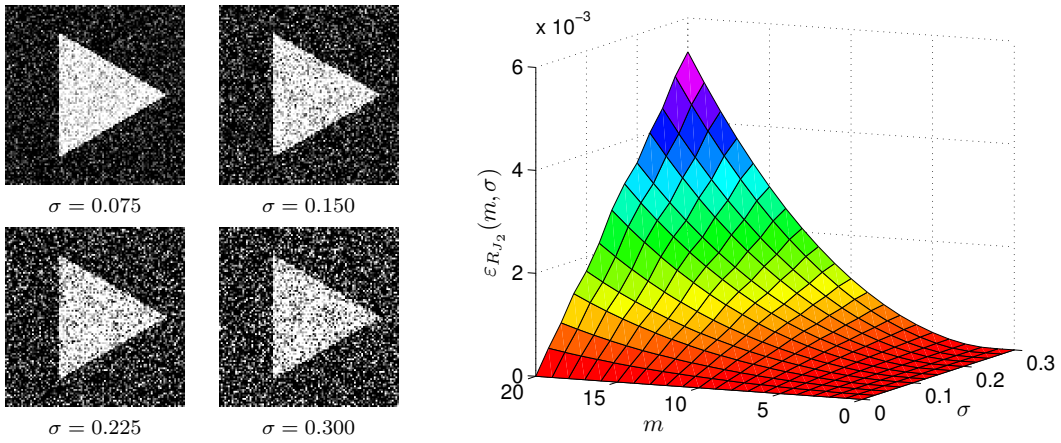
$$\lim_{m\to+\infty} R_{fm}(\theta) = \lim_{m\to+\infty} \int_{-\infty}^{\infty} \mathcal{R}_f^m(\theta,\rho)\,\mathrm{d}\rho = +\infty.$$

Even though $m = \infty$ has no practical meaning, the above equation implies that in practice m cannot take an excessive large value. Furthermore, negative value of m should also be avoided due to the sensitivity of negative power function to very small values. More precisely, at the pattern's furthest point from the pattern centroid's position, the intersection between the tangent line $L(\theta^\star, \rho^\star)$ and the pattern f has infinitesimal length, inducing a very small value in $\mathcal{R}_f(\theta^\star, \rho^\star)$. Taking negative power of this value produces a very large value and is sometimes out of the representing capability of digital systems.

Sensitivity to sampling/quantization and additive noise

By definition, the Radon transform is essentially the projection of the pattern f along all the lines in the spatial domain. Due to this projection, the Radon transform has the ability to suppress variation in the pattern. However, as the generic R-transform is defined based on exponentiation, the remaining variation in the Radon transform data due to noise will result in variation in R_{fm} at different levels according to the value of the exponent m. A too large value of m will cause a high variation in R_{fm} and make it very different from the ideal analytical one. The heavily deformed R_{fm} due to noise will make the representation inappropriate for recognition/matching.

Sampling and quantization could be considered as processes that add noise to the original patterns. In this sense, the patterns processed in digital systems are noisy and the variation in their Radon transforms is unavoidable. Fig. 2.11 illustrates the sensitivity of the generic R-transform to sampling and quantization. The pattern image J_2 of size 100×100 in Fig. 2.11b



(a) Samples of noisy pattern images (b) $\varepsilon_{R_{J_2}}(m,\sigma)$

Figure 2.12: The dependence on the noise level σ and the exponent m of the average difference between the generic R-transform of a noise-free pattern and those of its noisy versions. (a) Samples of noisy pattern images generated from the pattern J_2 in Fig. 2.11b. (b) The average difference between the generic R-transforms, $\varepsilon_{R_{J_2}}(m,\sigma)$.

is a sampled and quantized version of an analytical triangle J_1 in Fig. 2.11a. The values of $R_{J_k m}$ with $k = 1$ and 2 in Figs. 2.11c and 2.11d have been normalized by the area they make with the θ axis, $\int_0^\pi R_{J_k m}(\theta)\, d\theta$, for better viewing. The difference in the normalized values of the generic R-transform of J_1 and J_2, $|R_{J_2 m} - R_{J_1 m}|$, in Fig. 2.11e shows clearly that a larger value of m will result in a larger difference, meaning a more severe deformation in $R_{J_2 m}$.

As the noise resulting from sampling and quantization is relatively small that may not demonstrate well the sensitivity of the generic R-transform to additive noise in general. A study has been carried out using noisy patterns generated from J_2 in Fig. 2.11b by adding white noise of different variances σ^2 to it. At each value of σ, 100 noisy images are generated for the computation of the average difference between the generic R-transform of J_2 and those of these noisy patterns:

$$\varepsilon_{R_{J_2}}(m,\sigma) = \frac{1}{|N_\sigma|} \sum_{J_{2i} \in N_\sigma} \int |R_{J_2 m}(\theta) - R_{J_{2i} m}(\theta)|\, \mathrm{d}\theta,$$

where N_σ is the subset of noisy images generated from J_2 having the same variance σ^2. Samples of noisy pattern images generated from J_2 are given in Fig. 2.12a and the values of $\varepsilon_{R_{J_2}}(m,\sigma)$ are plotted in Fig. 2.12b over a range of m and σ. It is observed that $\varepsilon_{R_{J_2}}(m,\sigma)$ increases with both σ and m, meaning that a larger value of σ and/or m will result in a more severe deformation in $R_{J_2 m}$. However, the increasing trend of $\varepsilon_{R_{J_2}}(m,\sigma)$ due to σ is different from that due to m: $\varepsilon_{R_{J_2}}(m,\sigma)$ tends to increase linearly with σ but exponentially with m. It is thus anticipated that the degradation in performance of the generic R-signature in invariant pattern recognition problems due to additive noise is linear with σ and exponential with m.

2.2.5 Robustness to noise

The Radon transform has been proven to be robust to additive noise due to the use of an integral function along straight lines in the spatial domain. For the case of the generic R-transform, the use of exponentiation in its definition in Eq. (2.14) destroys the linearity and the uncorrelation between

signal and noise. This in turn excludes the possibility of analyzing signal and noise separately and hinders the formulation of SNR_R, i.e., the SNR of the generic R-transform. Nevertheless, the following arguments give some intuition on the sensitivity of SNR_R to the exponent m.

Assuming that $m > 1$ and f is a scalar signal contaminated by noise η, the SNR before and after exponentiation are

$$\mathrm{SNR}_b = \frac{f^2}{\eta^2} \quad \text{and} \quad \mathrm{SNR}_a = \frac{f^2}{((f+\eta)^m - f^m)^2}$$

respectively. It is not difficult to see that

$$\frac{\mathrm{SNR}_a}{\mathrm{SNR}_b} = \frac{\eta^2}{((f+\eta)^m - f^m)^2}$$

decreases exponentially with the increase in m or, in other words, the decrease in SNR due to exponentiation depends exponentially on the exponent m. In the case of the generic R-transform, the integral after exponentiation has a smoothing property which alleviates the problem, especially for large-sized images, similar to the smoothing property of the Radon transform's projection discussed in Subsection 2.1.3. However, the compensation is relatively small that the decrease in SNR still exists when m is reasonably large. This observation leads to a conclusion that a larger value of m will result in a smaller value in SNR_R. An experimental support for this conclusion can be observed from Fig. 2.12b where the average difference $\varepsilon_{R_{J_2}}(m, \sigma)$ increases exponentially with m.

Apart from the above observation, the robustness to noise of the generic R-signature has its roots not only from the noise-suppressing property of the Radon transform but also from the ability of the generic R-transform to encode the dominant directions of patterns. Due to the exponentiation insides the integral in Eq. (2.14), the contribution of $\Psi_f(\theta) = \mathcal{R}_f(\theta, \rho^\star)$ with $\rho^\star = \mathrm{argmax}_\rho \mathcal{R}_f(\theta, \rho)$ to $R_{fm}(\theta)$ increases as m increases, and furthermore

$$\lim_{m \to +\infty} \frac{\Psi_f^m(\theta)}{R_{fm}(\theta)} = 1. \tag{2.16}$$

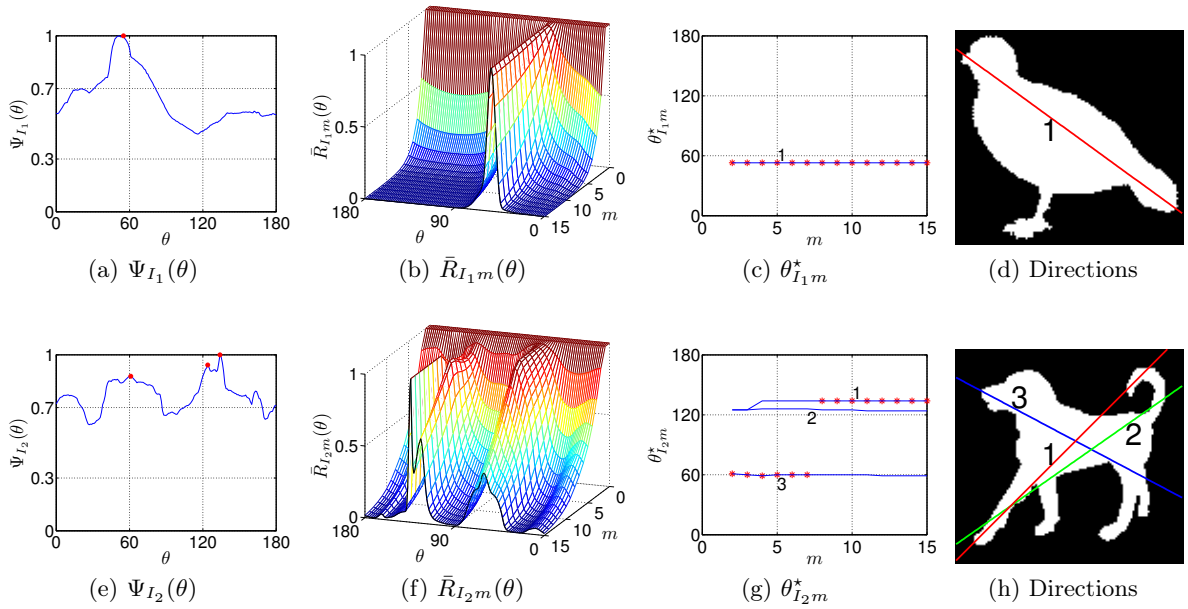
This means that, at a reasonable large value of m, $R_{fm}(\theta)$ represents the highest accumulation of f along all the parallel lines that makes an angle θ with the y axis, which is similar to the "longest line" feature proposed in [42]. The highest accumulation profiles Ψ_{I_1} and Ψ_{I_2} of the patterns I_1 and I_2 in Figs. 2.2a and 2.2b) (normalized by $\max_\theta \Psi_{I_1}(\theta)$ and $\max_\theta \Psi_{I_2}(\theta)$ for better viewing) are plotted in Figs. 2.13a and 2.13e respectively.

Similarly, by denoting $\bar{R}_{fm}(\theta) = \frac{R_{fm}(\theta)}{R_{fm}(\theta^\star_{fm})}$ where $\theta^\star_{fm} = \mathrm{argmax}_\theta R_{fm}(\theta)$ as the normalization of $R_{fm}(\theta)$, it is evident that $\bar{R}_{fm}(\theta)$ with $\theta \neq \theta^\star_{fm}$ decreases exponentially with the increase in m, and

$$\lim_{m \to +\infty} \bar{R}_{fm}(\theta) = \lim_{m \to +\infty} \frac{R_{fm}(\theta)}{R_{fm}(\theta^\star_{fm})} = \delta_{\theta\theta^\star_{fm}}, \tag{2.17}$$

where $\delta_{\theta\theta^\star_{fm}} = [\theta = \theta^\star_{fm}]$ is the Kronecker delta function. Thus, when m is reasonably large, the function $\bar{R}_{fm}$ encodes only the direction $\theta^\star_{fm}$, called the principle direction. Combining Eqs. (2.16) and (2.17) leads to a conclusion that $\bar{R}_{fm}(\theta^\star_{fm})$ corresponds to the highest accumulation of f along all the lines in the spatial domain at a reasonable large value of m.

In real applications, Eqs. (2.16) and (2.17) do not hold since m does not have a large enough value due to the sensitivity of the generic R-transform to quantization/sampling and additive



(a) $\Psi_{I_1}(\theta)$ (b) $\bar{R}_{I_1 m}(\theta)$ (c) $\theta^{\star}_{I_1 m}$ (d) Directions

(e) $\Psi_{I_2}(\theta)$ (f) $\bar{R}_{I_2 m}(\theta)$ (g) $\theta^{\star}_{I_2 m}$ (h) Directions

Figure 2.13: Illustration of the ability of the generic R-transform to encode pattern's dominant directions using two patterns I_1 and I_2 in Figs. 2.2a and 2.2b respectively. Ψ_{I_i} (first column) represent the highest accumulations of I_i along all the parallel lines that make an angle θ with the y axis and $\theta^{\star}_{I_i m}$ (third column) represent the ridges in the surfaces of $\bar{R}_{I_i m}$ (second column). The fourth column shows the one and three dominant directions of I_1 and I_2 that correspond to the one and three ridges in $\bar{R}_{I_1 m}$ and $\bar{R}_{I_2 m}$ respectively.

noise as discussed in Subsection 2.2.4. However, there is an evolution in the profile of $\bar{R}_{fm}$ as m increases, transforming a constant function ($m = 0, 1$) into the Kronecker delta function ($m = \infty$). During this process, the information encoded by $\bar{R}_{fm}$ also changes, roughly from all directions to a single direction $\theta^{\star}_{fm}$. The interpretation here is that the dominant directions of f are encoded at different levels, depending on m. Illustration of this evolution is given in Figs. 2.13b and 2.13f containing the plots of $\bar{R}_{I_1 m}$ and $\bar{R}_{I_2 m}$. The traces of the ridges in $\bar{R}_{I_1 m}$ and $\bar{R}_{I_2 m}$ are plotted in blue lines and the values of $\theta^{\star}_{I_1 m}$ and $\theta^{\star}_{I_2 m}$ are denoted by red asterisks in Figs. 2.13c and 2.13g respectively. It is observed that the one and three ridges in the surfaces of $\bar{R}_{I_1 m}$ and $\bar{R}_{I_2 m}$ correspond to the one and three local maxima of Ψ_{I_1} and Ψ_{I_2} in Figs. 2.13a and 2.13e, which in turn represent the one and three dominant directions of I_1 and I_2, as shown in Figs. 2.13d and 2.13h respectively. For the pattern I_1, as there exists only one ridge, the principal direction always coincides with that ridge. In the case of the pattern I_2, as m increases, the roles of the three maxima in $\bar{R}_{I_2 m}$ interchange along with a change in the principal direction from ridge 3 to ridge 1 at $m = 7$. However, the dominant directions are still reflected in the surface of $\bar{R}_{I_2 m}$ as the local maxima of its three ridges.

When the pattern f is contaminated by additive noise to be $\hat{f}$, due to the noise-suppressing property of the Radon transform and the sensitivity of the generic R-transform to m, the difference in the dominant directions of f and $\hat{f}$ is negligible when m is not too large. For this reason, $\bar{R}_{fm}$ and the generic R-signature can be used to estimate the orientation of patterns and to recognize noisy patterns respectively when m is not too large. Combining this observation with the dependance of the discrimination power of the generic R-transform on m that has been discussed in Subsection 2.2.2, it can be concluded here that the selected value of m is a compromise between

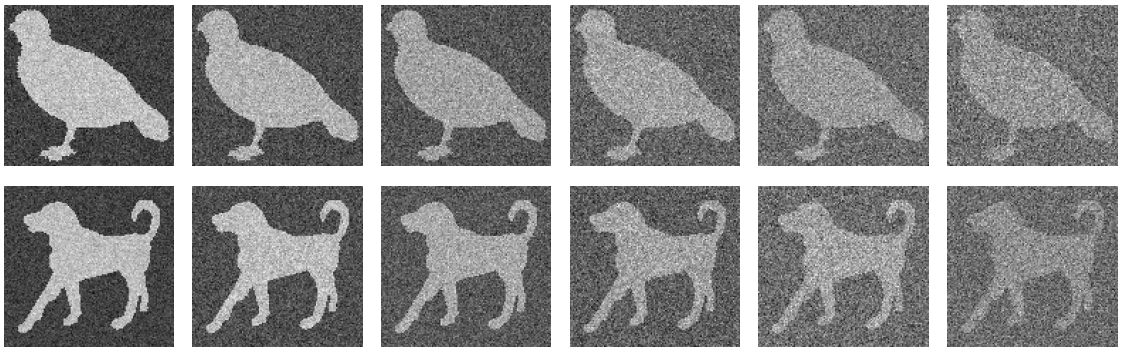


Figure 2.14: Example pattern images of different SNR from the datasets *OriA* (top row) and *OriB* (bottom row) for principal direction estimation using the generic R-transform. These patterns are generated by adding white noise to the two patterns I_1 and I_2 in Figs. 2.2a and 2.2b respectively.

two contradicting desires: a larger value is preferred for a higher discrimination power whereas a smaller one is for noise robustness. These theoretical analysis and observations will be supported by experimental evidence in Section 2.4.

The principal directions of patterns

The stability of the estimated principal directions of patterns by the generic R-transform to additive noise has been evaluated on the two datasets *OriA* and *OriB* of noisy images generated from the two patterns I_1 and I_2 in Figs. 2.2a and 2.2b respectively by adding white noise to them. These two patterns have been chosen because they are representatives of two classes of patterns: I_1 belongs to the "easy" class while I_2 belongs to the "difficult" class. The difficulty with I_2 is due to the existence of the three ridges in the plot of $\bar{R}_{I_2 m}$ in Fig. 2.13f. Let SNR be the signal-to-noise ratio defined as

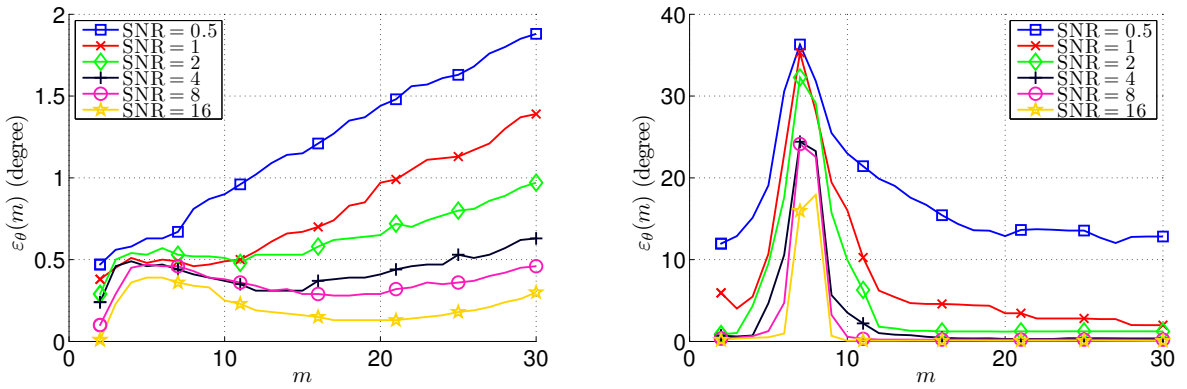
$$\text{SNR} = \frac{\sum_{x,y} f^2(x,y)}{\sum_{x,y} \eta^2(x,y)},$$

where f is the noise-free pattern and η is the added white noise. Each dataset contains 600 noisy patterns of six possible values of SNR $= \{0.5, 1, 2, 4, 8, 16\}$, meaning 100 patterns for each SNR. Example pattern images of different SNR from these two datasets are given in Fig. 2.14: top row for *OriA* and bottom row for *OriB*. The generic R-transforms of the noise-free patterns I_1, I_2 and all the noisy patterns in *OriA* and *OriB* have been computed along with their principal directions $\theta^\star_{fm}$ for evaluation.

The adopted evaluation criterion is the average difference between the estimated principal directions of all noisy patterns of the same SNR in one dataset and that of their corresponding noise-free pattern as

$$\varepsilon_\theta(m) = \frac{1}{|N_k|} \sum_{f_i \in N_k} |\theta^\star_{fm} - \theta^\star_{f_i m}| \quad \text{with } k \in \{0.5, 1, 2, 4, 8, 16\},$$

where f is a noise-free pattern and N_k is the subset of all noisy patterns generated from f having SNR $= k$. This criteria measures statistically the effect of additive white noise on the accuracy of the estimated principal directions at different noise levels and at different exponents m. Shown in



(a) *OriA* (b) *OriB*

Figure 2.15: The average difference in the estimated principal directions $\varepsilon_\theta(m)$ between a noise-free pattern and its noisy versions of different SNR of the two datasets *OriA* (a) and *OriB* (b). A small value of $\varepsilon_\theta(m)$ means that the principal directions estimated by the generic R-transform are very stable under noise perturbation.

Fig. 2.15 are the plots of the computed $\varepsilon_\theta(m)$ at different SNR for the two datasets *OriA* and *OriB*, it is observed that:

- *OriA*: The values of $\varepsilon_\theta(m)$ shown in Fig. 2.15a are generally small ($\varepsilon_\theta(m) < 2^\circ$ for $m < 30$), demonstrating the stability of the estimated principal direction $\theta^\star_{I_1 m}$. Additionally, the values of $\varepsilon_\theta(m)$ increase with the increase in m and high accuracy ($\varepsilon_\theta(m) < 0.5^\circ$) is obtained when $m \simeq 2$ for all SNR. Explanation for this, besides the noise-suppressing property of the Radon transform, comes from the "averaging" phenomena in Eq. (2.14) when m is small, allowing the accumulations of f along all the parallel lines that make an angle θ with the y axis to participate in $\bar{R}_{fm}(\theta)$ and thus further reducing the effect of additive white noise. When m increases, this "averaging" phenomena gradually disappears as the role of $\mathcal{R}_f(\theta, \rho^\star)$ in $\bar{R}_{fm}(\theta)$ gradually dominates.

- *OriB*: The values of $\varepsilon_\theta(m)$ shown in Fig. 2.15b have a different trend from those of $\varepsilon_\theta(m)$ in Fig. 2.15a; $\varepsilon_\theta(m)$ has its peak at $m = 7$ for almost all SNR. This is due to the existence of the three ridges in the plot of $\bar{R}_{I_2 m}$ in Fig. 2.13f: the role of encoding the principal direction of the pattern I_2 changes from ridge 3 to ridge 1 at $m = 7$ (shown in Fig. 2.13g) whereas, in the presence of additive noise, the changing point is not always at $m = 7$. Additionally, as seen in Fig. 2.13f, the ridge encoding the principal direction at each value of m does not have a decisive role, the two remaining ridges always have inference, making the estimated principal direction vulnerable to additive noise and resulting in a large value of $\varepsilon_\theta(m) \simeq 12^\circ$ when SNR $= 0.5$. However, when the noise is weak (SNR ≥ 2), high accuracy is still obtained (for example, $\varepsilon_\theta(m) < 2^\circ$ at $m = 30$).

The above observations lead to a conclusion that the estimated principal directions of patterns by the generic R-transform are very stable under noise perturbation, and by a simple extension, similar conclusion for the dominant directions of patterns could also be reached. They provide experimental evidence for the theoretical arguments at the beginning of this subsection.

2.3 The RMF descriptor

Besides the generic R-signature that has been presented in the previous section, this section presents another invariant pattern descriptor defined based on the Radon transform, called the RFM descriptor. The basic idea here is the use of the 1D Fourier–Mellin transform, which is a combination of the Fourier and Mellin transforms, to overcome the residual influences that remain in the radial slices of the Radon transform data. This section provides the formulation of the 1D Fourier–Mellin transform and then defines the RFM descriptor. A reasonably fast and accurate implementation of the Mellin transform is also given afterwards.

2.3.1 The Fourier transform

Let f be a 1D function. The Fourier transform [98] of f, denoted by $\mathcal{F}_f$, is a function defined for every real number ξ by

$$\mathcal{F}_f(\xi) = \int_{-\infty}^{\infty} f(x) \mathrm{e}^{-i2\pi\xi x} \, \mathrm{d}x. \tag{2.18}$$

It is well-known that the Fourier transform possesses a shift or translation invariance property. Consider a function g which is a version of f shifted by a distance x_0, then

$$\begin{aligned} g(x) &= f(x - x_0), \\ \mathcal{F}_g(\xi) &= \int_{-\infty}^{\infty} g(x) \mathrm{e}^{-i2\pi\xi x} \, \mathrm{d}x = \mathrm{e}^{-i2\pi\xi x_0} \mathcal{F}_f(\xi). \end{aligned}$$

Taking the magnitude of the two sides of the above equation results in

$$|\mathcal{F}_g(\xi)| = |\mathcal{F}_f(\xi)|,$$

or the magnitude of the Fourier transform of a 1D function is invariant to translation.

The discrete version of Eq. (2.18) defined for a sequence of numbers $\{f(n) : n = 0, \ldots, N-1\}$ has the following definition:

$$\mathcal{DF}_f(k) = \sum_{n=0}^{N-1} f(n) \, \mathrm{e}^{-\frac{2\pi i}{N} kn}, \qquad k = 0, \ldots, N-1.$$

Similar to the continuous counterpart, the discrete Fourier transform possesses a circular-shift invariance property: a circular shift in the input $\{f(n)\}$ corresponds to multiplying the output $\mathcal{DF}_f$ by a linear phase. Let $\{g(n)\}$ be a sequence obtained by circular-shifting $\{f(n)\}$ by a distance n_0, then

$$\begin{aligned} g(n) &= f(n - n_0), \\ \mathcal{DF}_g(k) &= \mathrm{e}^{-\frac{2\pi i}{N} kn_0} \, \mathcal{DF}_f(k). \end{aligned}$$

Taking the magnitude of the two sides of the above equation results in

$$|\mathcal{DF}_g(k)| = |\mathcal{DF}_f(k)|,$$

or the magnitude of the discrete Fourier transform of a 1D function is invariant to circular shift.

2.3.2 The Mellin transform

Let f be a 1D function. The Mellin transform [19] of f, denoted by $\mathcal{M}_f$, is a function defined by

$$\mathcal{M}_f(s) = \int_0^\infty f(x)x^{s-1}\,\mathrm{d}x, \tag{2.19}$$

where $s = \sigma + i\tau$. The real part σ of s is a constant chosen such that the integral in Eq. (2.19) converges. The imaginary part τ of s is the transform variable. Consider a function g which is a scaling of f by a factor α ($\alpha > 0$), then

$$\begin{aligned} g(x) &= f(\alpha x), \\ \mathcal{M}_g(s) &= \int_0^\infty g(x)x^{s-1}\,\mathrm{d}x = \alpha^{-s}\mathcal{M}_f(s). \end{aligned}$$

Taking the magnitude of the two sides of the above equation results in

$$|\mathcal{M}_g(s)| = \alpha^{-\sigma}|\mathcal{M}_f(s)|.$$

Thus, except for a constant multiplicative factor $\alpha^{-\sigma}$, the magnitude of the Mellin transform is scale invariant. The remaining scaling factor can be easily eliminated by normalization or it can be used to find the relative scale between two functions.

2.3.3 The 1D Fourier–Mellin transform

Combinations of the Fourier and Mellin transforms were proposed in the literature to have signal representations that do not vary with rotation/scaling or translation/scaling. For 2D signals, they are first converted from Cartesian coordinates into polar coordinates then Fourier and Mellin transforms are performed independently on the circular and radial slices of the converted signals respectively [37, 199]. In this way, the magnitude of the obtained representation is invariant to rotation and scaling. For 1D signals, Fourier and Mellin transforms are performed in sequence directly on the signals [3, 248] to have a signal representation whose magnitude is invariant to translation and scaling.

Consider the Fourier transform of a 1D function g which is a scaled and then shifted version of f by a factor α and a distance x_0 respectively, then

$$\begin{aligned} g(x) &= f(\alpha x - x_0), \\ \mathcal{F}_g(\xi) &= \int_{-\infty}^\infty f(\alpha x - x_0)\mathrm{e}^{-i2\pi\xi x}\,\mathrm{d}x. \end{aligned}$$

By denoting $y = \alpha x - x_0$:

$$\mathcal{F}_g(\xi) = \frac{1}{\alpha}\mathrm{e}^{-i2\pi\frac{\xi}{\alpha}x_0}\int_{-\infty}^\infty f(y)\mathrm{e}^{-i2\pi\frac{\xi}{\alpha}y}\,\mathrm{d}y = \frac{1}{\alpha}\mathrm{e}^{-i2\pi\frac{\xi}{\alpha}x_0}\,\mathcal{F}_f\left(\frac{\xi}{\alpha}\right).$$

Taking the magnitude of the two sides of the above equation results in

$$|\mathcal{F}_g(\xi)| = \frac{1}{\alpha}\left|\mathcal{F}_f\left(\frac{\xi}{\alpha}\right)\right|. \tag{2.20}$$

The translation parameter x_0 has disappeared and the remaining scaling parameter α could be eliminated by applying the Mellin transform on both sides of Eq. (2.20):

$$\mathcal{M}_{|\mathcal{F}_g|}(s) = \int_0^\infty \frac{1}{\alpha} \left| \mathcal{F}_f\left(\frac{\xi}{\alpha}\right) \right| \xi^{s-1}\,\mathrm{d}\xi = \alpha^{s-1}\mathcal{M}_{|\mathcal{F}_f|}(s),$$

or

$$\left|\mathcal{M}_{|\mathcal{F}_g|}(s)\right| = \alpha^{\sigma-1}\left|\mathcal{M}_{|\mathcal{F}_f|}(s)\right|.$$

Therefore, by defining

$$\mathcal{MF}_f(s) = \left|\mathcal{M}_{|\mathcal{F}_f|}(s)\right| = \left| \int_0^\infty \left| \int_{-\infty}^\infty f(x)\mathrm{e}^{-i2\pi\xi x}\,\mathrm{d}x \right| \xi^{s-1}\,\mathrm{d}\xi \right|, \tag{2.21}$$

as the combined 1D Fourier–Mellin transform of a function f, $\mathcal{MF}_f$, is invariant to translation and scaling, except for a constant multiplicative factor.

2.3.4 The proposed RFM descriptor

Attractive invariance properties of the Radon and 1D Fourier–Mellin transforms lead to the proposal of a novel region-based pattern descriptor, called the RFM descriptor. The proposed descriptor of an image I, RFM_I, is computed by

$$\mathrm{RFM}_I(k,s) = \left| \frac{\mathcal{DF}_{\mathcal{MF}_{\mathcal{R}_I}}(k,s)}{\mathcal{DF}_{\mathcal{MF}_{\mathcal{R}_I}}(0,s)} \right|.$$

Step 1: The Radon transform performed on the image I: $\mathcal{R}_I$.

Step 2: The 1D Fourier–Mellin transform performed on the radial slices of the obtained Radon transform data: $\mathcal{MF}_{\mathcal{R}_I}$.

Step 3: The magnitude of the discrete Fourier transform performed on the angular slices of the obtained, discretized Fourier–Mellin transform data normalized by the DC component.

Invariance properties of the proposed RFM descriptor described above are proven as follows.

Properties. *The proposed RFM descriptor is invariant to translation, rotation, and scaling.*

Proof. Let J be the image obtained by scaling, rotating, and translating an image I using transformation parameters α, θ_0, and $\vec{u} = (x_0, y_0)$ respectively. Properties $P4$–6 of the Radon transform imply

$$\mathcal{R}_J(\theta,\rho) = \frac{1}{\alpha}\mathcal{R}_I(\theta+\theta_0, \alpha\rho - d), \tag{2.22}$$

where $d = x_0\cos(\theta+\theta_0) + y_0\sin(\theta+\theta_0)$ is the shifting distance depending on θ. The above equation indicates that, except for a constant multiplicative factor $\frac{1}{\alpha}$, each constant-θ slice $\mathcal{R}_J(\theta,\cdot)$ of the Radon transform of J can be obtained by scaling and translating the constant-$(\theta+\theta_0)$ slice $\mathcal{R}_I(\theta+\theta_0,\cdot)$ of the Radon transform of I by a factor α and a distance d respectively.

Applying the 1D Fourier–Mellin transform on $\mathcal{R}_J(\theta,\cdot)$ and $\mathcal{R}_I(\theta+\theta_0,\cdot)$ and using the transform's invariance property, Eq. (2.22) becomes

$$\mathcal{MF}_{\mathcal{R}_J}(\theta,s) = \alpha^{\sigma-2}\mathcal{MF}_{\mathcal{R}_I}(\theta+\theta_0,s).$$

By varying the values of θ and s, the two transform data, $\mathcal{MF}_{\mathcal{R}_J}$ and $\mathcal{MF}_{\mathcal{R}_I}$, are obtained. Moreover, except for a constant multiplicative factor $\alpha^{\sigma-2}$, $\mathcal{MF}_{\mathcal{R}_J}$ can be directly obtained by circular-shifting $\mathcal{MF}_{\mathcal{R}_I}$ along the angular axis by a distance $-\theta_0$. Applying the discrete Fourier transform on the angular slices of the Fourier–Mellin transform data, $\mathcal{MF}_{\mathcal{R}_J}$ and $\mathcal{MF}_{\mathcal{R}_I}$, then ignoring the phase information in the coefficients leads to

$$\left|\mathcal{DF}_{\mathcal{MF}_{\mathcal{R}_J}}(k,s)\right| = \alpha^{\sigma-2}\left|\mathcal{DF}_{\mathcal{MF}_{\mathcal{R}_I}}(k,s)\right|. \tag{2.23}$$

This equation demonstrates that, except for a constant multiplicative factor $\alpha^{\sigma-2}$, the proposed descriptor computed on a scaled, rotated, and translated version J of an image I is exactly the same as the descriptor computed on the original image I. The remaining factor $\alpha^{\sigma-2}$, however, can be easily eliminated by a normalization step using the DC component as $\left|\frac{\mathcal{DF}_{\mathcal{MF}_{\mathcal{R}_I}}(k,s)}{\mathcal{DF}_{\mathcal{MF}_{\mathcal{R}_I}}(0,s)}\right|$ or it can be used to determine the relative scale between any two patterns of the same category. □

Calculating the proposed RFM descriptor as described above does not require any normalization regarding the size, position, or orientation of the input patterns, it only requires a normalization in the intensity of the computed descriptor. As a consequence, the proposed descriptor is totally invariant to RST transformations.

2.3.5 Mellin transform implementation

The Mellin transform defined in Eq. (2.19) has a very attractive property of scaling invariance and it can be implemented optically by using an optical scale invariant correlator [37]. However, there are reported problems with its implementation in today's digital systems [55]. Traditionally, the Mellin transform is implemented by changing variables $x = \mathrm{e}^y$:

$$\mathcal{M}_f(s) = \int_0^\infty f(x)x^{s-1}\,\mathrm{d}x = \int_0^\infty f(x)x^{\sigma-1}\mathrm{e}^{i\tau\ln x}\,\mathrm{d}x = \int_{-\infty}^\infty [f(\mathrm{e}^y)\mathrm{e}^{\sigma y}]\,\mathrm{e}^{i\tau y}\,\mathrm{d}y.$$

This is, by definition, the Fourier transform of the distorted function g with $g(y) = f(\mathrm{e}^y)$ weighted by $\mathrm{e}^{\sigma y}$. For sampled data, the discrete (fast) Mellin transform is implemented through FFT by re-sampling the data exponentially. Exponential sampling means interpolating the data to make them uniformly sampled in the y domain [50]. This process introduces errors into the transform and accentuates the low frequency components. Additionally, if f is sampled at the Nyquist rate and N is the number of data samples in the x domain then the number of the required data samples in the y domain will be $M = N\ln N$ [190]. Likewise, when f is nonzero at $x = 0$, g is nonzero at $y = -\infty$ and the implementation is not realizable. In this case, the Mellin transform can be approximated by using a correction term defined based on the value of $f(0)$ [190].

Another problem is on the application of the combined Fourier–Mellin transform in Subsection 2.3.3 for feature extraction, despite its invariance to scaling and translation. The problem is the obscurity of the discriminative information contained in the input function [109]. Possible reasons for this problem are the discard of the phase information from the output of the Fourier and Mellin transforms and the accentuation of low frequency components. Moreover, since FFT is applied on the radial slices of the Radon transform data, the DC component is always nonzero and it in turn is the value of the Mellin transform's input function f at $x = 0$.

To avoid these problems, an alternative implementation of the Mellin transform proposed [248], which is called the direct Mellin transform, is adopted for this work. Assuming that f is in

the form of sampled data with the sampling period T and the number of samples N, expanding Eq. (2.19) gives

$$\mathcal{M}_f(s) = \int_0^T f(x)x^{s-1}\,\mathrm{d}x + \int_T^{2T} f(x)x^{s-1}\,\mathrm{d}x + \cdots + \int_{(N-1)T}^{NT} f(x)x^{s-1}\,\mathrm{d}x. \tag{2.24}$$

The value of f is assumed to be piecewise constant in any interval T, then

$$s\mathcal{M}_f(s) = f(0)x^s\,|_0^1 + f(T)x^s\,|_1^2 + \cdots + f((N-1)T)x^s\,|_{N-1}^N.$$

Denoting $f(iT) = f_{i+1}$, $\Delta_k = f_k - f_{k-1}$ and, without loss of generality, assuming that $T = 1$ and $f_N = 0$, the above equation becomes

$$s\mathcal{M}_f(s) = f_1x^s\,|_0^1 + f_2x^s\,|_1^2 + \cdots + f_Nx^s\,|_{N-1}^N = \sum_{k=1}^{N-1} k^s\,(f_k - f_{k+1}) = \sum_{k=1}^{N-1} k^s\Delta_k, \tag{2.25}$$

or equivalently by using $s = \sigma + i\tau$:

$$(\sigma + i\tau)\mathcal{M}_f(\sigma + i\tau) = \sum_{k=1}^{N-1} k^{\sigma+i\tau}\Delta_k. \tag{2.26}$$

Because $k^{\sigma+i\tau} = \mathrm{e}^{(\sigma+i\tau)\ln k}$ is bounded for any fixed constant value of σ, the right hand side of Eq. (2.26) is bounded, meaning that $|\mathcal{M}_f(s)|$ converges to zero when $|s|$ increases. This indicates the low-pass filtering characteristic of the Mellin transform.

For a set of $s_i = \sigma + i\tau_i$ $(i = 1, \ldots, m)$ with τ_i are the arbitrary spectral components, and denoting $k^{s_i} = \phi_{i,k}$, Eq. (2.25) can be rewritten in matrix form as

$$\begin{bmatrix} s_1\mathcal{M}_f(s_1) \\ s_2\mathcal{M}_f(s_2) \\ \vdots \\ s_m\mathcal{M}_f(s_m) \end{bmatrix} = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,N-1} \\ \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{m,1} & \phi_{m,2} & \cdots & \phi_{m,N-1} \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_{N-1} \end{bmatrix}. \tag{2.27}$$

The direct Mellin transform, as defined in Eq. (2.24), is an exact implementation of the Mellin transform for sampled data and has the following properties:

- It requires neither exponential re-sampling nor a correction term.

- Only the differences in the values of adjacent data points are used for computing the transform.

- The coefficients $\phi_{i,k}$ in Eq. (2.27) can be computed off-line and stored. For each specific value of s_i, the number of stored coefficients for direct Mellin transform is N whereas the number of stored coefficients for fast Mellin transform is $N \ln N$.

- Computing $s_i\mathcal{M}_f(s_i)$ consists of only an inner product of two vectors, one of which has been pre-computed and stored and the other could be easily obtained from the input data by a simple subtraction. For this reason, Eq. (2.27) is very fast in implementation.

(a) $\mathcal{MF}_{\mathcal{R}_{I_1}}(\theta, s)$ (b) $\mathcal{MF}_{\mathcal{R}_{I_2}}(\theta, s)$ (c) $\mathcal{MF}_{\mathcal{R}_{I_3}}(\theta, s)$ (d) $\mathcal{MF}_{\mathcal{R}_{I_4}}(\theta, s)$ (e) $\mathcal{MF}_{\mathcal{R}_{I_5}}(\theta, s)$

Figure 2.16: Illustration of the invariance properties of the 1D Fourier–Mellin transform. These images are obtained by performing the 1D Fourier–Mellin transform on the Radon transform data in the second row of Fig. 2.2 using 150 values of τ ranging from 2.0 to 16.9 with increment of 0.1. The intensity of these images has been rescaled to fit the display range.

In order to eliminate the low-pass filtering characteristic of the direct Mellin transform, a modification is carried out by removing the $s = (\sigma + i\tau)$ factor in Eq. (2.26). The resulting transform is called the modified direct Mellin transform, denoted by $\mathcal{MM}$, as follows:

$$\mathcal{MM}_f(s) = s\mathcal{M}_f(s) = \sum_{k=1}^{N-1} k^s \Delta_k.$$

In the time domain, the modified direct Mellin transform is defined as

$$\mathcal{MM}_f(s) = s \int_0^\infty f(x) x^{s-1} \, \mathrm{d}x.$$

It is easy to verify that the modified direct Mellin transform maintains the scaling invariance property and the combined Fourier-modified direct Mellin transform, defined as

$$\mathcal{MF}_f(s) = \left|\mathcal{MM}_{|\mathcal{F}_f|}(s)\right| = \left| s \int_0^\infty |\mathcal{F}_f| \, x^{s-1} \, \mathrm{d}x \right|, \tag{2.28}$$

is invariant to both translation and scaling. Therefore, the 1D Fourier–Mellin transform used in the definition of the RFM descriptor in Subsection 2.3.4 is henceforth defined as in Eq. (2.28), instead of Eq. (2.21), and implemented through Eq. (2.27).

To qualitatively illustrate the invariance properties of $\mathcal{MF}_{\mathcal{R}_I}$, Fig. 2.16 provides the images obtained by performing the 1D Fourier–Mellin transform on the Radon transform data in the second row of Fig. 2.2 using 150 values of τ ranging from 2.0 to 16.9 with increment of 0.1. It should be noted here that, due to the periodicity and semi-symmetry properties of the Radon transform ($P2$–3), the effective range of θ used in the computation is $[0, \pi)$ (rad) or $[0, 180)$ (degree). The two patterns I_1, I_2 in Figs. 2.2a, 2.2b are not similar and, as a consequence, $\mathcal{MF}_{\mathcal{R}_{I_1}}$, $\mathcal{MF}_{\mathcal{R}_{I_2}}$ in Figs. 2.16a, 2.16b have different surfaces. The patterns I_3, I_4, and I_5 in Figs. 2.2c–2.2e are transformed versions of the pattern I_2 in Fig. 2.2b then $\mathcal{MF}_{\mathcal{R}_{I_3}}$, $\mathcal{MF}_{\mathcal{R}_{I_4}}$, and $\mathcal{MF}_{\mathcal{R}_{I_5}}$ in Figs. 2.16c–2.16e have surfaces that are similar to that of $\mathcal{MF}_{\mathcal{R}_{I_2}}(\theta, s)$ in Fig. 2.16b. The images in Figs. 2.16b–2.16e demonstrate clearly the scaling and translation invariance properties of the 1D Fourier–Mellin transform:

- be invariant to scaling and translation;
- converts a rotation in the pattern into a circular shift in the angular axis of the Fourier–Mellin transform data.

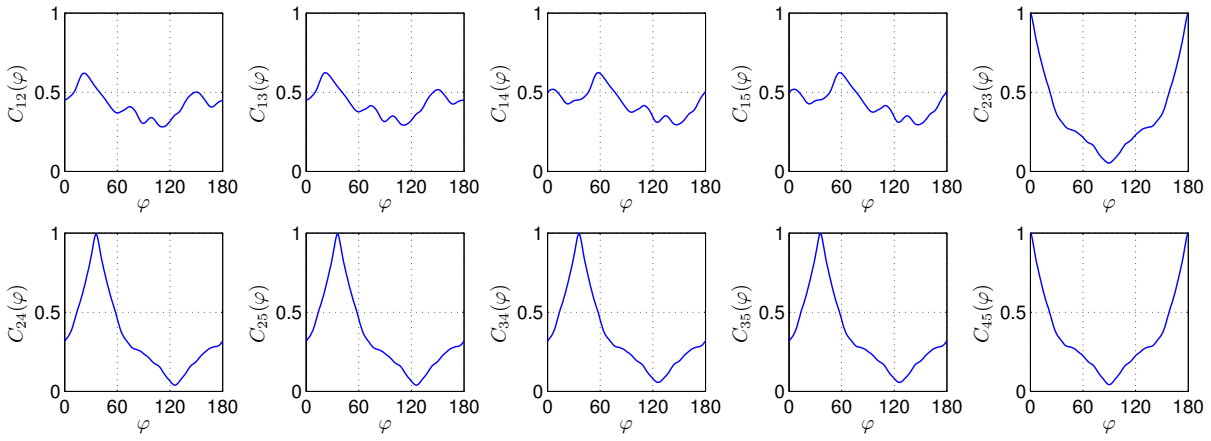


Figure 2.17: The normalized cross-correlation between all possible pairs of Fourier–Mellin transform data from Fig. 2.16. For each pair of patterns, 180 correlation values are calculated by circular-shifting one of the two Fourier–Mellin transform data along the angular axis by 180 possible values from 0 to 179 with increment of 1.

A quantitative evaluation of the invariance properties of the Fourier–Mellin transform is given in Fig. 2.17 using the normalized cross-correlation between all possible pairs of $\mathcal{MF}_{\mathcal{R}_{I_i}}$ and $\mathcal{MF}_{\mathcal{R}_{I_j}}$ with I_i and I_j $(i, j = 1, 2, \ldots, 5; i \neq j)$ are from the first row of Fig. 2.2. Normalized cross-correlation is selected for the sake of overcoming the constant multiplicative factor $\alpha^{\sigma-2}$ in Eq. (2.23) and the residual influence caused by rotation that remains in the angular axis of the Fourier–Mellin transform data. The correlation is calculated for all possible rotation angles, meaning that one of the two Fourier–Mellin transform data is circular-shifted along the angular axis by 180 possible values from 0 to 179 with increment of 1. Denoting φ as the shifting distance, the correlation between the Fourier-Mellin transforms $\mathcal{MF}_{\ mathcalR_{I_i}}$ and $\mathcal{MF}_{\mathcal{R}_{I_j}}$ of the two patterns I_i and I_j at φ is defined as

$$C_{ij}(\varphi) = \text{corr2}\left(\mathcal{MF}_{\mathcal{R}_{I_i}}, \mathcal{MF}^{\varphi}_{\mathcal{R}_{I_j}}\right), \tag{2.29}$$

where $\mathcal{MF}^{\varphi}_{\mathcal{R}_{I_j}}(\theta, s) = \mathcal{MF}_{\mathcal{R}_{I_j}}(\theta + \varphi, s)$ and $\text{corr2}(A, B)$ is the normalized cross-correlation between the two 2D input data A and B of size $m \times n$ calculated using the following formula:

$$\text{corr2}(A, B) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij} - \bar{A})(B_{ij} - \bar{B})}{\sqrt{\left(\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij} - \bar{A})^2\right)\left(\sum_{i=1}^{m}\sum_{j=1}^{n}(B_{ij} - \bar{B})^2\right)}},$$

where $\bar{A}$ and $\bar{B}$ are the mean values of A and B respectively.

The 10 curves C_{ij} that correspond to the 10 possible pairs of five Fourier–Mellin transform data have two different patterns. It is observed that C_{ij} is peaky only when the two patterns I_i and I_j are similar $(i, j = 2, 3, 4, 5)$. The maximum values of C_{ij} are 0.6200, 0.6236, 0.6239, 0.6239, 0.9962, 0.9943, 0.9943, 0.9970, 0.9970, 1.0000 respectively from left to right, top to bottom. The non-peaky and peaky maxima exhibit the discrimination power of the proposed RFM descriptor and the maximum of nearly 1 means that the Fourier–Mellin transform data is invariant to translation and scaling. The value of φ^* that corresponds to the peak in C_{ij} denotes the difference in orientation (in degree) between the corresponding two patterns I_i and I_j.

2.4 Experimental results

In order to demonstrate the effectiveness of the proposed generic R-signature and RFM descriptor, three experiments on grayscale and binary image datasets were carried out. The robustness of the proposed descriptors to additive white noise is first demonstrated by using two sets of datasets generated by adding white noise of different levels to images of 26 Latin characters and to images from the COIL-20 dataset [161]. Secondly, the proposed descriptors are computed on a set of datasets generated by adding "salt & pepper" noise of different levels to images from the UMD Logo dataset [58]. The aim of the second experiment is to demonstrate the robustness to "salt & pepper" noise of the proposed descriptors. Finally, the proposed descriptors are computed on the reference Shapes216 dataset [198] to evaluate their robustness to shape's occlusion and deformation. Thus, the first experiment deals with grayscale patterns and the last two ones with binary patterns.

The proposed descriptors are compared with angular radial transform (ART) [22], generic Fourier descriptor (GFD) [243], Zernike moments [116], and Radon 2D Fourier–Mellin transform (R2DFM) [227]. All comparison descriptors need normalizations in order to be invariant to RST transformations and, additionally, the R2DFM descriptor is also defined based on the Radon transform. These descriptors are selected because they are commonly used and have good reported performance. The two issues that relate to the comparison, similarity measure and evaluation criterion, are addressed as follows.

Similarity measure

For any two patterns f and g represented invariantly by $\mathcal{I}(f)$ and $\mathcal{I}(g)$ respectively, where $\mathcal{I}$ is an invariant operator taking either the generic R-signature or the RFM descriptor as its definition. The measure of similarity between f and g is defined as the ℓ_2-norm distance between their descriptors as

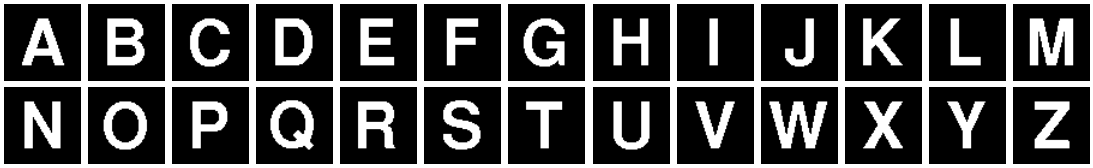
$$\mathrm{dist}(f,g) = \|\mathcal{I}(f) - \mathcal{I}(g)\|_2. \tag{2.30}$$

Providing the availability of $\mathcal{I}(f)$ and $\mathcal{I}(g)$, the computation of $\mathrm{dist}(f,g)$ is simple and fast, permitting the generic R-signature and RFM descriptor to be used in pattern matching problems with large-sized datasets. More sophisticated distances like the weighted Euclidean distance [116] could be used to reduce the dominance of some of the coefficients in the generic R-signature and RFM descriptor. However, since small-valued coefficients usually correspond to high-frequency components, meaning that they are more sensitive to additive noise and sampling/quantization effect, balancing the coefficient contributions thus reduces the performance of these descriptors in noisy environment. The performance degradation that results from coefficient weighting was observed from some preliminary experiments. Moreover, due to the orthogonality in the basis of the discrete Fourier transform, there is no correlation among the coefficients of the generic R-signature and RFM descriptor and thus the Mahalanobis distance [141], if employed, reduces to the weighted Euclidean distance.

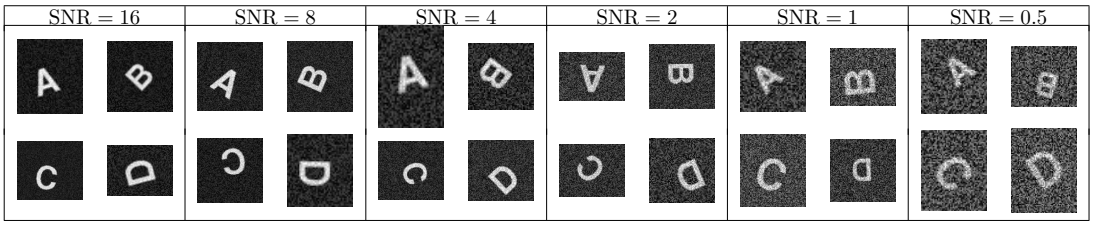
Evaluation criterion

The criterion used for comparison among descriptors is the precision–recall curve defined in information retrieval context [8]. Denoting for a given query model,

- *retrieved images* as all images that are returned by a matching process and
- *relevant images* as all images in the dataset that are in the same category with the query,



(a) Noise-free images



(b) Samples of noisy images

Figure 2.18: (a) Images of 26 Latin characters of size 64×64 in Arial bold font used to generate the six alphabet datasets. (b) Sample images from the six alphabet datasets generated from the first four character images with six possible values of SNR $= \{16, 8, 4, 2, 1, 0.5\}$.

then, precision is defined as the fraction of retrieved images that are relevant to the search:

$$\text{Precision} = \frac{|\{\text{relevant images}\} \cap \{\text{retrieved images}\}|}{|\{\text{retrieved images}\}|},$$

and recall is defined as the percent of all relevant images that is returned by the search:

$$\text{Recall} = \frac{|\{\text{relevant images}\} \cap \{\text{retrieved images}\}|}{|\{\text{relevant images}\}|}.$$

In computing the precision–recall curve for each dataset in the experiment, each of the images in the dataset is used as a query model to which all the images in the dataset are compared/matched with. The matching is realized by using the similarity measure defined in Eq. (2.30). The obtained matching results are then sorted, or ranked, for the determination of the nth nearest matches for each query model.

2.4.1 Grayscale pattern recognition

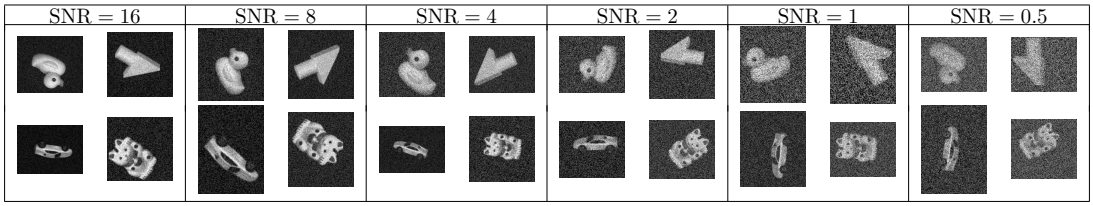
The performance of the proposed R-signature and RFM descriptor is first evaluated on grayscale noisy images to demonstrate their robustness to additive white noise. Two experiments were carried out on two different sets of datasets:

- *ExpA*: The first set of six alphabet datasets was generated from images of 26 Latin characters shown in Fig. 2.18a. Each of these six datasets has 260 images of 26 categories, each category contains 10 images.

- *ExpB*: The second set of six object datasets was generated from 20 object images from the COIL-20 dataset [161] shown in Fig. 2.19a. Each of these six datasets has 220 images of 20 categories, each category contains 11 images.

The main characteristic that differentiates *ExpA* and *ExpB*, besides the semantic content of their images, is the number of intensity levels in the original noise-free images: character images

(a) Noise-free images



(b) Samples of noisy images

Figure 2.19: (a) Twenty object images from the COIL-20 dataset used to generate the six object datasets. (b) Sample images from the six object datasets generated from the four object images with six possible values of SNR = $\{16, 8, 4, 2, 1, 0.5\}$.

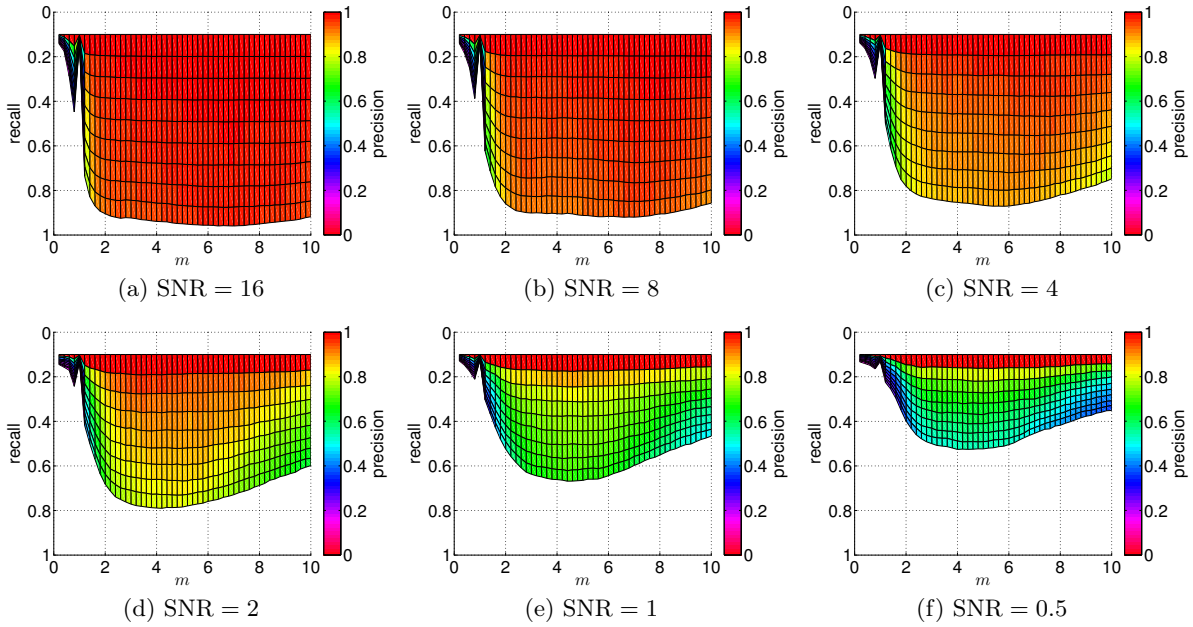
have only two-level intensity whereas object images have multi-level intensity. Noisy grayscale images were generated from the corresponding noise-free images by randomly scaling, rotating, translating, and then adding white noise to them. The value of SNR for each dataset is kept constant and, in each experiment, SNR has six possible values, $\{0.5, 1, 2, 4, 8, 16\}$, that correspond to the six datasets. Some sample images from the six datasets in *ExpA* and *ExpB* are given in Figs. 2.18b and 2.19b respectively.

Figs. 2.20/2.21 provide the precision–recall curves obtained by using the generic R-signature computed on the six character/object datasets of *ExpA*/*ExpB*. In each sub-figure and at a specific value of m in the horizontal axis, there is a precision–recall curve with recall and precision rates illustrated as the ordinate and the color of the grid points having abscissa m. It is observed that the performance of the generic R-signature varies according to m. As m increases from 0.2 to 10 and except for the singularity at $m = 1$, the precision–recall curve, when plotted in the traditional 2D Cartesian coordinate system with recall and precision rates as the abscissa and ordinate respectively, goes upwards till a certain value of m and then downwards, meaning an increase and then a decrease in performance of the generic R-signature. In general, the peak in performance is obtained at $m \simeq 5$ and $m \simeq 3.2$ for *ExpA* and *ExpB* respectively, leading to a conclusion that the selected value of m to have the best performance is robust to the level of noise. As m increases,

- The increase in performance when m is "small" agrees with the increase in the discrimination power of the generic R-signature, which results from exploiting the variation in the accumulations of patterns along all parallel lines, as discussed in Subsection 2.2.2;

- The decrease in performance when m is "large" agrees with the discussion on the sensitivity of the generic R-signature to additive noise in Subsection 2.2.5.

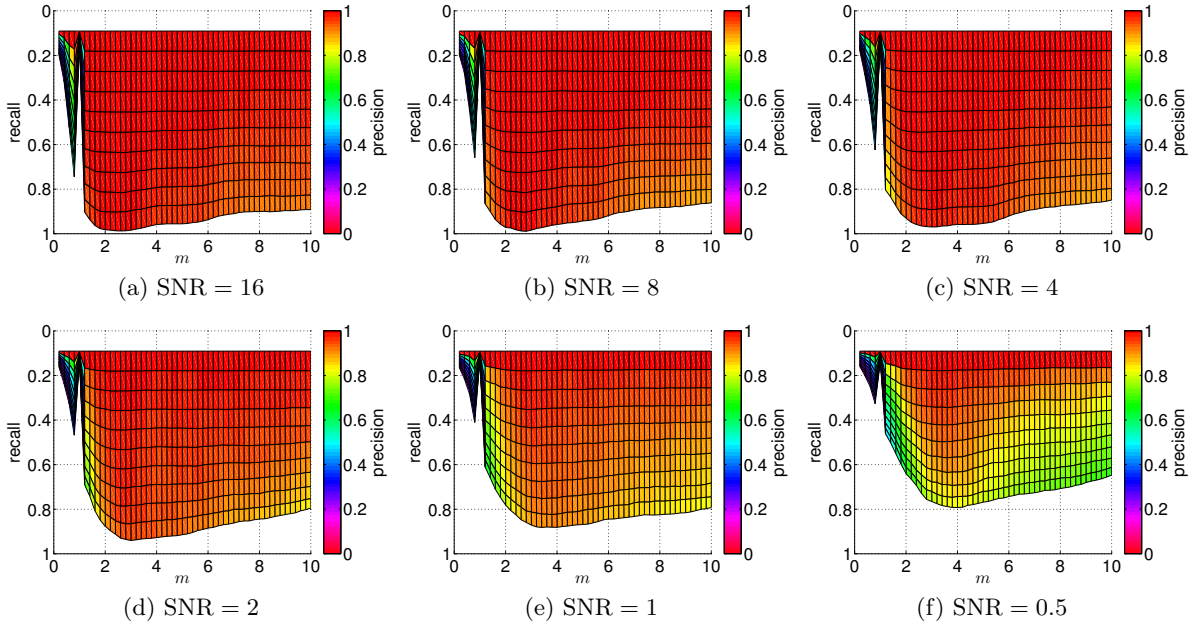
In addition, as SNR increases, the performance of the generic R-signature generally deteriorates at each value of m, agreeing with the dependance of the generic R-signature on noise level presented in Subsection 2.2.4. However, the deterioration speed is slower at $m \simeq 5$ and $m \simeq 3.2$



(a) SNR = 16 (b) SNR = 8 (c) SNR = 4

(d) SNR = 2 (e) SNR = 1 (f) SNR = 0.5

Figure 2.20: Precision–recall curves of the generic R-signature on the six alphabet datasets at different values of m. In each sub-figure and at a specific value of m in the horizontal axis, there is a precision–recall curve with recall and precision rates illustrated as the ordinate and the color of the grid points having abscissa m.



(a) SNR = 16 (b) SNR = 8 (c) SNR = 4

(d) SNR = 2 (e) SNR = 1 (f) SNR = 0.5

Figure 2.21: Precision–recall curves of the generic R-signature on the six object datasets at different values of m. In each sub-figure and at a specific value of m in the horizontal axis, there is a precision–recall curve with recall and precision rates illustrated as the ordinate and the color of the grid points having abscissa m.

for *ExpA* and *ExpB*, meaning that the generic R-signature is robust to additive white noise. This experimental evidence agrees with the theoretical arguments on the robustness of the Radon transform to additive white noise presented in Subsection 2.1.3.

Due to the existence of a class of R-signatures, their combination was also investigated to see if it leads to any possible increase in performance. For simplicity and for the reasons that will be clear later, two R-signatures of exponents m_1 and m_2 have been combined as

$$FR_{Im_1m_2} = \left[FR_{Im_1}, FR_{Im_2}\right]$$

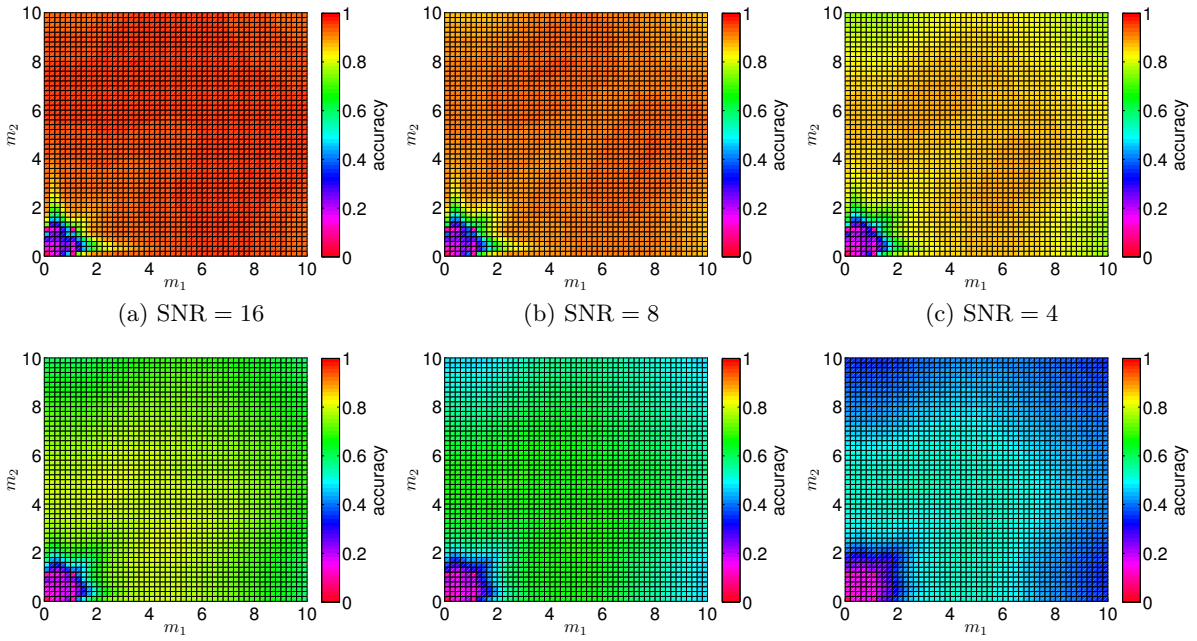
to be used as the invariant descriptor for the pattern I. Figs. 2.22/2.23 provide the accuracy obtained by using the combined R-signature, $FR_{Im_1m_2}$, on the six character/object datasets of *ExpA*/*ExpB*. In each sub-figure and at specific values of m_1 and m_2 in the horizontal and vertical axes, the accuracy is illustrated as the color of the grid point (m_1, m_2). It can be seen that the color patterns of these sub-figures are symmetric with respect to the minor diagonal and a change in (m_1, m_2) generally leads to a change in the color, meaning that the performance of $FR_{Im_1m_2}$ varies according to (m_1, m_2). Since m_1 and m_2 are interchangeable and should be different to avoid duplicates, it is required that $m_1 < m_2$. The peak in performance is then obtained at $(m_1, m_2) \simeq (2.6, 5.2)$ and $(m_1, m_2) \simeq (2.4, 3.8)$ for *ExpA* and *ExpB* respectively. Note from these values of m_1 and m_2 that one is smaller and the other is larger than $m \simeq 5$ and $m \simeq 3.2$ for *ExpA* and *ExpB*. These relations among the selected values of exponents have the following possible explanations:

- m_1 and m_2 should be separated enough to make use of the difference in the discriminative information contained in FR_{Im_1} and FR_{Im_2}.
- m_1 and m_2 should be close to m so that FR_{Im_1} and FR_{Im_2} individually a has high discrimination power, similar to that of FR_{Im}.

Comparison of the proposed generic R-signature and RFM descriptor with ART, GFD, Zernike, and R2DFM descriptors on these noisy datasets was performed and the obtained results are given in Figs. 2.24 and 2.25 respectively. In this comparison, besides the conventional value at $m = 2$, the value of the exponent m was selected to reflect the relatively best performance of the generic R-signature on *ExpA* ($m = 5$) and *ExpB* ($m = 3.2$). It is observed from these sets of sub-figures that:

- ART, GFD, Zernike, and R2DFM descriptors are not robust to additive white noise at all, their performance is similarly poor at different levels of noise.
- There is a substantial increase in the performance of the generic R-signature from that of the conventional R-signature ($m = 2$) when an appropriate value of the exponent m is used.
- As SNR decreases (i.e., the images get noisier), the precision–recall curves of the generic R-signature and RFM descriptor generally move downwards. Their comparable and good performances are nearly perfect when the noise is weak (SNR $= 16, 8, 4$), demonstrating their robustness to additive white noise.
- The combined R-signature does perform better than the single one. However, the increase in performance is very small and negligible.

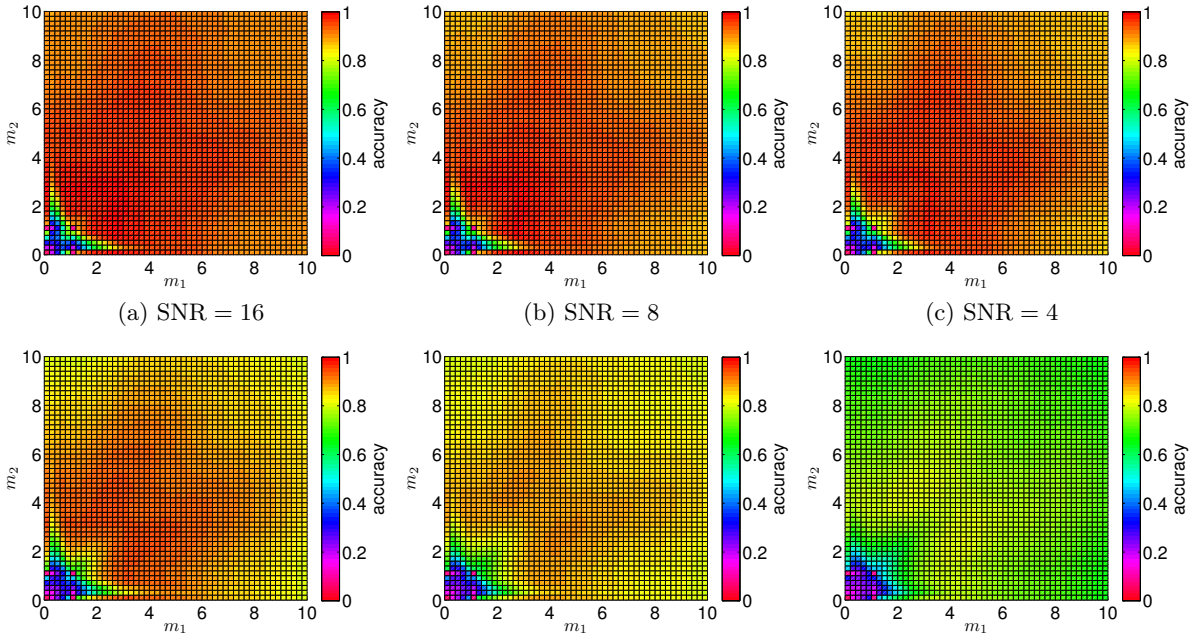
It thus can be concluded that the proposed generic R-signature and RFM descriptor have comparable performances and are more robust to additive white noise than the comparison ART,



(a) SNR = 16 (b) SNR = 8 (c) SNR = 4

(d) SNR = 2 (e) SNR = 1 (f) SNR = 0.5

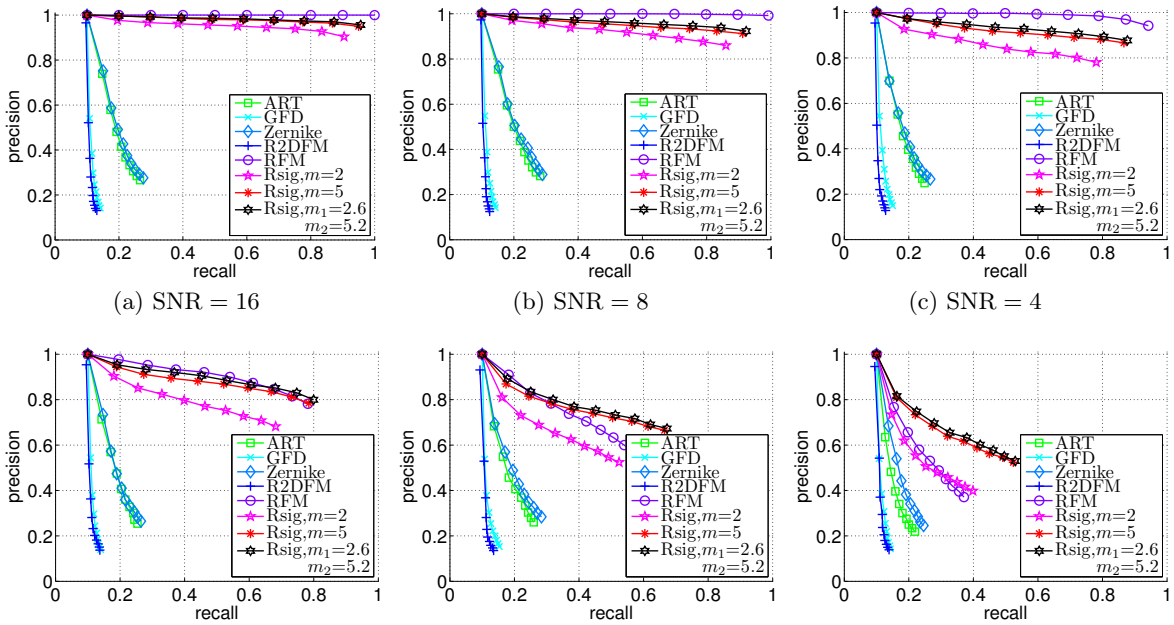
Figure 2.22: The accuracy of the generic R-signature on the six alphabet datasets at different values of (m_1, m_2). In each sub-figure and at specific values of (m_1, m_2), the accuracy is denoted as the color of the grid point having abscissa m_1 and ordinate m_2.



(a) SNR = 16 (b) SNR = 8 (c) SNR = 4

(d) SNR = 2 (e) SNR = 1 (f) SNR = 0.5

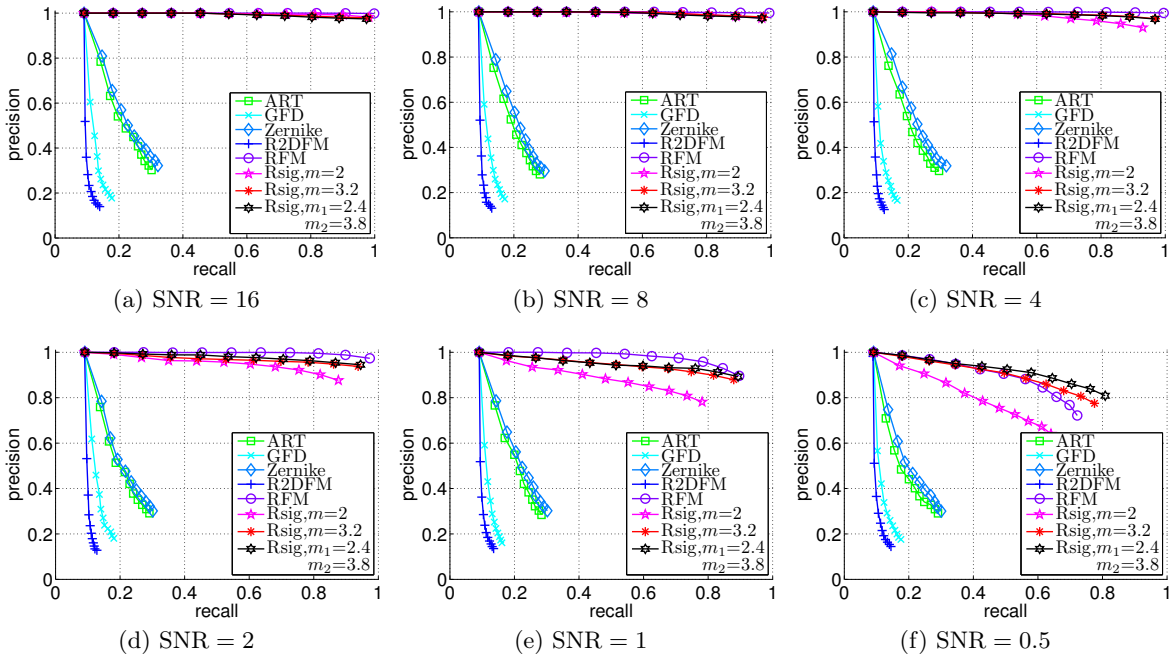
Figure 2.23: The accuracy of the generic R-signature on the six object datasets at different values of (m_1, m_2). In each sub-figure and at specific values of (m_1, m_2), the accuracy is denoted as the color of the grid point having abscissa m_1 and ordinate m_2.



(a) SNR = 16 (b) SNR = 8 (c) SNR = 4

(d) SNR = 2 (e) SNR = 1 (f) SNR = 0.5

Figure 2.24: Precision–recall curves of comparison descriptors on the six alphabet datasets. ART, GFD, Zernike, and R2DFM descriptors are not robust to additive white noise, their curves are similarly poor at different values of SNR. The generic R-signature and RFM descriptor are robust to additive white noise, their curves generally move downwards as SNR decreases.



(a) SNR = 16 (b) SNR = 8 (c) SNR = 4

(d) SNR = 2 (e) SNR = 1 (f) SNR = 0.5

Figure 2.25: Precision–recall curves of comparison descriptors on the six object datasets. ART, GFD, Zernike, and R2DFM descriptors are not robust to additive white noise, their curves are similarly poor at different values of SNR. The generic R-signature and RFM descriptor are robust to additive white noise, their curves generally move downwards as SNR decreases.

GFD, Zernike, and R2DFM descriptors on grayscale noisy datasets. This provides empirical evidence for the analytical results developed in Subsections 2.1.3 and 2.2.5. The disappointing increase in performance that results from combination of R-signatures can be explained by their "similar nature" in describing patterns. The only difference among R-signatures is the difference in the value of the exponent m, meaning a difference in the exploitation of the variation in the accumulations of patterns along all parallel lines (Subsection 2.2.2). This conclusion can also be generalized that even the combined R-signature is composed of more R-signatures, the performance increase is still not noteworthy. Moreover, poor performance of ART, GFD, Zernike, and R2DFM descriptors has its root in the required normalizations in their computation and can be explained as

- To have invariance to translation, the origin of the polar coordinate system needs to be located at the centroid of the pattern. In the presence of noise, the position of the centroid is shifted arbitrarily according to the actual noise.

- To have invariance to scaling, the radial coordinates of all pattern's points are normalized by the distance from the origin of the polar coordinate system to the farthest point. In the presence of noise, this farthest point might not belong to the actual pattern but to the noise.

Furthermore, it is also evident from the two sets of experiments that the performance of the generic R-signature and RFM descriptor is better on *ExpB* than on *ExpA* at each noise level, leading to a conclusion that the proposed descriptors perform better on multi-level than on two-level grayscale pattern images. Possible explanation for this comes from the Radon transform data: multi-level pattern images tend to have more variation in their Radon transform than two-level ones. In addition, recall from Subsection 2.2.2 that the role of m is to exploit the variation in the accumulations of patterns along all parallel lines with more a variation usually leads to a higher discrimination power. Thus at the same value of m, the generic R-signature of multi-level pattern images contains more discrimination power than that of two-level ones. Similar explanation can also be used for the RFM descriptor.

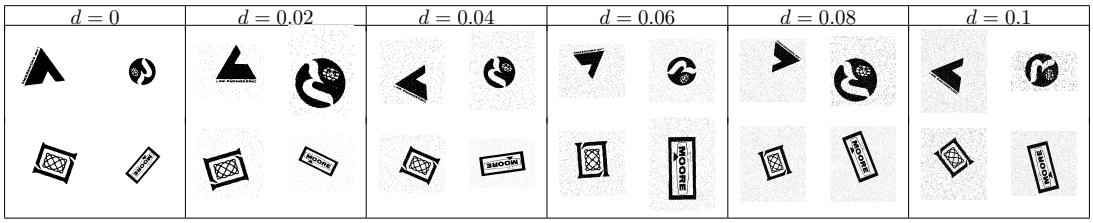
2.4.2 Binary pattern recognition

Noisy datasets

The robustness of the proposed generic R-signature and RFM descriptor to additive "salt & pepper" noise is demonstrated by using a set of six logo datasets generated from the first 25 logo images of the UMD Logo dataset [58] shown in Fig. 2.26a. Each of these six logo datasets has 275 images of 25 categories, each category contains 11 images generated by randomly scaling, rotating, translating, and then adding "salt & pepper" noise to the corresponding noise-free logo images. Let d be the percentage of pixels flipped by the noise, the value of d for each generated dataset is kept constant and has one of the six possible values, ranging from 0 to 0.1 with increment of 0.02, that correspond to the six datasets. The first dataset with $d = 0$ is actually a noise-free dataset; its use is intended for checking the invariance properties of the proposed and comparison descriptors. The values of d of the other five noisy datasets make up an arithmetic progression with a common difference of 0.02. These five datasets are, therefore, used to evaluate the robustness of the proposed and comparison descriptors at incrementing levels of additive "salt & pepper" noise. Some sample images from the six datasets are given in Fig. 2.26b.

Fig. 2.27 provides the precision–recall curves obtained by using the generic R-signature on the six logo datasets. The evolution of these curves according to m has a similar trend with

(a) Noise-free images



(b) Samples of noisy images

Figure 2.26: (a) Twenty-five logo images from the UMD Logo dataset used to generate the six logo datasets. (b) Sample images from the six logo datasets generated from the first four logo images with six possible values of $d = \{0, 0.02, 0.04, 0.06, 0.08, 0.1\}$.

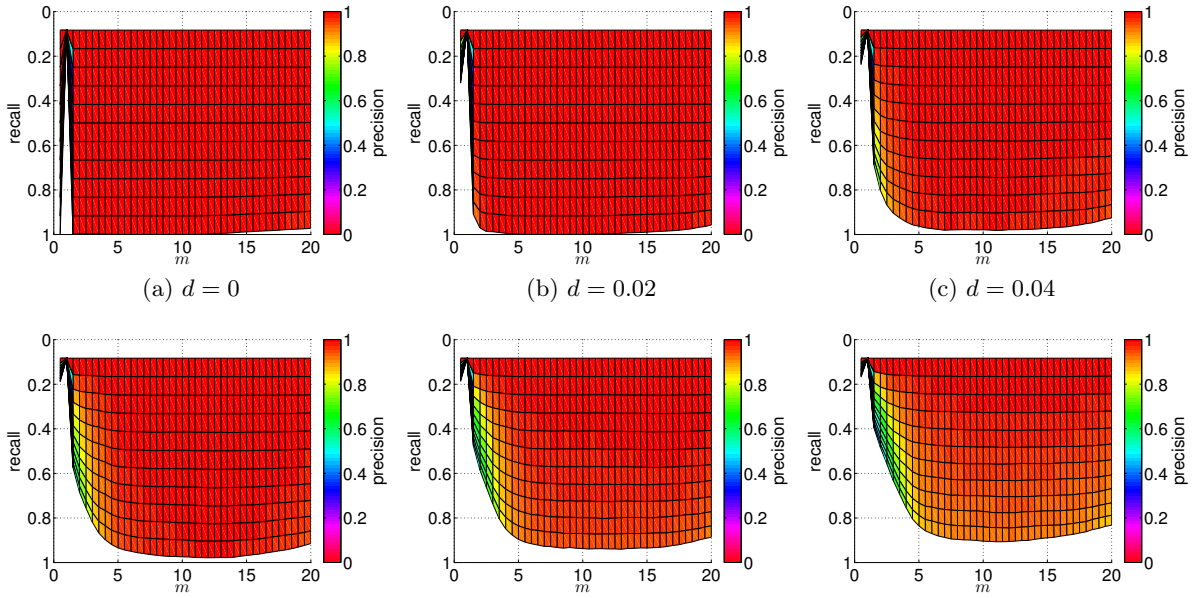
that on the six alphabet/object datasets in Figs. 2.20/2.21. That is, except for the singularity at $m = 1$, an increase then a decrease in performance are observed as m increases, agreeing with the discussions in Subsections 2.2.2 and 2.2.5 respectively. The peak in performance is obtained at $m \simeq 11$. In addition, as d increases, the performance of the generic R-signature generally deteriorates at each value of m, agreeing with the dependance of the generic R-signature on noise level presented in Subsection 2.2.4. However, the deterioration speed is slow at $m \simeq 11$, meaning that the generic R-signature is robust to additive "salt & pepper" noise. This experimental evidence agrees with the theoretical arguments on the robustness of the Radon transform to additive "salt & pepper" noise presented in Subsection 2.1.3

The accuracy obtained by using the combined R-signature, $FR_{Im_1m_2}$, on the six logo datasets is given in Fig. 2.28. Similar to the accuracy plots in Figs. 2.22 and 2.23, it can be seen that the color patterns of these sub-figures are symmetric with respect to the minor diagonal; and a change in (m_1, m_2) generally leads to a change in the color, meaning that the performance of $FR_{Im_1m_2}$ varies according to (m_1, m_2). The peak in performance is obtained at $(m_1, m_2) \simeq (5.5, 11.5)$. It should be noted again from these values of m_1 and m_2 that one is smaller and the other is larger than $m \simeq 11$.

The proposed generic R-signature/RFM descriptor are again compared with ART, GFD, Zernike, and R2DFM descriptors using these six datasets and the computed precision–recall curves of these descriptors are depicted in Fig. 2.29. In this comparison, the value of m is fixed at 11. For the noise-free dataset with $d = 0$ (Fig. 2.29a), all descriptors have ideal performance, demonstrating the total invariance to RST transformations of the proposed descriptors. When $d \neq 0$ (Fig. 2.29b–2.29f), deterioration appears in the performance of all descriptors and their precision–recall curves move downwards. However, the impact of d on precision-recall curves differs from one descriptor to another. Among all the descriptors, the proposed descriptors have the best performance for all the five noisy datasets while ART and Zernike descriptors have similarly worse performance. It is also observed that:

- As d increases, the curves of all descriptors generally move downwards.

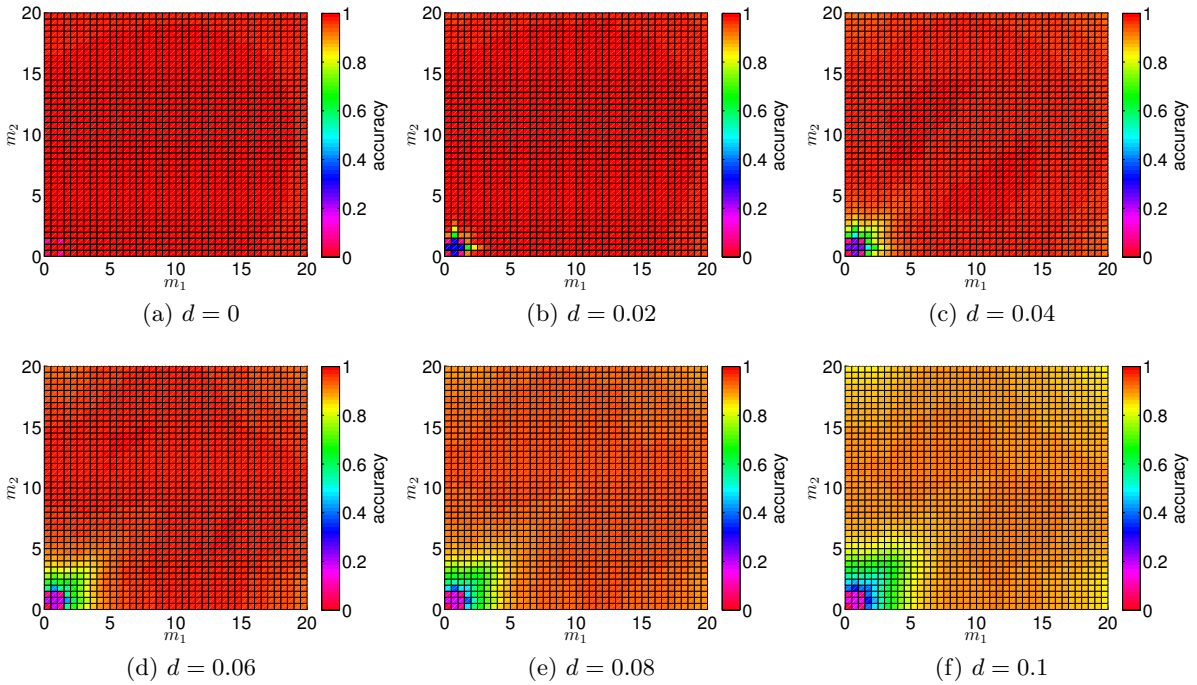
- ART and Zernike descriptors are not robust to "salt & pepper" noise at all, their performance



(a) $d = 0$ (b) $d = 0.02$ (c) $d = 0.04$

(d) $d = 0.06$ (e) $d = 0.08$ (f) $d = 0.1$

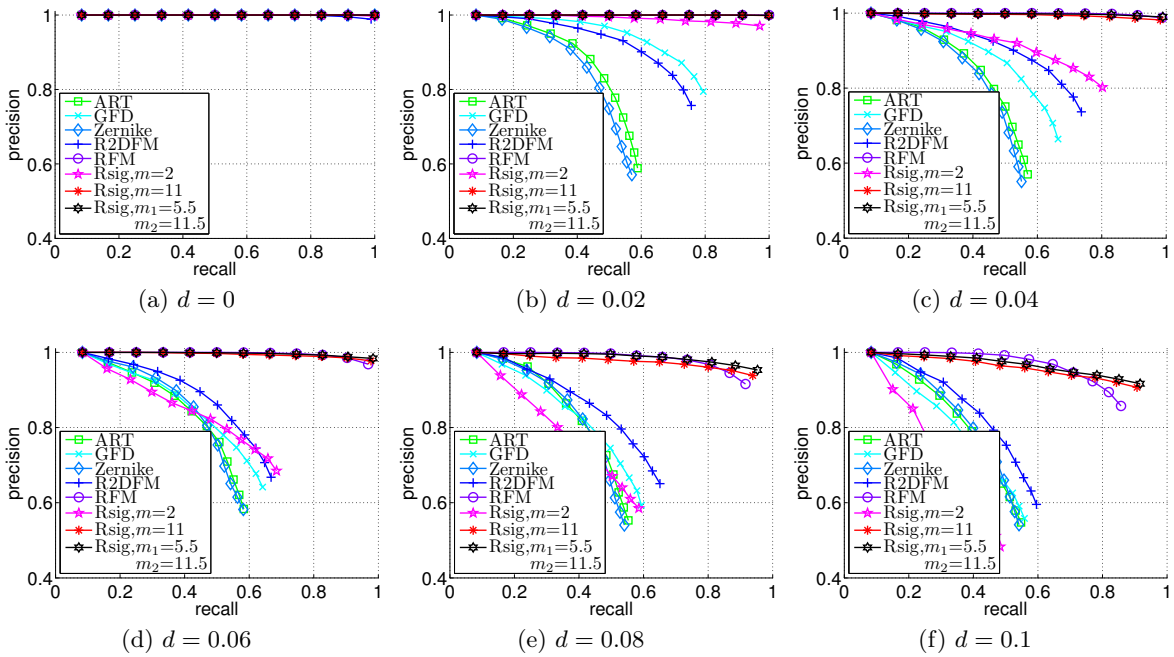
Figure 2.27: Precision–recall curves of the generic R-signature on the six logo datasets at different values of m. In each sub-figure and at a specific value of m in the horizontal axis, there is a precision–recall curve with recall and precision rates illustrated as the ordinate and the color of the grid points having abscissa m.



(a) $d = 0$ (b) $d = 0.02$ (c) $d = 0.04$

(d) $d = 0.06$ (e) $d = 0.08$ (f) $d = 0.1$

Figure 2.28: The accuracy of the generic R-signature on the six logo datasets at different values of (m_1, m_2). In each sub-figure and at specific values of (m_1, m_2), the accuracy is denoted as the color of the grid point having abscissa m_1 and ordinate m_2.



(a) $d = 0$ (b) $d = 0.02$ (c) $d = 0.04$

(d) $d = 0.06$ (e) $d = 0.08$ (f) $d = 0.1$

Figure 2.29: Precision–recall curves of comparison descriptors on the six logo datasets. For the noise-free dataset $d = 0$ (a), all pattern descriptors have ideal performance. When $d \neq 0$ (b)–(f), deterioration appears in the performance of all descriptors and their curves move downwards. However, the impact of d on those curves differs from one descriptor to another.

is similarly poor for different levels of noise.

- GFD has more resistance to "salt & pepper" noise than ART and Zernike because its curves are pushed away from the ideal curves (when $d = 0$) a distance which increases along with the increase in d. However, the resistance of GFD is weaker than that of descriptors defined based on the Radon transform.

- Among the three Radon transform-based descriptors, the shifts in the curves of the generic R-signature and RFM descriptor are comparable and are more regular than that of R2DFM.

- There is a substantial increase in the performance of the generic R-signature from that of the conventional R-signature ($m = 2$) when an appropriate value of the exponent m is used.

- The increase in performance obtained by combining R-signatures is small and negligible.

The above observations lead to a conclusion that the proposed generic R-signature and RFM descriptor have comparable performances and are more robust to additive "salt & pepper" noise than the comparison ART, GFD, Zernike, and R2DFM descriptors on binary noisy datasets. This provides empirical evidence for the analytical results developed in Subsections 2.1.3 and 2.2.5. Explanations for a small increase in performance due to combination of R-signatures and the poor performance of ART, GFD, Zernike, and R2DFM descriptors on binary noisy datasets are similar to those given in the previous subsection on grayscale noisy datasets.

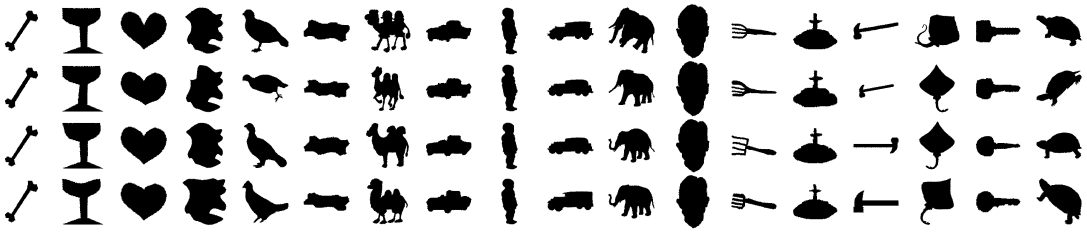


Figure 2.30: Sample shape images from the Shapes216 dataset. There are 18 shape categories and each category contains 12 shapes (shown in the figure are four shapes for each category), each of these shapes cannot be obtained by RST transforming any other shape from the same category.

Occlusion and deformation dataset

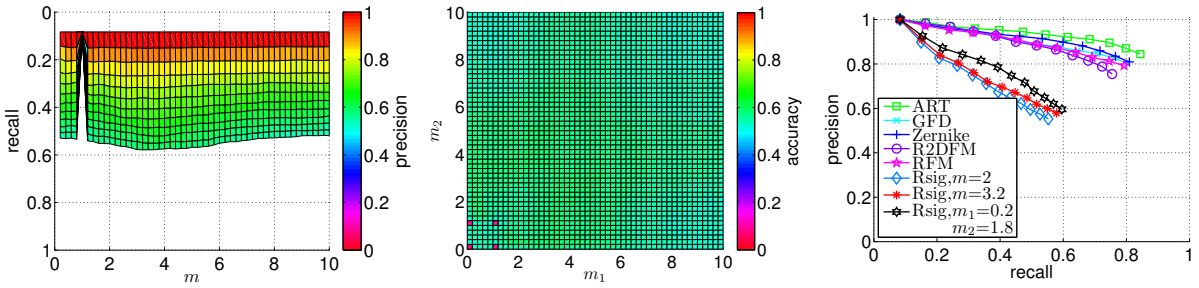
The performance of the proposed generic R-signature and RFM descriptor on occlusion and deformation shapes was evaluated using the reference Shapes216 dataset [198]. This dataset is composed of 18 shape categories with 12 samples per category, each of these shapes cannot be obtained by RST transforming any other shape from the same category. Some sample shapes from the Shapes216 dataset are shown Fig. 2.30.

Fig. 2.31a provides the precision–recall curves obtained by using the generic R-signature on the Shapes216 dataset. The evolution of these curves according to m has a similar trend with that on the six alphabet, object, and logo datasets given in Figs. 2.20, 2.21, and 2.27 respectively. That is, except for the singularity at $m = 1$, an increase then a decrease in performance are observed as m increases, agreeing with the discussions in Subsections 2.2.2 and 2.2.5 respectively. However, even though the performance peak at $m \simeq 3.2$ is noticeable, the variation in the performance of the generic R-signature due to m is small. This phenomenon does not exist on noisy datasets in previous experiments. It is thus can be concluded that the generalization of the R-signature has a little impact on occlusion and deformation shapes. Similarly, the accuracy obtained by using the combined R-signature, $FR_{Im_1m_2}$, on the Shapes216 dataset in Fig. 2.31b has almost a constant color, meaning similar performances of the combined R-signatures at different values of (m_1, m_2).

Comparison of the proposed generic R-signature and RFM descriptor with commonly used descriptors in this direction was carried out and the obtained results are given in Fig. 2.31c. In this comparison, $m = 3.2$ and $(m_1, m_2) = (0.2, 1.8)$. It can be seen that the performance of the RFM descriptor outperforms that of the generic R-signature and is comparable to the performances of ART, GFD, Zernike, and R2DFM descriptors. Moreover, it is clear that the categorizations by the proposed descriptors are not as good as that given in [198] where the precision of each nearest match for all categories are reported as (100, 100, 100, 100, 99, 97, 99, 96, 96, 95, 91, 80). However, it should be noted here that the proposed descriptors are not intended nor designed to work solely with binary patterns [198]. They are designed, instead, to work also with grayscale patterns under RST transformations allowing a certain level of additive noise, which methods such as the one in [198] cannot work with.

2.5 Conclusions

In this chapter, the Radon transform has been used to represent patterns invariantly by employing its beneficial properties concerning RST transformations. By applying the Radon transform on an RST-transformed pattern, the transformation parameters are encoded in the radial (for translation and scaling) and angular (for rotation) slices of the obtained Radon transform data. The residual



(a) Generic R-signature (b) Generic R-signature (c) Performance comparison

Figure 2.31: Experimental results on the Shapes216 dataset: (a) Precision–recall curves of the generic R-signature at different values of m. (b) The accuracy of the generic R-signature at different values of (m_1, m_2). (c) Precision–recall curves of comparison descriptors.

influences of RST transformations on the Radon transform of a pattern could be overcome by using appropriate transforms, of which different choices lead to different descriptors. This chapter has unified the view on possible directions that can be followed to define invariant pattern descriptors based on the Radon transform. It further proposed two novel pattern descriptors that are totally invariant to RST transformations:

- *The generic R-signature* is obtained by using an integration and then an exponentiation on the radial slices, followed by the discrete Fourier transform on the angular slices of the Radon transform data. This definition brings in a class of descriptors that has the beneficial properties of the conventional R-signature while spatially describing patterns at all directions and at different levels. This generalization gives more flexibility in definition and, more importantly, the generic R-signature has been proven to be robust to additive noise. It has been demonstrated that the generic R-signature is superior to existing invariant pattern descriptors in terms of retrieval rate on grayscale and binary noisy datasets.

- *The RFM descriptor* is obtained by applying the 1D Fourier–Mellin and discrete Fourier transforms on the radial and angular slices of the Radon transform data respectively. It has been proven to be invariant to rotation, scaling, and translation, without the need of any normalization step. The computation of the RFM descriptor is reasonably fast and correct, based mainly on the fusion of the Radon and Fourier transforms and on a modification of the Mellin transform. It has been shown to be robust to additive noise both theoretically and experimentally. It has also been demonstrated that the RFM descriptor is superior to existing invariant pattern descriptors in terms of retrieval rate on grayscale and binary noisy datasets.

Additionally, the Radon transform has been proven theoretically to have the property of suppressing additive white/"salt & pepper" noise. This is due to the use of an integral function which accumulates pattern's intensity values along straight lines in the spatial domain. These theoretical arguments have been consolidated by experimental results where the proposed generic R-signature and RFM descriptor outperform the commonly used pattern descriptors. However, on an occlusion and deformation dataset, RFM descriptor has comparable performance with comparison descriptors whereas poor performance has been observed from the generic R-signature.

For the generic R-signature, the proper value of the exponent m, the only parameter of the generalization which has been proven to be robust to the level of noise, depends on the

semantic content of images and is constrained by the two contradicting desires: a larger value is preferred for a higher discrimination power whereas a smaller one is for noise robustness. Since the discrimination power results from exploiting variation in the accumulations of patterns along all parallel lines, it is anticipated that a pattern that has less variation in the spatial domain will require a larger value of m for best performance. Evidence is $m \simeq 3.2, 5, 11$ for the three types of datasets used in experiments: object, alphabet, and logo. Moreover, due to the blunt maxima in the accuracy curve of the generic R-signature, a small deviation of the selected value of m from the best choice has almost no effect on the performance.

Complexity consideration

From the definition of the generic R-signature in Subsection 2.2.1, its calculation could be separated into three steps: Radon transform, generic R-transform, and discrete Fourier transform. The Radon transform requires $O(N^2 \log N)$ operations for a pattern image of size $N \times N$. For 1D digital data of M samples, the discrete Fourier transform can be implemented using the FFT algorithm requiring $O(M \log M)$ operations and the remaining generic R-transform requires $O(M)$ operations. Apparently, there is no increase in the computational complexity when generalizing the R-signature. The generic R-signature maintains the simplicity of the conventional R-signature proposed [211], meaning a simple and reasonably fast computation.

Similarly, from the definition of the RFM descriptor in Subsection 2.3.4, its calculation could also be separated into three steps: Radon transform, 1D Fourier–Mellin transform, and discrete Fourier transform. For 1D digital data of M samples, the Mellin transform can be implemented using Eq. (2.27) requiring $O(M)$ operations. Thus, by definition, the complexity of the RFM descriptor is similar to that of the generic R-signature. However, due to the interpretation of the Radon transform by means of the Fourier transform in Eq. (2.8), a computational reduction is possible by fusing these two transforms. If Radon transform is implemented via 2D Fourier transform, due to the successive applications of inverse and forward Fourier transforms, performing the 1D Fourier–Mellin transform on each radial slice of the Radon transform data is equivalent to directly performing the Mellin transform on the corresponding 1D radial slice of the 2D Fourier transform data. This equivalence results in a computational reduction with no change in complexity. For this reason, when the RFM descriptor is chosen to be implemented through 2D Fourier transform, attention to this fusion should be paid for computational benefit.

Chapter 3

Image Analysis by Generic Polar Harmonic Transforms

Contents

3.1 Unit disk-based orthogonal moments

This section provides some basics on the orthogonal moments that are defined on the unit disk: the general definition by means of the inner product and the conditions for them to be orthogonal and "invariant in form" with respect to rotations. A detailed review on existing formulations of unit disk-based orthogonal moments based on polynomials, eigenfunctions, and harmonic functions is also given. All these aspects are followed by a sketch of contributions that will be presented in this chapter.

3.1.1 Definition

Consider the Hilbert space $\mathcal{H}$ of square-integrable continuous complex-valued functions on the unit disk $\mathcal{D} = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$. The theory of image moments over the unit disk is built on the following definition of the inner product $\langle f, V \rangle$ of two functions f and $V \in \mathcal{H}$:

$$\langle f, V \rangle = \iint_{\mathcal{D}} f(x, y) V^*(x, y) \, \mathrm{d}x\mathrm{d}y,$$

where the asterisk $*$ denotes the complex conjugate. The direct geometric interpretation of $\langle f, V \rangle$ is that it is the projection of f onto V or, in other words, $\langle f, V \rangle$ is the information of f that is contained in V.

When a set of functions $\{V_{nm} : (n, m) \in \mathbb{Z}^2\}$ is available, $\{\langle f, V_{nm} \rangle\}$ is the representation of f in the subspace formed by all linear combinations of $\{V_{nm}\}$ and $\{\langle f, V_{nm} \rangle\}$ can further be used as a set of features for the analysis and recognition of f. In this way, $\{V_{nm}\}$ is usually called the set of decomposing *kernels* and, for any square-integrable continuous complex-valued function $f \in \mathcal{H}$, $\{H_{nm} = \langle f, V_{nm} \rangle\}$ is the set of its corresponding *moments*. Since there exists an infinite number of sets of kernels, it is natural to require the set $\{V_{nm}\}$ to have some "structures" that lead to certain beneficial properties in $\{H_{nm}\}$. The two common preferences in image analysis and pattern recognition are *invariance in form* with respect to rotations about the origin and *orthogonality*.

Invariance in form

Invariance in form with respect to rotations about the origin is a "must" structure if the moments H_{nm} are going to be used in rotation-invariant pattern recognition problems. By such invariance, any rotation in the spatial domain by an angle ϕ as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

is equivalent to a transformation of each kernel V_{nm} into a kernel of the same form. In other words, V_{nm} should satisfies the following relation:

$$V_{nm}(x', y') = G(\phi) V_{nm}(x, y),$$

where G is a periodic continuous function with period 2π and $G(0) = 1$. The following theorem imposes constraints on possible explicit forms of V_{nm}.

Theorem 3.1. *A kernel V_{nm} will be invariant in form with respect to rotations of axes about the origin $x = y = 0$ if and only if, when expressed in polar coordinates (r, θ), it is of the form*

$$V_{nm}(r\cos\theta, r\sin\theta) = R_n(r) A_m(\theta), \tag{3.1}$$

with $A_m(\theta) = \mathrm{e}^{im\theta}$ $(i = \sqrt{-1})$ *and* $R_n(r)$ *could be of any form.*

Proof. Refer to [21] for details. □

The possibility to decompose a kernel V_{nm} into a *radial kernel* R_n and an *angular kernel* A_m due to invariance in form simplifies the definition of the set of decomposing kernels. For example, *rotational moments* (RM) [215] and *complex moments* (CM) [1] are defined using $R_n(r) = r^n$; the continuous *generic Fourier descriptor* (GFD) [243] employs $\mathrm{e}^{i2\pi nr}$ for $R_n(r)$; and *angular radial transform* (ART) [22] uses harmonic functions for $R_n(r)$ as

$$R_n(r) = \begin{cases} 1, & n = 0 \\ \cos(\pi n r), & n \neq 0. \end{cases}$$

It is not difficult to see that the obtained kernels V_{nm} of RM, CM, GFD, and ART are not orthogonal and, as a result, information redundancy exists in the moments H_{nm}, leading to difficulties in image reconstruction and low accuracy in pattern recognition, etc. Undoubtedly, orthogonality between kernels V_{nm} comes as a natural solution to these problems. The preference for orthogonality, which will be discussed below, will impose another constraint, besides $A_m(\theta) = \mathrm{e}^{im\theta}$, on the radial kernels R_n.

Orthogonality

Orthogonality occurs when any two kernels V_{nm} and $V_{n'm'}$ from the set of decomposing kernels are uncorrelated or they are "perpendicular" in geometric term, resulting in no redundancy in the two corresponding moments. The orthogonality condition over the unit disk can be written in the form

$$\langle V_{nm}, V_{n'm'} \rangle = \iint\limits_{x^2+y^2 \leq 1} V_{nm}(x,y) V^*_{n'm'}(x,y)\,\mathrm{d}x\mathrm{d}y = \delta_{nn'}\delta_{mm'},$$

where $\delta_{ij} = [i = j]$ is the Kronecker delta function. Due to the possible decomposition of a kernel into its radial and angular components, the above equation becomes

$$\begin{aligned} \langle V_{nm}, V_{n'm'} \rangle &= \int_0^{2\pi}\int_0^1 R_n(r) A_m(\theta)\, R^*_{n'}(r) A^*_{m'}(\theta)\, r\,\mathrm{d}r\mathrm{d}\theta \\ &= \int_0^1 R_n(r) R^*_{n'}(r)\, r\,\mathrm{d}r \int_0^{2\pi} A_m(\theta) A^*_{m'}(\theta)\,\mathrm{d}\theta. \end{aligned}$$

In addition, from the orthogonality between angular kernels:

$$\int_0^{2\pi} A_m(\theta) A^*_{m'}(\theta)\,\mathrm{d}\theta = \int_0^{2\pi} \mathrm{e}^{im\theta}\mathrm{e}^{-im'\theta}\,\mathrm{d}\theta = 2\pi\delta_{mm'},$$

the remaining condition on the radial kernels is

$$\int_0^1 R_n(r) R^*_{n'}(r)\, r\,\mathrm{d}r = \frac{1}{2\pi}\,\delta_{nn'}. \tag{3.2}$$

The above equation is the regulating condition for the definition of a set of radial kernels R_n in order to have orthogonality between kernels V_{nm}. It will be seen in the next subsection that there exists an infinity of such sets of radial kernels and, therefore, the number of sets of

kernels V_{nm} which are orthogonal over the unit disk is also infinite. However, despite the fact that several different forms of kernels V_{nm} have been used to define different orthogonal moments over the unit disk, a fixed form of the angular kernels $A_m(\theta) = \mathrm{e}^{im\theta}$ due to Theorem 3.1 means that the difference between existing methods lies only in the definition of the radial kernels R_n. The following subsection provides an insight into the derivation/definition of R_n of existing unit disk-based orthogonal moments.

3.1.2 Related works

There exists several proposed methods that have their radial kernels satisfying the condition in Eq. (3.2) and they can be roughly classified into three groups. The first employs *Jacobi polynomials* of order n for R_n by orthogonalizing sequences of polynomial functions or by directly using existing orthogonal polynomials for R_n. The second employs the *eigenfunctions* of the Laplacian or Hamiltonian for V_{nm}. And the last uses *harmonic functions* for R_n, taking advantage of their orthogonality. Members of each group are to be briefly visited in the following by providing their radial kernels' derivation strategies, of which more detailed procedures are available in the corresponding reference.

Shifted Jacobi polynomials

Zernike moments (ZM) [242] and *pseudo-Zernike moments* (PZM) [21]: For a fixed value of the angular order m, the radial kernels R_{nm} of ZM and P_{nm} of PZM are defined to be the polynomials of order n that arise out from Gram–Schmidt orthogonalization of the polynomial sequences $\{r^{|m|}, r^{|m|+2}, r^{|m|+4}, r^{|m|+6}, \ldots\}$ and $\{r^{|m|}, r^{|m|+1}, r^{|m|+2}, r^{|m|+3}, \ldots\}$ respectively with the weighting function r over the range $0 \leq r \leq 1$. It was shown that R_{nm} have the following explicit definition:

$$R_{nm}(r) = \sum_{k=0}^{\frac{n-|m|}{2}} (-1)^k \frac{(n-k)!}{k! \left(\frac{n+|m|}{2}-k\right)! \left(\frac{n-|m|}{2}-k\right)!} r^{n-2k},$$

where $n \in \mathbb{N}$ and $m \in \mathbb{Z}$ satisfying $n - |m| =$ even and $|m| \leq n$. The explicit definition of P_{nm} is

$$P_{nm}(r) = \sum_{k=0}^{n-|m|} (-1)^k \frac{(2n+1-k)!}{k!\,(n+|m|+1-k)!\,(n-|m|-k)!} r^{n-k},$$

where $n \in \mathbb{N}$ and $m \in \mathbb{Z}$ satisfying $|m| \leq n$. Then, it is straightforward that

$$\int_0^1 R_{nm}(r) R^*_{n'm}(r)\, r\, \mathrm{d}r = \frac{1}{2n+2}\, \delta_{nn'},$$
$$\int_0^1 P_{nm}(r) P^*_{n'm}(r)\, r\, \mathrm{d}r = \frac{1}{2n+2}\, \delta_{nn'}.$$

Orthogonal Fourier–Mellin moments (OFMM) [200]: Similar to ZM and PZM, the radial kernels of OFMM are obtained by changing the polynomial sequence to be orthogonalized to be $\{1, r, r^2, r^3, \ldots\}$. It is also not difficult to arrive at the following definition of the radial kernels Q_n of OFMM with $n \in \mathbb{N}$:

$$Q_n(r) = \sum_{k=0}^{n} (-1)^{n+k} \frac{(n+k+1)!}{(n-k)!\,k!\,(k+1)!} r^k.$$

Note that Q_n do not depend on the angular order m and satisfy the following identity:

$$\int_0^1 Q_n(r)Q_{n'}^*(r)\,r\,\mathrm{d}r = \frac{1}{2n+2}\,\delta_{nn'}.$$

Chebyshev–Fourier moments (CHFM) [178]: The radial kernels R_n of CHFM are defined based on the shifted Chebyshev polynomials of the second kind $U_n^\star$ [120] of the same order. By definition, $U_n^\star$ are themselves orthogonal with the weighting function w defined as $w(r) = (r - r^2)^{\frac{1}{2}}$ over the range $0 \le r \le 1$:

$$\int_0^1 U_n^\star(r)\big[U_{n'}^\star(r)\big]^*\,w(r)\,\mathrm{d}r = \frac{\pi}{8}\,\delta_{nn'}.$$

By defining R_n as

$$\begin{aligned} R_n(r) &= \sqrt{\frac{8w(r)}{\pi r}}\,U_n^\star(r) \\ &= \left[\frac{64\,(1-r)}{\pi^2\,r}\right]^{\frac{1}{4}} \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \frac{(n-k)!}{k!\,(n-2k)!}\,(4r-2)^{n-2k}, \end{aligned}$$

it is straightforward that

$$\int_0^1 R_n(r)R_{n'}^*(r)\,r\,\mathrm{d}r = \delta_{nn'}.$$

Pseudo Jacobi–Fourier moments (PJFM) [4]: Similar to CHFM, the radial kernels R_n of PJFM are defined based on the shifted Jacobi polynomials G_n (to be discussed later) of the same order using the following explicit definition:

$$R_n(r) = \left[\frac{2n+4}{(n+3)(n+1)}\,(r-r^2)\right]^{\frac{1}{2}} \sum_{k=0}^{n} (-1)^{n+k} \frac{(n+k+3)!}{(n-k)!\,k!\,(k+2)!}\,r^k,$$

which leads to the following identity:

$$\int_0^1 R_n(r)R_{n'}^*(r)\,r\,\mathrm{d}r = \delta_{nn'}.$$

Shifted Jacobi polynomials [21]: It has been demonstrated recently that all the above polynomial-based radial kernels turn out to be special cases of the shifted Jacobi polynomials G_n which are obtained by orthogonalizing the polynomial sequence $\{1, r, r^2, r^3, \ldots\}$ with a more general weighting function w defined as

$$w(p,q,r) = r^{q-1}(1-r)^{p-q}, \qquad (q > 0, p-q > -1)$$

over the range $0 \le r \le 1$ [177]. The explicit definition of G_n is

$$\begin{aligned} G_n(p,q,r) &= \frac{n!\,(q-1)!}{(p+n-1)!} \sum_{k=0}^{n} (-1)^k \frac{(p+n+k-1)!}{(n-k)!\,k!\,(q+k-1)!}\,r^k \\ &= (-1)^n \frac{n!\,(q-1)!}{(n+q-1)!}\,P_n^{(p-q,q-1)}(2r-1), \end{aligned}$$

where $P_n^{(\alpha,\beta)}$ denote the Jacobi polynomials [120]. The orthogonality property is governed by

$$\int_0^1 G_n(p,q,r)\, G_{n'}(p,q,r)\, w(p,q,r)\, \mathrm{d}r = b_n(p,q)\delta_{nn'},$$

where

$$b_n(p,q) = \frac{n!\left[(q-1)!\right]^2 (p-q+n)!}{(q+n-1)!\,(p+n-1)!\,(p+2n)}.$$

It is evident that for each (p,q) with $q>0$ and $p-q>-1$, the polynomials $J_n^{(p,q)}$ defined as $J_n^{(p,q)}(r) = \sqrt{\frac{w(r)}{r b_n(p,q)}}\, G_n(p,q,r)$ can be used as radial kernels since they satisfy the condition in Eq. (3.2) as

$$\int_0^1 J_n^{(p,q)}(r)\left[J_{n'}^{(p,q)}(r)\right]^* r\, \mathrm{d}r = \delta_{nn'}.$$

By varying the value of p and/or q, an infinite number of sets of radial kernels $\{J_n^{(p,q)} : n \in \mathbb{N}\}$ is obtained, meaning that the number of sets of orthogonal kernels $\{V_{nm}^{(p,q)} : n \in \mathbb{N}, m \in \mathbb{Z}\}$ having the following definition

$$V_{nm}^{(p,q)}(r\cos\theta, r\sin\theta) = J_n^{(p,q)}(r) A_m(\theta)$$

is also infinite. For example, the aforementioned polynomial-based radial kernels are obtained with the following values of p and q:

- ZM: $p = |m|+1, q = |m|+1$
- PZM: $p = 2\,|m|+2, q = 2\,|m|+2$
- OFMM: $p = 2, q = 2$
- CHFM: $p = 2, q = \frac{3}{2}$
- PJFM: $p = 4, q = 3$.

Explicit relations between the radial kernels of existing polynomial-based moments and the shifted Jacobi polynomials G_n are given in Table 3.1. Polynomial-based orthogonal moments have been used extensively in image analysis and pattern recognition and many successful applications have been reported. This is due in part to the long-history reputation of Zernike polynomials in optics and in part to their early adoption for the representation of images. However, despite their popularity and the ease of defining a new set of kernels from scratch by properly choosing the values of p and q, this group of orthogonal moments involves computation of factorial terms, resulting in high computational complexity and numerical instability which often limit their practical usefulness.

Eigenfunctions

Helmholtz equation: A set of orthogonal kernels on the unit disk could be obtained by defining them as the set of eigenfunctions of the Laplacian ∇^2 on the same domain, similar to the

Table 3.1: Relations between the radial kernels of existing polynomial-based moments and the shifted Jacobi polynomials G_n. Each of the existing polynomial-based radial kernels is a special case of G_n obtained by properly setting the values of the two parameters.

Method	Relationship
ZM	$R_{nm}(r) = (-1)^{\frac{1}{2}(n-\lvert m\rvert)} \binom{\frac{1}{2}(n+\lvert m\rvert)}{\frac{1}{2}(n-\lvert m\rvert)} r^{\lvert m\rvert} G_{\frac{1}{2}(n-\lvert m\rvert)}(\lvert m\rvert+1, \lvert m\rvert+1, r^2)$
PZM	$P_{nm}(r) = (-1)^{n-\lvert m\rvert} \binom{n+\lvert m\rvert+1}{n-\lvert m\rvert} r^{\lvert m\rvert} G_{n-\lvert m\rvert}(2\lvert m\rvert+2, 2\lvert m\rvert+2, r)$
OFMM	$Q_n(r) = (-1)^n (n+1)\, G_n(2,2,r)$
CHFM	$R_n(r) = (-1)^n \left[\frac{64\,(1-r)}{\pi^2\, r}\right]^{\frac{1}{4}} (n+1)\, G_n\big(2, \frac{3}{2}, r\big)$
PJFM	$J_n(r) = (-1)^n \left[\frac{(n+1)(n+2)^3(n+3)}{2}(r-r^2)\right]^{\frac{1}{2}} G_n(4,3,r)$

interpretation of Fourier basis as the set of eigenfunctions of ∇^2 on a rectangular domain. In this way, the general solution to the Helmholtz equation $\nabla^2 V + \lambda^2 V = 0$ in polar coordinates,

$$\frac{\partial^2 V}{\partial r^2} + \frac{1}{r}\frac{\partial V}{\partial r} + \frac{1}{r^2}\frac{\partial^2 V}{\partial \theta^2} + \lambda^2 V = 0,$$

needs to be obtained. By using the separable form $V(r\cos\theta, r\sin\theta) = R(r)A(\theta)$, this equation is then separated into

$$\frac{\partial^2 A}{\partial \theta^2} + m^2 A = 0,$$
$$r^2\frac{\partial^2 R}{\partial r^2} + r\frac{\partial R}{\partial r} + \big(\lambda^2 r^2 - m^2\big)R = 0,$$

using the separation constant m. The solutions to these equations are

$$A_m(\theta) = \mathrm{e}^{im\theta},$$
$$R_{nm}(r) = \alpha J_m(\lambda_{nm} r) + \beta Y_m(\lambda_{nm} r),$$

where $m \in \mathbb{Z}$ to ensure the 2π periodicity in A_m; J_m and Y_m are the mth order Bessel functions of the first and second kinds respectively [27]; α and β are constant multipliers. Since Y_m is singular at $r = 0$, a nonsingular requirement of R at the origin leaves

$$R_{nm}(r) = J_m(\lambda_{nm} r).$$

Being the eigenfunctions of the Laplacian, Bessel functions of the first kind possess the following orthogonality property:

$$\int_0^\infty J_m(\lambda_{nm} r) J_m(\lambda_{n'm} r)\, r\, \mathrm{d}r = \frac{1}{\lambda_{nm}}\delta_{nn'}.$$

However, a unit disk domain means that $r \in [0,1]$ and the above integral becomes

$$\int_0^1 J_m(\lambda_{nm} r) J_m(\lambda_{n'm} r)\, r\, \mathrm{d}r$$

$$= \begin{cases} \frac{1}{\lambda_{nm}^2-\lambda_{n'm}^2}\left[\lambda_{nm}J_m(\lambda_{nm})J'_m(\lambda_{n'm}) - \lambda_{n'm}J_m(\lambda_{n'm})J'_m(\lambda_{nm})\right], & \lambda_{nm} \neq \lambda_{n'm} \\ \frac{1}{2}\left[J_m^2(\lambda_{nm}) - J_{m-1}(\lambda_{nm})J_{m+1}(\lambda_{nm})\right], & \lambda_{nm} = \lambda_{n'm}. \end{cases} \tag{3.3}$$

Imposing the orthogonality condition on the right-hand side of the above equation leads to

$$\lambda_{nm}J_m(\lambda_{nm})J'_m(\lambda_{n'm}) - \lambda_{n'm}J_m(\lambda_{n'm})J'_m(\lambda_{nm}) = 0,$$

and different forms of solutions to this equation will result in different orthogonal moments [226]. In the literature, for the purpose of simplicity, only two trivial forms of solutions resulting from two different boundary conditions have been used:

- Dirichlet boundary condition: $J_m(\lambda_{nm}) = 0, \forall n$ (i.e., λ_{nm} should be the nth positive zero of J_m), then

  $$\frac{1}{2}\left[J_m^2(\lambda_{nm}) - J_{m-1}(\lambda_{nm})J_{m+1}(\lambda_{nm})\right] = \frac{J_{m+1}^2(\lambda_{nm})}{2}.$$

  This condition was first employed in [91] for the proposal of *Fourier–Bessel modes* (FBM). A slightly modification by fixing $m = \text{const}$ in the radial kernels leads to *Bessel–Fourier moments* (BFM) [235].

- Neumann boundary condition: $J'_m(\lambda_{nm}) = 0, \forall n$ (i.e., λ_{nm} should be the nth positive zero of J'_m), then

  $$\frac{1}{2}\left[J_m^2(\lambda_{nm}) - J_{m-1}(\lambda_{nm})J_{m+1}(\lambda_{nm})\right] = \begin{cases} \frac{1}{2}, & m = 0, n = 1 \\ \frac{1}{2}\left(1 - \frac{m^2}{\lambda_{nm}^2}\right)J_m^2(\lambda_{nm}), & \text{otherwise.} \end{cases}$$

  This condition was used in [222] for the proposal of *disk–harmonic coefficients* (DHC).

Obviously, each of these two boundary conditions implies a specific behavior of the kernels V_{nm} at the unit disk boundary: Dirichlet condition requires that V_{nm} have zero value, whereas Neumann condition means that V_{nm} have zero slope in the radial direction. Due to these identical restrictions on all V_{nm}, any linear combination of these kernels will result in a function having the same behavior at the disk boundary, that is zero value or zero slope in the radial direction according to the employed boundary condition. The orthogonal set $\{V_{nm}\}$ resulting from any of these two boundary conditions is thus theoretically incomplete, it cannot completely represent many functions on the unit disk. It is easy to find a function that lies outside the function space formed by any of these two sets of kernels.

***Stationary Schrödinger equation**:* By replacing the Laplacian ∇^2 with the simplified Hamiltonian $\hat{H} = -\frac{1}{2}\nabla^2 + \frac{1}{2}r^2$, solutions to the corresponding stationary Schrödinger equation $\hat{H}V = EV$ are pairwise orthogonal and thus could also be used for the definition of an orthogonal set. In Cartesian coordinates, these solutions are defined based on the Hermite polynomials [120], leading to the Hermite transform [149]. In polar coordinates, the equation becomes

$$\frac{\partial^2 V}{\partial r^2} + \frac{1}{r}\frac{\partial V}{\partial r} + \frac{1}{r^2}\frac{\partial^2 V}{\partial \theta^2} + \big(-r^2 + 2E\big)V = 0,$$

and they could also be solved by using the separable form $V(r\cos\theta, r\sin\theta) = R(r)A(\theta)$ as

$$\frac{\partial^2 A}{\partial \theta^2} + m^2 A = 0,$$

$$r^2\frac{\partial^2 R}{\partial r^2} + r\frac{\partial R}{\partial r} + \big(-r^4 + 2Er^2 - m^2\big)R = 0,$$

using the separation constant m. The solution to the angular equation is

$$A_m(\theta) = \mathrm{e}^{im\theta},$$

where $m \in \mathbb{Z}$ to ensure the 2π periodicity in A_m. For the radial equation, applying a change of variables $\xi = r^2$ results in

$$\xi\frac{\partial^2 R}{\partial \xi^2} + \frac{\partial R}{\partial \xi} + \left(-\frac{\xi}{4} + \frac{E}{2} - \frac{m^2}{4\xi}\right)R = 0.$$

This is the differential equation for the Gauss–Laguerre functions defined at ξ as $\mathrm{e}^{-\frac{\xi}{2}}\,\xi^{\frac{|m|}{2}}L_n^{(m)}(\xi)$, where $L_n^{(m)}$ denote the associated Laguerre polynomials [120] and $E = 2n + m + 1$ denote the eigenvalues of energy. Gauss–Laguerre functions are known to possess the following orthogonality property:

$$\int_0^\infty \mathrm{e}^{-\xi}\,\xi^{|m|}L_n^{(m)}(\xi)L_{n'}^{(m)}(\xi)\,\mathrm{d}\xi = \frac{(n+|m|)!}{n!}\,\delta_{nn'}.$$

In this way, the radial kernels R_{nm} could be defined as

$$R_{nm}(r) = \left(\frac{n!}{\pi(n+|m|)!}\right)^{\frac{1}{2}}\mathrm{e}^{-\frac{r^2}{2}}\,r^{|m|}L_n^{|m|}(r^2)$$

and it is straightforward to have

$$\int_0^\infty R_{nm}(r)R^*_{n'm}(r)\,r\,\mathrm{d}r = \frac{1}{2\pi}\,\delta_{nn'}.$$

However, unlike Bessel functions in Eq. (3.3), the integral $\int_0^1 R_{nm}(r)R^*_{n'm}(r)\,r\,\mathrm{d}r$ cannot be expressed in a closed form. This obstacle prevents the use of R_{nm} to define a set of kernels whose members are orthogonal over the unit disk. Instead, $V_{nm}(r\cos\theta, r\sin\theta) = R_{nm}(r)A_m(\theta)$ are orthogonal over the infinite-radius disk and hence using $\{V_{nm}\}$ to represent a function whose domain is a closed region is not optimal. This is because the value of V_{nm} on the region out of the function's domain (i.e., the unit disk) are not used. Nevertheless, due to the use of the Gaussian $\mathrm{e}^{-\frac{r^2}{2}}$, the energy of R_{nm} concentrates near the origin. Scaling the radial axis by a factor β is a common practice used to control this concentration for better representation of objects, R_{nm} then have the following definition

$$R_{nm\beta}(r) = \frac{1}{\beta^{|m|+1}}\left(\frac{n!}{\pi(n+|m|)!}\right)^{\frac{1}{2}}\mathrm{e}^{-\frac{r^2}{2\beta^2}}\,r^{|m|}L_n^{|m|}\left(\frac{r^2}{\beta^2}\right).$$

Originated from physical problems, the above formula of $R_{nm\beta}$ was first used in image processing for the definition of *polar Hermite transform* [150] and then for the construction of *Laguerre–Gauss pyramid* [105]. It was reiterated in [18, 152] in the effort to define *shapelets* for astronomical images.

Harmonic functions

It is well-known in Fourier analysis that the set of complex exponential functions $\{e^{i2\pi nr} : n \in \mathbb{Z}\}$ forms an orthonormal basis for the Hilbert space $\mathcal{H}$ of square-integrable continuous complex-valued functions on the unit interval $[0, 1]$ due to

$$\int_0^1 e^{i2\pi nr} e^{-i2\pi n'r} \, dr = \delta_{nn'} \tag{3.4}$$

and the Riesz–Fischer theorem, which states that a measurable function on $[0, 1]$ is square-integrable if and only if the corresponding Fourier series converges in the space $\mathcal{L}^2([0, 1])$. Similarly, the set of trigonometric functions $\{1, \cos(2\pi nr), \sin(2\pi nr) : n \in \mathbb{Z}_+\}$ also forms an orthogonal basis for $\mathcal{H}$ due to the following integral identities:

$$\int_0^1 \cos(2\pi nr) \, dr = 0, \tag{3.5}$$

$$\int_0^1 \sin(2\pi nr) \, dr = 0, \tag{3.6}$$

$$\int_0^1 \cos(2\pi nr) \cos(2\pi n'r) \, dr = \frac{1}{2} \delta_{nn'}, \tag{3.7}$$

$$\int_0^1 \sin(2\pi nr) \sin(2\pi n'r) \, dr = \frac{1}{2} \delta_{nn'}, \tag{3.8}$$

$$\int_0^1 \cos(2\pi nr) \sin(2\pi n'r) \, dr = 0. \tag{3.9}$$

The integrands in Eqs. (3.4)–(3.9) are "similar in form" with that in Eq. (3.2), except for the absence of the weighting term r, which prevents a direct use of harmonic functions as radial kernels. This obstacle was overcome

- by eliminating r using a multiplicative factor $\frac{1}{\sqrt{r}}$ in the radial kernels to define *radial harmonic Fourier moments* (RHFM) [187]:

$$R_n(r) = \frac{1}{\sqrt{r}} \begin{cases} 1, & n = 0 \\ \sqrt{2} \sin(\pi(n+1)r), & n > 0 \ \& \ n \text{ odd} \\ \sqrt{2} \cos(\pi nr), & n > 0 \ \& \ n \text{ even} \end{cases} \tag{3.10}$$

- or by moving r into the variable of integration dr to be $\frac{1}{2}dr^2$ to define *polar harmonic transforms* in three different forms [239]. *Polar complex exponential transform* (PCET):

$$R_n(r) = e^{i2\pi nr^2}, \tag{3.11}$$

polar cosine transform (PCT):

$$R_n^C(r) = \begin{cases} 1, & n = 0 \\ \sqrt{2} \cos(\pi nr^2), & n > 0 \end{cases} \tag{3.12}$$

and *polar sine transform* (PST):

$$R_n^S(r) = \sqrt{2} \sin(\pi nr^2), \quad n > 0. \tag{3.13}$$

It is easy to verify that the radial kernels of RHFM and PCET in Eqs. (3.10) and (3.11) satisfy the orthogonality condition in Eq. (3.2) and thus their sets of kernels are orthogonal over the unit disk. For the cases of PCT and PST, their radial kernels in Eqs. (3.12) and (3.13) do not directly satisfy the orthogonality condition in Eq. (3.2). However, if a function h to be decomposed by R_n^C or R_n^S is defined on $[-1,1]$ instead of $[0,1]$ as usual, the orthogonal basis for h using trigonometric functions then becomes $\{1, \cos(\pi nr), \sin(\pi nr) : n \in \mathbb{Z}_+\}$. In addition, if h is an even function in the case of PCT and an odd function in the case of PST, the decomposition of h using trigonometric functions simplifies to

- For the case of PCT (even function):

$$\begin{aligned}\int_{-1}^{1} h(r)\,\mathrm{d}r &= 2\int_{0}^{1} h(r)\,\mathrm{d}r,\\ \int_{-1}^{1} h(r)\cos(\pi nr)\,\mathrm{d}r &= 2\int_{0}^{1} h(r)\cos(\pi nr)\,\mathrm{d}r,\\ \int_{-1}^{1} h(r)\sin(\pi nr)\,\mathrm{d}r &= 0.\end{aligned}$$

- For the case of PST (odd function):

$$\begin{aligned}\int_{-1}^{1} h(r)\,\mathrm{d}r &= 0,\\ \int_{-1}^{1} h(r)\cos(\pi nr)\,\mathrm{d}r &= 0,\\ \int_{-1}^{1} h(r)\sin(\pi nr)\,\mathrm{d}r &= 2\int_{0}^{1} h(r)\sin(\pi nr)\,\mathrm{d}r.\end{aligned}$$

These simplifications explain for the definitions of R_n^C and R_n^C in Eqs. (3.12) and (3.13) respectively (i.e., R_n^C uses only the constant and cosine functions whereas R_n^S uses only the sine functions). Since $f(r\cos\theta, r\sin\theta)$ is only defined in the polar domain $[0,1]\times[0,2\pi)$, which corresponds to a unit disk in the Cartesian coordinate system, the above even/odd conditions are satisfied by implicitly defining $f(r\cos\theta, r\sin\theta)$ in the domain $[-1,0)\times[0,2\pi)$. In other words, the radial kernels of PCT and PST also satisfy the orthogonality condition.

It should also be noted that, the radial kernel of RHFM in Eq. (3.10) is actually equivalent to $\frac{1}{\sqrt{r}}\mathrm{e}^{i2\pi nr}$ in terms of representation, similar to the equivalence between different forms of Fourier series (namely trigonometric and complex exponential functions). The resemblance between the complex exponential form of RHFM's radial kernel and PCET's radial kernel provokes a suspicion that they are actually special cases of a generic radial kernel defined based on complex exponential functions. Similar observation also leads to three generic radial kernels defined based on trigonometric functions. Explicit forms of all four generic radial kernels will be derived in the remaining of this chapter, along with discussions on their beneficial properties and their implementation strategies.

3.1.3 Contributions

In pursuing the derivation of moments that are orthogonal over the unit disk using harmonic functions, this chapter makes the following main contributions:

- It provides a unified view on strategies that have been used to define unit disk-based orthogonal moments.
- It introduces four generic harmonic radial kernels which correspond to four sets generic of polar harmonic moments and take existing sets of polar harmonic moments as special cases.
- It proves theoretically that the sets of generic polar harmonic kernels are complete in the Hilbert space of all square-integrable continuous complex-valued functions on the unit disk.
- It proposes several strategies for fast computation of polar harmonic kernels/moments based on recursive computation of complex exponential and trigonometric functions.
- It shows experimentally that, when compared with existing moments of similar nature, the proposed polar harmonic moments are superior in terms of computational complexity and comparable in terms of representation capability and discrimination power.

The remainder of this chapter is organized as follows. Section 3.2 derives the formulas for the generic polar harmonic moments along with a proof on the completeness of their corresponding kernels. Beneficial properties of these moments are presented in Section 3.3 and strategies for their fast computation are then discussed in Section 3.4. Experimental results are given in Section 3.5, and finally conclusions are drawn in Section 3.6.

3.2 The generic polar harmonic transforms

This section presents generic definitions of the polar harmonic transforms along with a discussion on the completeness of the corresponding sets of orthogonal decomposing kernels. A formulation to extend these generic transforms for three-dimensional (3D) patterns is also given.

3.2.1 Definition

In order to formulate the generalization, assuming that the harmonic radial kernel has the generic exponential form $R_{ns}(r) = \kappa(r)\,\mathrm{e}^{i2\pi nr^s}$, where $s \in \mathbb{R}$ and κ is a real function. Then

$$\int_0^1 R_{ns}(r)R^*_{n's}(r)\,r\,\mathrm{d}r = \int_0^1 \kappa^2(r)\,\mathrm{e}^{i2\pi nr^s}\mathrm{e}^{-i2\pi n'r^s}\,r\,\mathrm{d}r.$$

Since $\mathrm{d}r^s = sr^{s-1}\mathrm{d}r = sr^{s-2}r\mathrm{d}r$,

$$\int_0^1 R_{ns}(r)R^*_{n's}(r)\,r\,\mathrm{d}r = \int_0^1 \kappa^2(r)\,\mathrm{e}^{i2\pi nr^s}\mathrm{e}^{-i2\pi n'r^s}\,\frac{1}{sr^{s-2}}\,\mathrm{d}r^s.$$

By letting $\frac{\kappa^2(r)}{sr^{s-2}} = \text{const} = C$,

$$\int_0^1 R_{ns}(r)R^*_{n's}(r)\,r\,\mathrm{d}r = \int_0^1 C\,\mathrm{e}^{i2\pi nr^s}\mathrm{e}^{-i2\pi n'r^s}\mathrm{d}r^s = C\delta_{nn'}.$$

In order to have orthonormality between kernels, it follows from Eq. (3.2) that $C = \frac{1}{2\pi}$. Then $\kappa(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}$ and $R_{ns}(r)$ has the following actual definition:

$$R_{ns}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}\,\mathrm{e}^{i2\pi nr^s}, \tag{3.14}$$

which leads to

$$V_{nms}(x,y) = V_{nms}(r\cos\theta, r\sin\theta) = R_{ns}(r)A_m(\theta) = \sqrt{\frac{sr^{s-2}}{2\pi}}\,\mathrm{e}^{i2\pi nr^s}\mathrm{e}^{im\theta}. \tag{3.15}$$

The *generic polar complex exponential transform* (GPCET) is consequently defined as

$$\begin{aligned} H_{nms} &= \iint\limits_{x^2+y^2\le 1} f(x,y)V^*_{nm}(x,y)\,\mathrm{d}x\mathrm{d}y \\ &= \int_0^{2\pi}\int_0^1 f(r\cos\theta, r\sin\theta)\sqrt{\frac{sr^{s-2}}{2\pi}}\mathrm{e}^{-i2\pi nr^s}\mathrm{e}^{-im\theta}\,r\,\mathrm{d}r\mathrm{d}\theta. \end{aligned} \tag{3.16}$$

By taking s in the above definition as a parameter, R_{ns} is a true generalization of the harmonic radial kernel of PCET [239]: $R_{ns}(r)$ in Eq. (3.14) becomes $R_n(r)$ in Eq. (3.11) when $s=2$, except for a constant multiplicative factor $\frac{1}{\sqrt{\pi}}$. A class of harmonic radial kernels is obtained by changing the value of s; and members of this class share beneficial properties to image representation and pattern recognition. However, each member possesses distinctive characteristics, determined by the actual value of s, that make it more suitable for some particular applications. Some beneficial properties of GPCET will be presented theoretically in Section 3.3 and supported by experimental evidence in Section 3.5. Illustration of the phases of GPCET kernels using four different values of $s = 0.5, 1, 2, 4$ for $\{(n,m) \in \mathbb{Z}^2 : 0 \le n,m \le 3\}$ is given in Fig. 3.1. The phase of V_{nms}, unlike those of the kernels defined based on polynomials or special functions, is the sum of the phases of R_{ns} and A_m, producing a Swiss roll pattern in the phase image when $n \neq 0$ or $m \neq 0$. The difference between these phase images lies in the circular slices due to the values of n, m, and s:

- an increase in n results in a Swiss roll made from a thinner and longer cake.
- an increase in m increases the number of cake layers.
- a change in s corresponds to a change in the uniformity in the thickness of each layer.

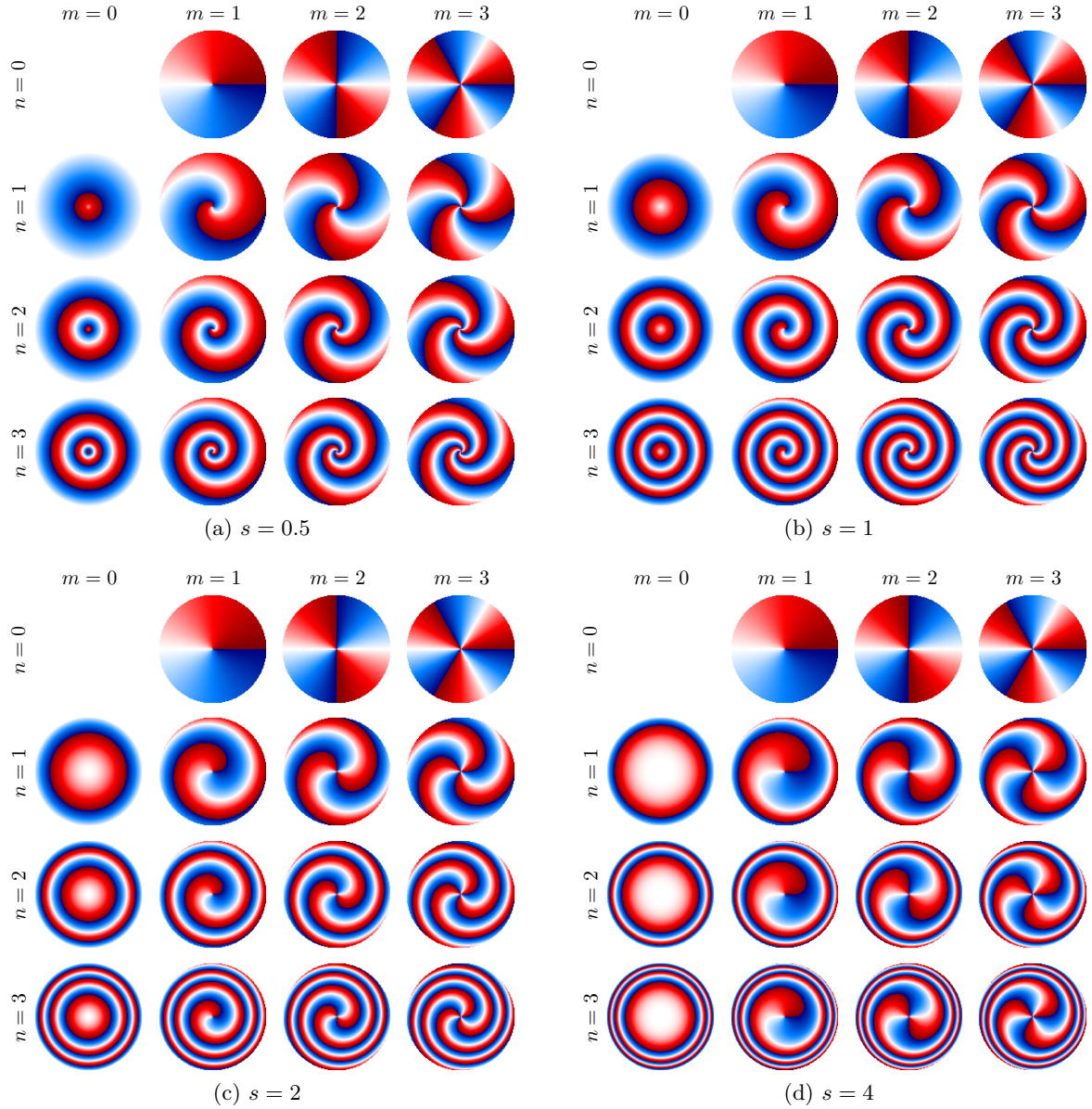
In addition to the generic harmonic radial kernel of GPCET defined in Eq. (3.14), there exist three other generic harmonic radial kernels as generalizations of the harmonic radial kernels of RHFM $\left(R^H_{ns}\right)$ [187], PCT $\left(R^C_{ns}\right)$, and PST $\left(R^S_{ns}\right)$ [239]. The formulations of these generic harmonic radial kernels follow strictly the procedure which has been used for R_{ns}. It is thus not difficult to have

$$R^H_{ns}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}\begin{cases} 1, & n=0 \\ \sqrt{2}\sin(\pi(n+1)r^s), & n>0 \ \ \& \ \ n \text{ odd} \\ \sqrt{2}\cos(\pi nr^s), & n>0 \ \ \& \ \ n \text{ even} \end{cases} \tag{3.17}$$

$$R^C_{ns}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}\begin{cases} 1, & n=0 \\ \sqrt{2}\cos(\pi nr^s), & n>0 \end{cases} \tag{3.18}$$

$$R^S_{ns}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}\sqrt{2}\sin(\pi nr^s), \quad n>0. \tag{3.19}$$

These three generic harmonic radial kernels in turn correspond to three generic transforms: the *generic radial harmonic Fourier moments* (GRHFM), the *generic polar cosine transform* (GPCT), and the *generic polar sine transform* (GPST).



(a) $s = 0.5$ (b) $s = 1$

(c) $s = 2$ (d) $s = 4$

Figure 3.1: 2D views of the phases of GPCET kernels V_{nms} in Eq. (3.15) using four different values of $s = 0.5, 1, 2, 4$ for $\{(n, m) \in \mathbb{Z}^2 : 0 \leq n, m \leq 3\}$. In each sub-figure (for a specific value of s), the row and column indices indicate the values of n and m respectively.

- GRHFM is actually a variant of GPCET in terms of representation, similar to the equivalence between different forms of Fourier series. It becomes RHFM in Eq. (3.10) when $s = 1$, except for a constant multiplicative factor $\frac{1}{\sqrt{2\pi}}$.
- GPCT/GPST arise naturally from GRHFM when the function to be represented by R_{ns}^H is considered as half of an even/odd periodic function. Again, GPCT/GPST become PCT/PST in Eqs. (3.12)/(3.13) when $s = 2$, except for a constant multiplicative factor $\frac{1}{\sqrt{\pi}}$.

Illustration of the real parts of GRHFM, GPCT, and GPST kernels using four different values of $s = 0.5, 1, 2, 4$ for $\{(n, m) \in \mathbb{Z}^2 : 0 \leq n, m \leq 2\}$ (GRHFM, GPCT) or $\{(n, m) \in \mathbb{Z}^2 : 1 \leq n \leq$

$3, 0 \leq m \leq 2\}$ (GPST) is given in Fig. 3.2. The phases of these example kernels are not used for illustration here due to the zero-valued phase in the generic harmonic radial kernels defined in Eqs. (3.17)–(3.19). The phases of V^H_{nms}, V^C_{nms}, and V^S_{nms} are actually the phase of $A_m(\theta)$ and, if employed for illustration, look identical. By fixing the values of (n, m) and for each kernel type, a variation in the value of s corresponds to a variation in the values of the harmonic radial kernels. This variation in turn results in a variation in the values of the real parts of kernels. However, this variation behaves like a deformation in the surface plots of the real parts of kernels and causes a variation in the color patterns as observed from the figure. Moreover, from Eqs. (3.17)–(3.19), it is straightforward that

$$R^H_{ns}(r) = \begin{cases} R^C_{ns}(r), & n \text{ even} \\ R^S_{ns}(r), & n \text{ odd} \end{cases} \quad \Rightarrow \quad V^H_{nms}(x, y) = \begin{cases} V^C_{nms}(x, y), & n \text{ even} \\ V^S_{nms}(x, y), & n \text{ odd}. \end{cases}$$

These relations make the first and last three columns of Fig. 3.2a identical to the first and last three columns of Fig. 3.2b. In a similar manner, the three middle columns of Fig. 3.2a are identical to the three middle columns of Fig. 3.2c.

Orthogonal sets

At a specific value of s, $\langle V_{nms}, V_{n'm's} \rangle = \delta_{nn'}\delta_{mm'}$ means that

$$\mathcal{B}_s = \{V_{nms} : n, m \in \mathbb{Z}\}$$

forms a set of orthonormal kernels on the unit disk. Similarly, there exist three other sets of orthonormal kernels on the unit disk at a specific value of s as follows:

$$\mathcal{B}^H_s = \{V^H_{nms} : n \in \mathbb{N}, m \in \mathbb{Z}\},$$
$$\mathcal{B}^C_s = \{V^C_{nms} : n \in \mathbb{N}, m \in \mathbb{Z}\},$$
$$\mathcal{B}^S_s = \{V^S_{nms} : n \in \mathbb{Z}_+, m \in \mathbb{Z}\},$$

where

$$V^H_{nms}(x, y) = V^H_{nms}(r\cos\theta, r\sin\theta) = R^H_{ns}(r)\,\mathrm{e}^{im\theta},$$
$$V^C_{nms}(x, y) = V^C_{nms}(r\cos\theta, r\sin\theta) = R^C_{ns}(r)\,\mathrm{e}^{im\theta},$$
$$V^S_{nms}(x, y) = V^S_{nms}(r\cos\theta, r\sin\theta) = R^S_{ns}(r)\,\mathrm{e}^{im\theta}.$$

$\mathcal{B}_s$, $\mathcal{B}^H_s$, $\mathcal{B}^C_s$, and $\mathcal{B}^S_s$ each can be used as the set of decomposing orthonormal kernels for GPCET, GRHFM, GPCT, and GPST respectively and thus their completeness is an important issue that needs consideration. The completeness issue will be discussed in the next subsection.

In spite of the relations between GPCET, GRHFM, GPCT, and GPST, at the same value of s, each transform captures different image information. This is due to the difference between Fourier (complex exponential and trigonometric), cosine, and sine series. This difference will be evident from the experimental results that will be given in Section 3.5. Nevertheless, in the remaining of this chapter, theoretical discussions will mainly focus on GPCET with an occasional foray into GRHFM, GPCT, and GPST only when necessary. This is to avoid unnecessary repetition since GRHFM, GPCT, and GPST essentially have many properties that are identical to those of GPCET. In addition, if not explicitly mentioned, the parameter s has a fixed value in the remaining discussions.

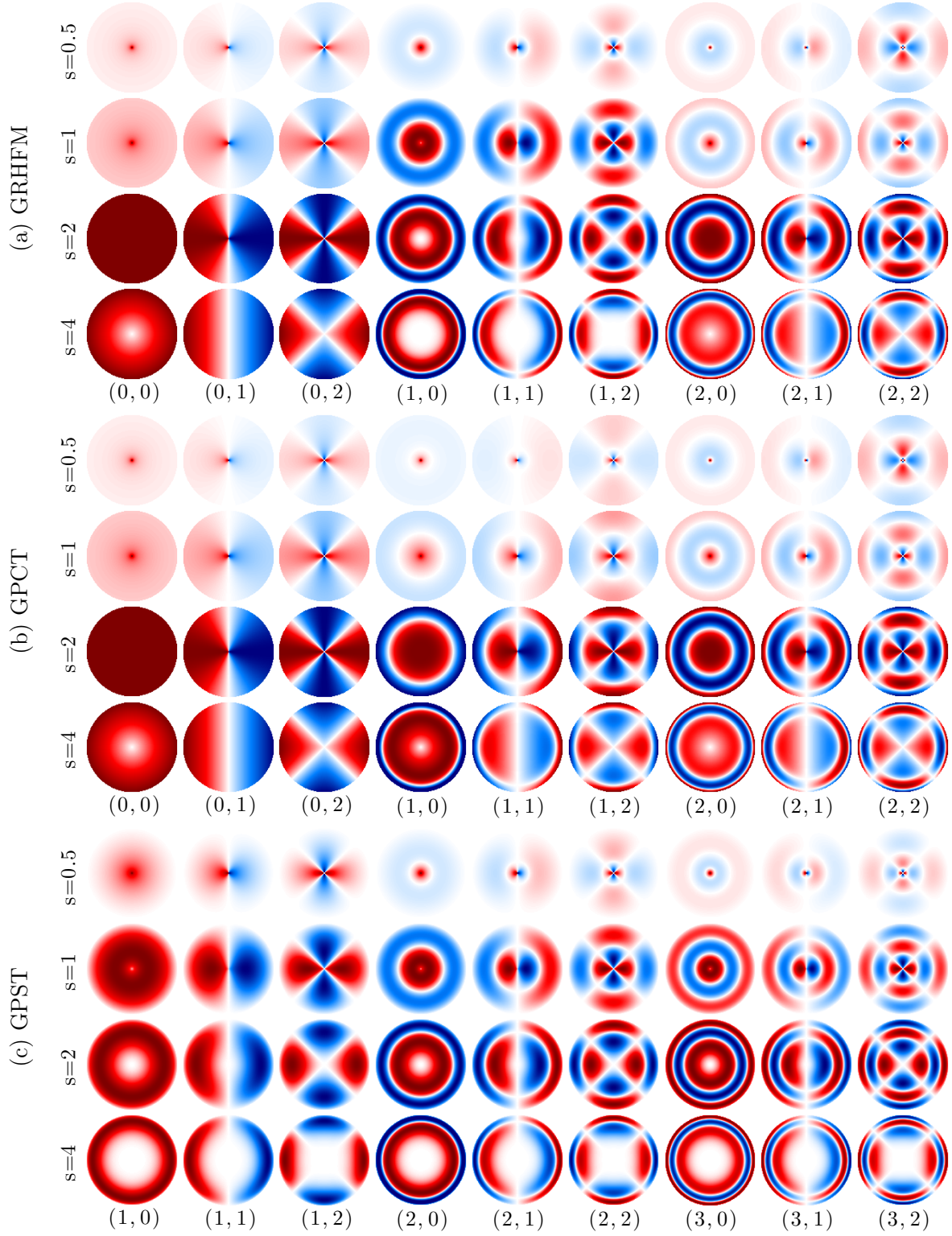


Figure 3.2: 2D views of the real parts of GRHFM, GPCT, and GPST kernels in Eqs. (3.17)–(3.19) respectively using four different values of $s = 0.5, 1, 2, 4$ for $\{(n, m) \in \mathbb{Z}^2 : 0 \leq n, m \leq 2\}$ (GRHFM, GPCT) or $\{(n, m) \in \mathbb{Z}^2 : 1 \leq n \leq 3, 0 \leq m \leq 2\}$ (GPST). In each row (for a specific kernel type and a specific value of s), the images are indexed horizontally by (n, m).

3.2.2 Completeness of $\mathcal{B}_s$

A set of orthogonal kernels is called *complete* in a Hilbert space $\mathcal{H}$ of functions if its linear span is dense in $\mathcal{H}$. The completeness of an orthogonal set in $\mathcal{H}$ is hence related to the ability of the set to represent functions in $\mathcal{H}$. For the case of $\mathcal{B}_s$:

- $\mathcal{H}$ is defined as the space of all square-integrable continuous complex-valued functions on the unit disk $\mathcal{L}^2(x^2+y^2\leq 1)$.

- Being complete makes $\mathcal{B}_s$ an orthonormal basis for $\mathcal{H}$, meaning that every function $f\in\mathcal{H}$ can be represented as an infinite linear combination of the kernels V_{nms} $(n,m\in\mathbb{Z})$ in $\mathcal{B}_s$ as

$$f_s(x,y)=\sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}H_{nms}V_{nms}(x,y) \tag{3.20}$$

  (this equation will be proven in Subsection 3.3.5). In addition, due to the Parseval's identity,

$$\sum_{(n,m)\in\mathbb{Z}^2}|H_{nms}|^2=\iint\limits_{x^2+y^2\leq 1}|f(x,y)|^2\,\mathrm{d}x\mathrm{d}y, \tag{3.21}$$

  GPCET moments H_{nms} are bounded if and only if f is square-integrable over the unit disk. The above identity is in fact stronger than the Bessel's inequality claimed in [239, Eq. (8)], where the loose inequality is replaced by an equality.

In this subsection, the completeness of $\mathcal{B}_s$ in $\mathcal{H}$ is established by means of GPCET's interpretation through Fourier series by rewriting Eq. (3.16) as

$$\begin{aligned}H_{nms}&=\int_0^{2\pi}\left[\int_0^1 f(r\cos\theta,r\sin\theta)\sqrt{\frac{sr^{s-2}}{2\pi}}\,\mathrm{e}^{-i2\pi nr^s}r\,\mathrm{d}r\right]\mathrm{e}^{-im\theta}\,\mathrm{d}\theta\\&=\int_0^{2\pi}\left[\int_0^1\frac{f(r\cos\theta,r\sin\theta)}{\sqrt{2\pi sr^{s-2}}}\,\mathrm{e}^{-i2\pi nr^s}\,\mathrm{d}r^s\right]\mathrm{e}^{-im\theta}\,\mathrm{d}\theta\\&=\frac{1}{2\pi}\int_0^{2\pi}\left[\int_0^1 g(r',\theta)\,\mathrm{e}^{-i2\pi nr'}\mathrm{d}r'\right]\mathrm{e}^{-im\theta}\mathrm{d}\theta,\end{aligned} \tag{3.22}$$

where $r'=r^s$ and

$$g(r',\theta)=\sqrt{\frac{2\pi}{s}}\,(r')^{\frac{2-s}{2s}}\,f\big(\sqrt[s]{r'}\cos\theta,\sqrt[s]{r'}\sin\theta\big). \tag{3.23}$$

In this way, g is a 2D function defined in a Cartesian coordinate system with r and θ-axes are the horizontal and vertical axes respectively. GPCET moments H_{nms} of a function $f\in\mathcal{H}$ are then 2D Fourier coefficients of g formulated as above: first 1D Fourier expansion on the radial slices, then followed by 1D Fourier expansion on the angular slices. This interpretation has transformed the completeness issue of $\mathcal{B}_s$ in $\mathcal{H}$ into the convergence issue of 2D Fourier series, leading to the following two questions:

- The convergence of partial sums of 2D Fourier series of functions? Almost everywhere convergence of "polygonal partial sums" of 2D Fourier series of a function in $\mathcal{L}^2([0,1]\times[0,2\pi))$ is established by Theorem 3.2.

- The square-integrability of g? The necessary and sufficient conditions for the square-integrability of g over the domain $[0,1]\times[0,2\pi)$ are established in Theorem 3.3.

Theorem 3.2. *Let $\mathcal{P}$ be an open polygonal region in $\mathbb{R}^2$ containing the origin. Set $\lambda\mathcal{P} = \{(\lambda x, \lambda y) : (x,y)\in\mathcal{P}\}$ for $\lambda > 0$. Then for*

$$f \sim \sum_{n,m=-\infty}^{\infty} \hat{f}(n,m)\, \mathrm{e}^{i(2\pi nx+my)}$$

in $\mathcal{L}^2([0,1]\times[0,2\pi))$, where $\hat{f}(n,m)$ are the Fourier coefficients of f computed as

$$\hat{f}(n,m) = \frac{1}{2\pi}\int_0^{2\pi}\left[\int_0^1 f(x,y)\,\mathrm{e}^{-i2\pi nx}\,\mathrm{d}x\right]\mathrm{e}^{-imy}\mathrm{d}y.$$

Then

$$f(x,y) = \lim_{\lambda\to\infty}\sum_{n,m\in\lambda\mathcal{P}} \hat{f}(n,m)\,\mathrm{e}^{i(2\pi nx+my)}$$

almost everywhere.

Proof. Refer to [77] for details. □

Theorem 3.3. *The function g defined in Eq. (3.23) is in $\mathcal{L}^2([0,1]\times[0,2\pi))$ if and only if f is in $\mathcal{L}^2(x^2+y^2\le 1)$.*

Proof. From the definition of g:

$$\int_0^{2\pi}\int_0^1 |g(r',\theta)|^2\,\mathrm{d}r'\mathrm{d}\theta = \int_0^{2\pi}\int_0^1 \frac{2\pi}{s}\left(r'\right)^{\frac{2-s}{s}}\left|f\left(\sqrt[s]{r'}\cos\theta,\sqrt[s]{r'}\sin\theta\right)\right|^2\mathrm{d}r'\mathrm{d}\theta.$$

By changing the variable $r = \sqrt[s]{r'} \to r' = r^s$ and $\mathrm{d}r' = sr^{s-1}\mathrm{d}r$, the above equation becomes

$$\begin{aligned}\int_0^{2\pi}\int_0^1 |g(r',\theta)|^2\,\mathrm{d}r'\mathrm{d}\theta &= \int_0^{2\pi}\int_0^1 \frac{2\pi}{s} r^{2-s}|f(r\cos\theta, r\sin\theta)|^2 sr^{s-1}\,\mathrm{d}r^s\mathrm{d}\theta\\ &= 2\pi\int_0^{2\pi}\int_0^1 |f(r\cos\theta, r\sin\theta)|^2\, r\,\mathrm{d}r\mathrm{d}\theta\\ &= 2\pi\iint\limits_{x^2+y^2\le 1} |f(x,y)|^2\,\mathrm{d}x\mathrm{d}y.\end{aligned}$$

Thus, it is straightforward that

$$\int_0^{2\pi}\int_0^1 |g(r',\theta)|^2\,\mathrm{d}r'\mathrm{d}\theta < \infty \quad\Leftrightarrow\quad \iint\limits_{x^2+y^2\le 1} |f(x,y)|^2\,\mathrm{d}x\mathrm{d}y < \infty$$

and the theorem is proven. □

Combining Eq. (3.22) and Theorems 3.2, 3.3, it can be concluded that the set $\mathcal{B}_s = \{V_{nms} : n,m\in\mathbb{Z}\}$ is complete in the Hilbert space $\mathcal{H}$ of all square-integrable continuous complex-valued functions on the unit disk $\mathcal{L}^2(x^2+y^2\le 1)$. As a result, $\mathcal{B}_s$ can be used as an orthonormal basis for $\mathcal{H}$ and writing f as in Eq. (3.20) is safe, meaning that the partial sums converge to the image function. In the literature, there exists no such conclusion for other sets of orthogonal kernels on the unit disk that are defined based on Jacobi polynomials or eigenfunctions. Eq. (3.20) has been extensively used by many authors without judgement on the completeness of the orthogonal sets.

3.2.3 Extension to 3D

GPCET could be easily extended for 3D patterns, similar to the extension of Zernike moments to 3D [35, 162], by replacing the complex exponential function in the circular kernel with a spherical harmonic [114] and then modifying the harmonic radial kernel R_{ns} defined in Eq. (3.14) to fulfill the requirement of orthonormality over the unit sphere. Denoting (r, θ, φ) as the radius, inclination, and azimuth of a spherical coordinate system respectively (Fig. 3.3), the spherical harmonic of degree ℓ and order m, $Y_{m\ell}$, is defined as

$$Y_{m\ell}(\theta, \varphi) = \sqrt{\frac{(2\ell+1)(\ell-m)!}{4\pi(\ell+m)!}}\, P_{m\ell}(\cos\theta)\, \mathrm{e}^{im\varphi},$$

with $\ell \in \mathbb{N}_0$, $m \in \mathbb{Z}$, $|m| \leq \ell$, and $P_{m\ell}$ denotes the associated Legendre functions [68]. Making use of the orthonormality of $Y_{m\ell}$:

$$\langle Y_{m\ell}, Y_{m'\ell'} \rangle = \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi} Y_{m\ell}(\theta, \varphi)\, Y^*_{m'\ell'}(\theta, \varphi)\, \mathrm{d}\Omega = \delta_{mm'}\delta_{\ell\ell'},$$

where Ω represents the solid angle with

$$\mathrm{d}\Omega = \sin\theta\, \mathrm{d}\theta \mathrm{d}\varphi, \tag{3.24}$$

if the generic 3D kernel $V_{nm\ell s}$ is defined as

$$V_{nm\ell s}(x, y, z) = V_{nm\ell s}(r\sin\theta\cos\varphi, s\sin\theta\sin\varphi, r\cos\varphi) = \frac{1}{\sqrt{r}}\, R_{ns}(r)\, Y_{m\ell}(\theta, \varphi)$$

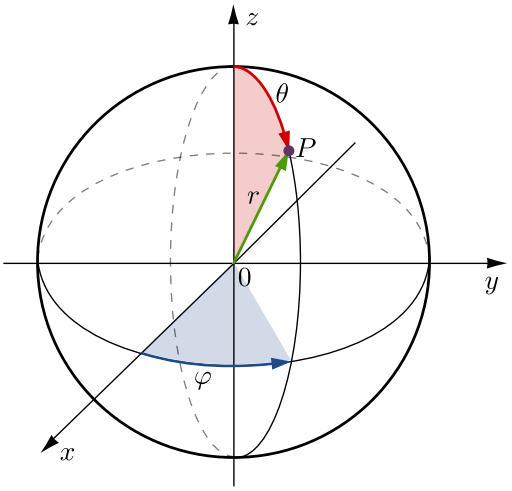
then

$$\begin{aligned}
\langle V_{nm\ell s}, V_{n'm'\ell's} \rangle &= \iiint\limits_{x^2+y^2+z^2\leq 1} V_{nm\ell s}(x, y, z) V^*_{n'm'\ell's}(x, y, z)\, \mathrm{d}x\mathrm{d}y\mathrm{d}z \\
&= \int_0^1 \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{1}{\sqrt{r}}\, R_{ns}(r)\, Y_{m\ell}(\theta, \varphi)\, \frac{1}{\sqrt{r}}\, R^*_{n's}(r)\, Y^*_{m'\ell'}(\theta, \varphi)\, r^2\, \mathrm{d}\Omega \mathrm{d}r \\
&= \int_0^1 R_{ns}(r) R^*_{n's}(r)\, r\, \mathrm{d}r \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi} Y_{m\ell}(\theta, \varphi)\, Y^*_{m'\ell'}(\theta, \varphi)\, \mathrm{d}\Omega \\
&= \delta_{nn'}\delta_{mm'}\delta_{\ell\ell'},
\end{aligned}$$

meaning that $\{V_{nm\ell s} : n, m \in \mathbb{Z}, \ell \in \mathbb{N}_0, |m| \leq \ell\}$ forms a set of orthonormal kernels on the unit sphere ($x^2 + y^2 + z^2 \leq 1$). Orthonormality implies that there is no redundancy in the representation of 3D patterns. Moreover, it should be noted here that $V_{nm\ell s}$ is again a generic 3D kernel with the parameter s, taking the kernel proposed in [202] as a special case ($s = 1$).

For a 3D pattern f confined to the unit sphere, its 3D GPCET moments $H_{nm\ell s}$ can be computed using the following formula:

$$\begin{aligned}
H_{nm\ell s} &= \iiint\limits_{x^2+y^2+z^2\leq 1} f(x, y, z) V^*_{nm\ell s}(x, y, z)\, \mathrm{d}x\mathrm{d}y\mathrm{d}z \\
&= \int_0^1 \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi} f(r\sin\theta\cos\varphi, s\sin\theta\sin\varphi, r\cos\varphi) \frac{1}{\sqrt{r}}\, R^*_{ns}(r)\, Y^*_{m\ell}(\theta, \varphi)\, r^2\, \mathrm{d}\Omega \mathrm{d}r.
\end{aligned}$$



$$r = \sqrt{x^2 + y^2 + z^2}$$
$$\theta = \arctan \frac{\sqrt{x^2+y^2}}{z}$$
$$\varphi = \arctan \frac{y}{x}$$

$$x = r \sin\theta \cos\varphi$$
$$y = r \sin\theta \sin\varphi$$
$$z = r \cos\theta$$

Figure 3.3: Illustration of the 3D Cartesian (x, y, z) and spherical (r, θ, φ) coordinate systems. Shown on the right are the forward and backward coordinate transformations between Cartesian and spherical coordinates of a point P: Cartesian $\rightarrow$ spherical (top right) and spherical $\rightarrow$ Cartesian (bottom right).

The construction of 3D invariants [137] from $\{H_{nm\ell s}\}$ may follow the approaches discussed in Subsection 3.3.2 with minor adaptation. The obtained 3D moments and invariants may then find applications in 3D object recognition, registration, segmentation, etc.

There exists another direction to extend existing 2D moments to 3D by using the complex exponential functions for the two angular directions, i.e., inclination and azimuth [189]. Even though this extension may be used to define rotation-invariant features of 3D patterns, it does not result in 3D kernels that are orthogonal over the unit sphere, since there is no way to overcome the remaining weighting function $\sin\theta$ in the differential solid angle in Eq. (3.24). The lack of orthogonality between decomposing kernels means information redundancy in the computed values of moments and thus is not preferred in many situations.

3.3 Properties

GPCET moments share similar properties with other unit disk-based orthogonal moments due to their similar nature. This similarity accounts for the possible use of GPCET moments to define rotation invariants, to estimate the rotation angle, and to reconstruct the original image functions. In addition, as a direct result of the definition of the generic harmonic radial kernel, GPCET moments possess distinct properties concerning their relation with rotational moments and the distributions of the zeros of their radial kernels. Each of the aforementioned properties will be discussed in detail in this section.

3.3.1 Relation with rotational moments

Since the harmonic radial kernels of GPCET can be expressed in terms of the radial kernels of RM [215] based on Taylor series of exponential functions, $\mathrm{e}^z = \sum_{k=0}^{\infty} \frac{z^k}{k!}$, GPCET moments can be expressed through RM moments as

$$H_{nms} = \int_0^{2\pi} \int_0^1 f(r\cos\theta, r\sin\theta) \sqrt{\frac{sr^{s-2}}{2\pi}} \mathrm{e}^{-i2\pi n r^s} \mathrm{e}^{-im\theta}\, r\, \mathrm{d}r\mathrm{d}\theta$$

$$
\begin{aligned}
&= \int_0^{2\pi}\int_0^1 f(r\cos\theta, r\sin\theta)\sqrt{\frac{sr^{s-2}}{2\pi}}\sum_{k=0}^{\infty}\frac{(-i2\pi n r^s)^k}{k!}\,\mathrm{e}^{-im\theta}\,r\,\mathrm{d}r\mathrm{d}\theta \\
&= \sum_{k=0}^{\infty}\frac{(-i2\pi n)^k\sqrt{s}}{k!\sqrt{2\pi}}\int_0^{2\pi}\int_0^1 f(r\cos\theta, r\sin\theta)\,r^{sk+\frac{s}{2}-1}\mathrm{e}^{-im\theta}\,r\,\mathrm{d}r\mathrm{d}\theta \\
&= \sum_{k=0}^{\infty}\frac{(-i2\pi n)^k\sqrt{s}}{k!\sqrt{2\pi}}\,D_{(sk+\frac{s}{2}-1)m},
\end{aligned}
$$

where

$$
D_{nm} = \int_0^{2\pi}\int_0^1 f(r\cos\theta, r\sin\theta)\,r^n\mathrm{e}^{-im\theta}r\,\mathrm{d}r\mathrm{d}\theta \tag{3.25}
$$

are RM moments. Each GPCET moment H_{nms} is hence an infinite linear combination of RM moments of orders $sk+\frac{s}{2}-1 \in \mathbb{R}$ and $m \in \mathbb{Z}$. In the theory of image moments, real-valued orders are rarely used and it is a common practice to use integer-valued orders. However, in the case of RM moments, integer ordering is not a strict requirement for its radial kernels because orthogonality of the kernels will never be reached. For this reason, the order n of the kernel in Eq. (3.25) can have an arbitrary value. On the contrary, it should be noted here that the order n of the kernel in Eq. (3.15) must be integer-valued to ensure the orthogonality of GPCET kernels.

3.3.2 Rotation invariance

GPCET moments of patterns have an inherent property of rotation invariance. Let f' be the function obtained by rotating clockwise f an angle ϕ $(0 \le \phi < 2\pi)$ about the origin of the Cartesian coordinate system, then

$$
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix},
$$

or $f'(r\cos\theta, r\sin\theta) = f\big(r\cos(\theta-\phi), r\sin(\theta-\phi)\big)$ in polar coordinates. GPCET moments of f', H'_{nms}, are related to those of f, H_{nms}, by

$$
\begin{aligned}
H'_{nms} &= \int_0^{2\pi}\int_0^1 f'(r\cos\theta, r\sin\theta)R^*_{ns}(r)\,\mathrm{e}^{-im\theta}\,r\,\mathrm{d}r\mathrm{d}\theta \\
&= \int_0^{2\pi}\int_0^1 f\big(r\cos(\theta-\phi), r\sin(\theta-\phi)\big)R^*_{ns}(r)\,\mathrm{e}^{-im\theta}\,r\,\mathrm{d}r\mathrm{d}\theta \\
&= \int_0^{2\pi}\int_0^1 f(r\cos\theta', r\sin\theta')R^*_{ns}(r)\,\mathrm{e}^{-im(\theta'+\phi)}\,r\,\mathrm{d}r\mathrm{d}\theta', \\
&= \mathrm{e}^{-im\phi}H_{nms},
\end{aligned} \tag{3.26}
$$

where $\theta' = \theta - \phi$. The above equation is the basis for the derivation of rotation invariants from GPCET moments.

Since a rotation in the spatial domain only influences the phase of GPCET moments, any approach to the construction of rotation invariants from GPCET moments should be based on a proper kind of phase cancelation. The classical method is to overcome the exponential factor $\mathrm{e}^{-im\phi}$ by a magnitude operator as

$$
|H'_{nms}| = |H_{nms}|, \tag{3.27}
$$

or similarly $H'_{nms}\left[H'_{nms}\right]^* = H_{nms}H^*_{nms}$. However, these simple forms of rotation invariants have been known to discard too much information from patterns: by taking the magnitude only, some useful information is missed [166] and the resulting invariants do not generate a complete set of invariants. For this problem, a more complex formulation [81], where phase cancelation is achieved by multiplication of a number of appropriate moment powers, could be employed due to the following theorem.

Theorem 3.4. *Let $N \geq 1$ and $n_i, m_i, k_i \in \mathbb{Z}$ $(i = 1, \ldots, N)$ such that*

$$\sum_{i=1}^{N} k_i m_i = 0. \tag{3.28}$$

Then, any product I_s defined as

$$I_s = \prod_{i=1}^{N} H^{k_i}_{n_i m_i s} \tag{3.29}$$

is invariant to rotation.

Proof. The invariant I'_s of the function f', a rotated version of f by an angle ϕ, is computed by definition as

$$I'_s = \prod_{i=1}^{N} H'^{k_i}_{n_i m_i s} = \prod_{i=1}^{N} \mathrm{e}^{-ik_i m_i \phi} H^{k_i}_{n_i m_i s} = \mathrm{e}^{-i\phi \sum_{i=1}^{N} k_i m_i} \prod_{i=1}^{N} H^{k_i}_{n_i m_i s} = \prod_{i=1}^{N} H^{k_i}_{n_i m_i s} = I_s,$$

which is the corresponding invariant of f. □

It is evident that the magnitude operator in Eq. (3.27) is a special case of the invariant in Eq. (3.29). The magnitude operator is obtained when $N = 2$, $k_1 = k_2 = 1$, $n_1 = -n_2 = n$, and $m_1 = -m_2 = m$:

$$I_s = H_{nms}H_{-n-ms} = H_{nms}H^*_{nms} = |H_{nms}|^2,$$

due to

$$\begin{aligned} H_{-n-ms} &= \int_0^{2\pi}\int_0^1 f(r\cos\theta, r\sin\theta)\sqrt{\frac{sr^{s-2}}{2\pi}}\mathrm{e}^{i2\pi n r^s}\mathrm{e}^{im\theta}\, r\,\mathrm{d}r\mathrm{d}\theta \\ &= \left[\int_0^{2\pi}\int_0^1 f(r\cos\theta, r\sin\theta)\sqrt{\frac{sr^{s-2}}{2\pi}}\mathrm{e}^{-i2\pi n r^s}\mathrm{e}^{-im\theta}\, r\,\mathrm{d}r\mathrm{d}\theta\right]^* = H^*_{nms}. \end{aligned}$$

The invariant I_s is in general complex-valued. If real-valued features are preferred, the real and imaginary parts (or equivalently the magnitude and phase) of I_s can be used instead. In addition, Theorem 3.4 allows the construction of an infinite number of invariants for any order of moments. Let $\mathcal{I}_s = \{I_{s1}, \ldots, I_{sk} : k \in \mathbb{Z}_+\}$ be the set of all rotation invariants being computed using Eq. (3.29), it is evident that only few members of $\mathcal{I}_s$ are mutually independent. Here naturally comes the problem of constructing a complete set of independent invariants, i.e., a basis of invariants, by means of which all other invariants can be generated using only multiplication, exponentiation with an integer exponent, and complex conjugation. The following theorem serves as a guideline for the construction of a basis of invariants up to a given order.

Theorem 3.5. *Let consider GPCET moments up to the order $\ell \geq 2$ and a set of rotation invariants $\mathcal{B}_s^I$ constructed as follows:*

$$\mathcal{B}_s^I = \{\Psi_{nms} = H_{nms}\big(H^*_{n_0m_0s}\big)^m : |n| + |m| \leq \ell\}, \tag{3.30}$$

where $m_0 = -1$, n_0 is an arbitrary index such that $|n_0| \leq \ell - 1$, and $H_{n_0m_0s} \neq 0$. Then $\mathcal{B}_s^I$ is a basis for all rotation invariants created from the moments up to the order ℓ.

Proof. It is clear that Ψ_{nms} is a rotation invariant of the form in Eq. (3.29) since the condition in Eq. (3.28) is satisfied. There are hence two remaining issues to be addressed in order to prove that $\mathcal{B}_s^I$ is a basis for all rotation invariants: the completeness and the independence of its members.

- *Completeness* of $\mathcal{B}_s^I$: Let I_s be an arbitrary member of $\mathcal{I}_s$, then $I_s = \prod_{i=1}^N H^{k_i}_{n_im_is}$ with $\sum_{i=1}^N k_im_i = 0$, or $\big(H^*_{n_0m_0s}\big)^{\sum_{i=1}^N k_im_i} = 1$. I_s can be rewritten as

$$I_s = \big(H^*_{n_0m_0s}\big)^{\sum_{i=1}^N k_im_i} \prod_{i=1}^N H^{k_i}_{n_im_is} = \prod_{i=1}^N \Big[H_{n_im_is}\big(H^*_{n_0m_0s}\big)^{m_i}\Big]^{k_i} = \prod_{i=1}^N \Psi^{k_i}_{n_im_is}.$$

  This equation means that I_s can be represented by members of $\mathcal{B}_s^I$ or, in other words, $\mathcal{B}_s^I$ as defined in Eq. (3.30) is a complete basis for $\mathcal{I}_s$.

- *Independence* of $\mathcal{B}_s^I$: Assuming that members of $\mathcal{B}_s^I$ are dependent, i.e., there exists $\Psi_{nms} \in \mathcal{B}_s^I$ such that it depends on $\mathcal{B}_s^I \setminus \Psi_{nms}$. From the independence of GPCET kernels and (or equivalently the independence of GPCET moments themselves), it follows that $n = n_0$ and $m = m_0$. Without loss of generality, assuming that $\Psi_{n_0m_0s}$ depends on $\{\Psi_{p_iq_is} \in \mathcal{B}_s^I \setminus \Psi_{n_0m_0s} : i = 1, \ldots, N\}$ as

$$\Psi_{n_0m_0s} = \prod_{i=1}^{N'} \Psi^{k_i}_{p_iq_is} \prod_{i=N'+1}^{N} \big(\Psi^*_{p_iq_is}\big)^{k_i},$$

  where $k_i \in \mathbb{Z}$, or

$$H_{n_0m_0s}\big(H^*_{n_0m_0s}\big)^{m_0} = \prod_{i=1}^{N'} H^{k_i}_{p_iq_is} \prod_{i=1}^{N'} \big(H^*_{n_0m_0s}\big)^{q_ik_i} \prod_{i=N'+1}^{N} \big(H^*_{p_iq_is}\big)^{k_i} \prod_{i=N'+1}^{N} H^{q_ik_i}_{n_0m_0s}.$$

  Equating the exponents of GPCET moments on both sides while taking into account their mutual independence leads to

$$\begin{cases} \sum_{i=N'+1}^{N} q_ik_i = 1 \\ \sum_{i=1}^{N'} q_ik_i = m_0 \\ k_i = 0, \qquad\qquad i = 1, \ldots, N \end{cases}$$

  This system of equations has no solution, meaning that the initial assumption on the dependence of the members of $\mathcal{B}_s^I$ does not hold. It thus can be concluded here that the members of $\mathcal{B}_s^I$ are independent.

Being complete and having independent members make $\mathcal{B}_s^I$ a basis for all rotation invariants of the form in Eq. (3.29). The theorem is proven. □

3.3.3 Rotation angle estimation

The relative rotation angle between two similar patterns remains in the phase of GPCET moments and could be eliminated in the definition of rotation invariants in the previous subsection. This angle, on the other hand, could also be estimated by directly using the phase information of GPCET moments. By using Eq. (3.26), the phase shift Θ_{nms} of the GPCET moment of order (n, m) resulting from a rotation in the spatial domain by an angle ϕ is

$$\Theta_{nms} = \arg\left(\frac{H'_{nms}}{H_{nms}}\right) = m\phi. \tag{3.31}$$

From this equation, the estimated rotation angle of order (n, m) can be simply obtained as

$$\hat{\phi}_{nms} = \frac{\Theta_{nms}}{m}. \tag{3.32}$$

However, since $0 \leq \phi < 2\pi$ then $0 \leq \Theta_{nms} < 2m\pi$ according to Eq. (3.31) whereas the actual measurable value of Θ_{nms} is in the range $[0, 2\pi)$. In this case, the ideal phase shift Θ_{nms} is a combination of the measurable phase shift Φ_{nms} $(0 \leq \Phi_{nms} < 2\pi)$ and an integer multiple of 2π:

$$\Theta_{nms} = \Phi_{nms} + 2k\pi, \quad k = 0, \ldots, m-1.$$

Eq. (3.32) may then yield m solutions that correspond to m possible values of k:

$$\hat{\phi}_{nms} = \frac{\Phi_{nms}}{m} + \frac{2k\pi}{m}, \quad k = 0, \ldots, m-1. \tag{3.33}$$

It is evident that only one of these m solutions corresponds to the correct rotation angle ϕ and there is a need for a proper value for k. A probabilistic approach, which has been used for rotation angle estimation using Zernike moments [117], can be employed here for GPCET moments. The probability density function of the estimated rotation angle $\hat{\phi}_s$ is defined as

$$P(\hat{\phi}_s) = \sum_n \sum_m \xi_{nms} P(\hat{\phi}_{nms}), \quad 0 \leq \hat{\phi}_s < 2\pi,$$

where ξ_{nms} is a weighting factor and $P(\hat{\phi}_{nms})$ is the value of the probability density function of the rotation angle estimated by the GPCET moment of order (n, m). ξ_{nms} is usually chosen to be proportional to $|H_{nms}|$ since a moment of a higher magnitude should be affected relatively less by noise. $P(\hat{\phi}_{nms})$ is originally defined as a convolution of an impulse chain with a scaled Gaussian kernel:

$$P(\hat{\phi}_{nms}) = \frac{1}{m} \sum_{k=0}^{m-1} \delta\left\{\hat{\phi}_{nms} - \left(\frac{\Phi_{nms}}{m} + \frac{2k\pi}{m}\right)\right\} * \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{\hat{\phi}_{nms}^2}{2\sigma^2}},$$

where $\sigma = \frac{\pi}{4m}$. Finally, the rotation angle is estimated by

$$\hat{\phi}_s = \underset{\hat{\phi}_s}{\operatorname{argmax}}\, P(\hat{\phi}_s).$$

Similar formulation to the above was also proposed recently [43]. The method does not employ a probabilistic framework, it instead computes $\hat{\phi}_s$ iteratively based on Eq. (3.33). In case of additive noise and/or unintentional image variations (i.e., image distortion caused by re-sampling and re-quantization after rotation or even elastic distortion), a more accurate estimation method [188], which casts the rotation angle estimation as an optimization problem, could also be employed for GPCET moments.

Table 3.2: The numbers of zeros of the nth-order radial kernels of existing unit disk-based orthogonal moments in the interval $(0, 1)$ (exclusive of $r = 0$ and $r = 1$ if available).

Method	Number of zeros	Method	Number of zeros
ZM	$\frac{n-m}{2}$	BFM	n
PZM	$n - m$	DHC	n
OFMM	n	GPCET	$2n$
CHFM	n	GRHFM	n
PJFM	n	GPCT	n
FBM	n	GPST	$n - 1$

3.3.4 Zeros of R_{ns}

The number of zeros of a radial kernel corresponds to the capability of the moment in representing high-frequency components in patterns. For the case of GPCET, R_{ns} is defined based on complex exponential function and can be rewritten in the following form:

$$R_{ns}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}} \Big[\cos(2\pi n r^s) + i \sin(2\pi n r^s) \Big].$$

The two equations

$$\begin{aligned} \text{real}(R_{ns}(r)) &= 0, \\ \text{image}(R_{ns}(r)) &= 0, \end{aligned}$$

which result from $R_{ns}(r) = 0$ each has $2n$ distinct roots in the interval $(0, 1)$. For a better perception of how large this number is, Table 3.2 provides the numbers of zeros of the nth-order radial kernels of existing unit disk-based orthogonal moments. It is observed that, except for ZM, PZM, and GPCET, the nth-order radial kernels of all other methods have approximately n zeros. For the case of GPCET, its number is almost double whereas, for ZM and PZM, their numbers depend on the angular order m. In order to have the same number of zeros n_0 as other methods, the orders of the radial kernels of ZM and PZM have to be $2n_0 + m$ and $n_0 + m$ respectively, much greater than that of GPCET, which is only $\frac{n_0}{2}$.

In addition to the quantity, the distribution of zeros is also an important property of a radial kernel since it relates to the *information suppression problem* [1]. Suppression is the situation when the computed values of moments put emphasis on certain regions and neglect the rest. When the essential discriminative information is distributed uniformly in the spatial domain, unfair emphasis of the extracted moments on certain regions has been shown to have a negative impact on the discrimination quality. On the contrary, when the essential discriminative information only exists in certain regions, it is preferable to move the emphasis towards those regions. In the case of GPCET, the distribution of zeros of its radial kernels can be controlled by the parameter s. This is the distinctive property of GPCET that existing methods do not have, they only have fixed distributions of zeros that depend on the definitions of their radial kernels.

When $s = 1$, the zeros of R_{n1} are distributed uniformly, meaning a uniform emphasis over the unit disk. The more deviation of the value of s from 1 is, the more "biased" to the inner (when $s < 1$) or outer (when $s > 1$) regions of the unit disk the distribution of zeros is, which in turn corresponds to the more emphasis on the inner or outer regions of patterns respectively.

The suppression can thus be controlled for particular purposes as demonstrated later in the experimental section. Evidence for the observations on the quantity and distribution of zeros of R_{ns} is given in Fig. 3.4 containing the plots of real$(R_{ns}(r))$ and image$(R_{ns}(r))$ of orders $n = 0, 1, \ldots, 4$ at $s = 0.5, 1, 2, 4$. It is clear from the figure that the real and imaginary parts of GPCET radial kernel of order n each has $2n$ zeros in the interval $(0, 1)$. Moreover, the distribution of these zeros is biased towards 0 at $s = 0.5$, uniform at $s = 1$, and biased towards 1 at $s = 2, 4$.

3.3.5 Image reconstruction

Since the kernels of GPCET form an orthonormal basis on the unit disk, the following theorem states that GPCET expansion gives the best $\mathcal{L}^2$ approximation to a function among all infinite linear combinations of similar GPCET kernels.

Theorem 3.6. *If f is in $\mathcal{L}^2(x^2+y^2 \leq 1)$ and $\mathcal{S}$ is a subset of $\mathbb{Z}^2$, for any set of complex numbers $\{\alpha_{nm} : (n, m) \in \mathcal{S}\}$, then*

$$\left\| f(x,y) - \sum_{(n,m)\in\mathcal{S}} H_{nms} V_{nms}(x,y) \right\| \leq \left\| f(x,y) - \sum_{(n,m)\in\mathcal{S}} \alpha_{nm} V_{nms}(x,y) \right\|.$$

Furthermore, equality holds only when $\alpha_{nm} = H_{nms}$.

Proof. Refer to [194, Theorem 4.14] for details. □

Another way of stating Theorem 3.6 is that the orthogonal projection of f onto the subspace of $\mathcal{L}^2(x^2 + y^2 \leq 1)$ spanned by $\{V_{nms} : (n, m) \in \mathcal{S} \subset \mathbb{Z}^2\}$ is

$$\hat{f}_s(x,y) = \sum_{(n,m)\in\mathcal{S}} H_{nms} V_{nms}(x,y), \tag{3.34}$$

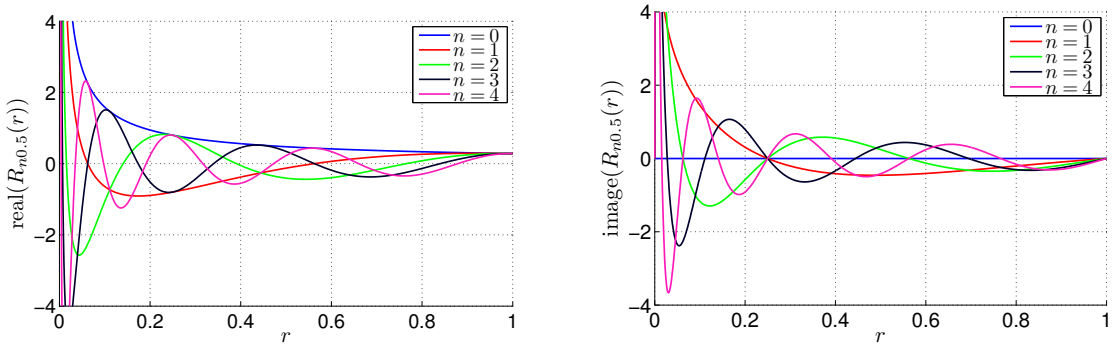
and $\hat{f}_s$ is interpreted as the reconstruction of f from the set of corresponding moments $\{H_{nms} : (n, m) \in \mathcal{S}\}$. It is straightforward here that, due to the completeness of $\mathcal{B}_s$, Eq. (3.34) becomes Eq. (3.20) when $\mathcal{S} = \mathbb{Z}^2$. In other words, any function can be expressed as an infinite linear combination of the GPCET kernels. It is owning to this result that writing f as in Eq. (3.20) is safe. The reconstruction error is then

$$\begin{aligned} \epsilon_s^2 &= \iint\limits_{x^2+y^2\leq 1} \left[f(x,y) - \hat{f}_s(x,y) \right]^2 \mathrm{d}x\mathrm{d}y \\ &= \iint\limits_{x^2+y^2\leq 1} \left[\sum_{(n,m)\in\mathbb{Z}^2\setminus\mathcal{S}} H_{nms} V_{nms}(x,y) \right]^2 \mathrm{d}x\mathrm{d}y = \sum_{(n,m)\in\mathbb{Z}^2\setminus\mathcal{S}} H_{nms}^2, \end{aligned} \tag{3.35}$$

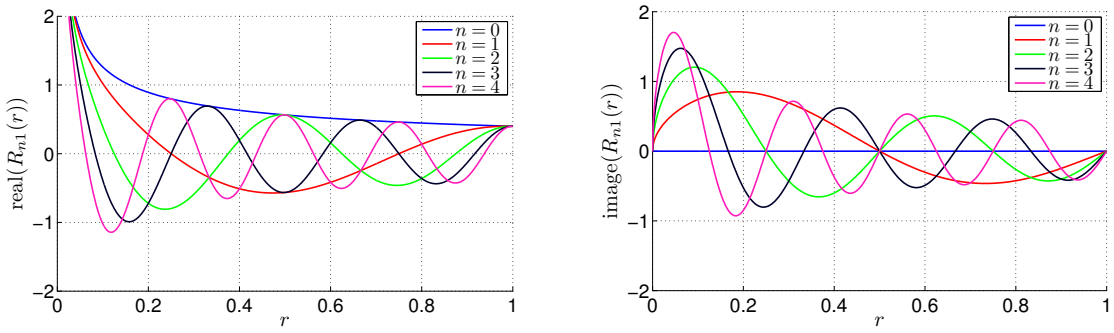
and the Parseval's identity in Eq. (3.21) is obtained naturally when $\mathcal{S} = \varnothing$.

3.4 Implementation

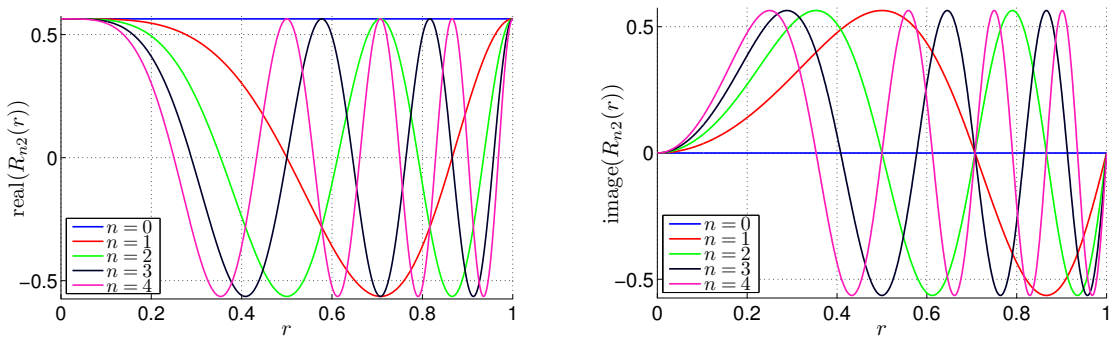
In practice, the images processed in digital systems are not defined in a continuous domain but a grid of pixels over which the image functions have constant values. Accordingly, the formula for computing GPCET moments in Eq. (3.16) needs to be discretized in a proper way. Depending on the actual discretization, there may, or may not, exist fast implementation of GPCET, relying on the existence of any inherent structure in the computation. This section discusses the discretization and implementation strategies for the computation of GPCET moments, followed by a discussion on their numerical stability.



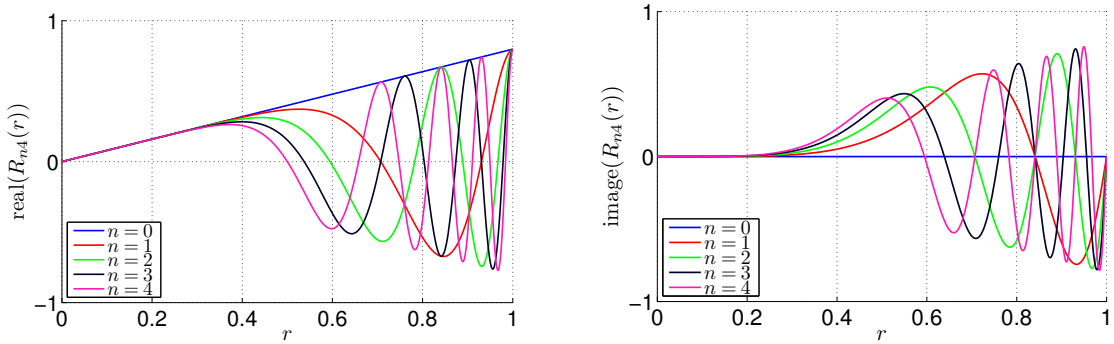
(a) $s = 0.5$



(b) $s = 1$

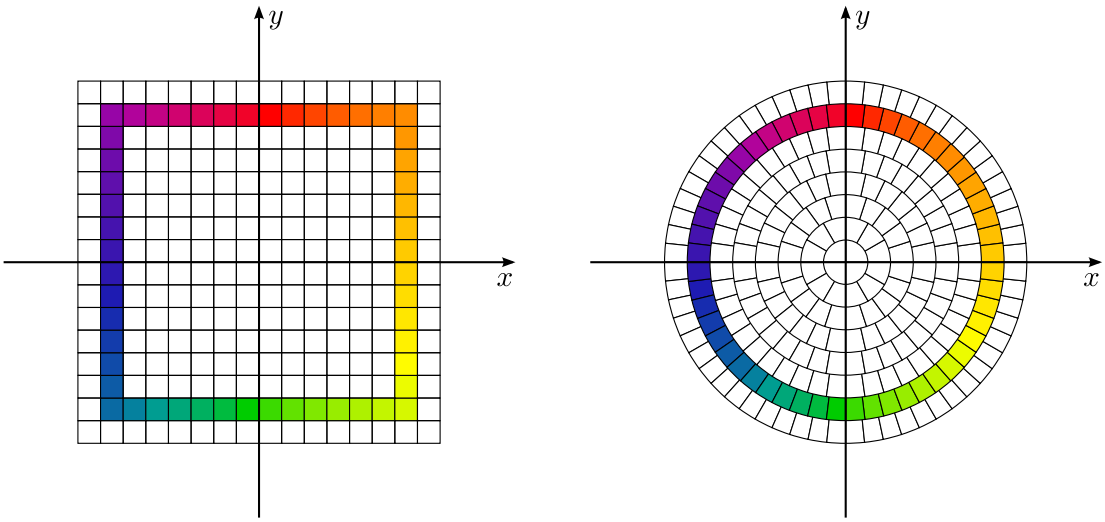


(c) $s = 2$



(d) $s = 4$

Figure 3.4: Real and imaginary parts of GPCET radial kernels of orders $n = 0, 1, \ldots, 4$ at $s = 0.5, 1, 2, 4$. These sub-figures demonstrate clearly that the real and imaginary parts of GPCET radial kernel of order n each has $2n$ zeros in the interval $(0, 1)$. The distribution of these zeros is uniform when $s = 1$ and biased towards 0 or 1 depending on whether $s < 1$ or $s > 1$.



(a) Square image (b) Circular image

Figure 3.5: Square-to-disk transformation of an image of size 16×16: (a) the input square image, (b) the transformed circular image. Each pixel in the square image is transformed into a circular trapezoid (the area enclosed within two radii and two concentric disks) in the circular image. The values of the circular image are piecewise-constant over circular trapezoids.

3.4.1 Discrete approximation

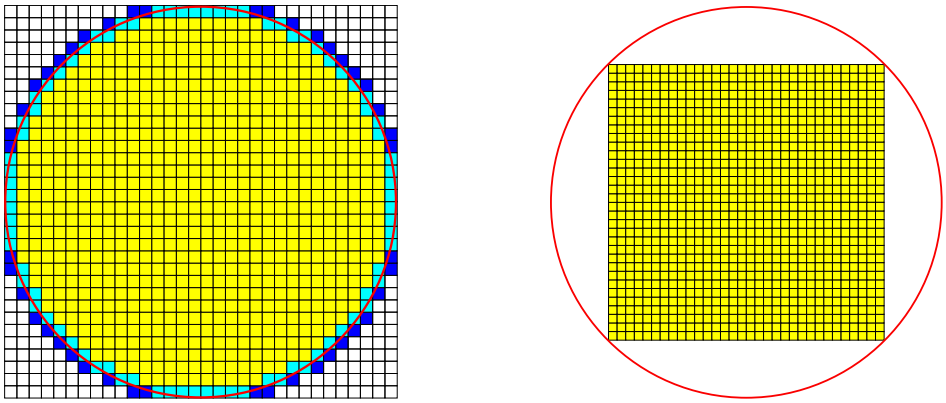
The continuous integration in Eq. (3.16) needs to be approximated in two separate ways: in the domain of the integration (geometrically) and in the values of GPCET kernels (numerically). More precisely, for geometric approximation, a lattice of pixels is selected to cover the unit disk and, for numerical approximation, a value is assigned to each GPCET kernel over a pixel region.

Geometric approximation

Geometric approximation is required to linearly map a rectangular grid of pixels onto a continuous and circular domain of the unit disk since an exact mapping is impossible due to their different geometric natures. There exist several strategies for this type of approximation, each has its own advantages and disadvantages:

- *Square-to-disk* [157]: The first strategy is to transform the square domain of an input image into a disk as illustrated in Fig. 3.5 where the circular color ring is obtained by transforming the square color ring. The result of this transformation is a circular image having piecewise-constant values over circular trapezoids. This allows a direct and exact computation of moments of the circular image using Eq. (3.16) by means of variable separation. However, this advantage has to be paid by the rotation-invariance property of the computed moments. From the two color rings shown in the figure, it is not difficult to see that a spatial rotation in the square image does not correspond to a spatial rotation in the circular image, and vice versa. In addition, if moments are computed from the circular image, they reflect the content of the circular image, not the input square image. These disadvantages have obviously limited the use of this strategy in invariant patterns recognition problems.

- *Incircle* [135]: The second strategy is based on the idea of putting the disk inside the



(a) Incircle (b) Circumcircle

Figure 3.6: Lattice-point approximations of a circular region of an image of 32×32 pixels using incircle (a) and circumcircle (b). There are four types of pixels according to the relative intersection between the image and the disk: (1) *white*: pixels that have no overlap with the disk (2) *blue* or (3) *cyan*: pixels that partially intersect the disk and have their centers lying outside or inside the disk respectively (4) *yellow*: pixels that lie entirely inside the disk.

image's rectangular domain as illustrated in Fig. 3.6a. In this way, almost all the circular region is employed. However, some image pixels lie outside the circular region (the white pixels) and are not included in the computation of GPCET moments. This exclusion will cause information loss when the image is to be reconstructed by using Eq. (3.20). Moreover, using incircle also leads to the consideration of image pixels that lie at the disk boundary (the blue and cyan pixels). Conventionally, pixels that have their centers lying outside the circular region (the blue pixels) are treated like the white pixels (i.e., not used in the computation process). Using the remaining cyan pixels means that there exists an inherent error, called *geometric error* in the literature, in the computation of GPCET moments because each cyan pixel has a part lying outside the circular region. The impact of this type of error on the computed moments depends heavily on the behavior of the radial kernels around the point $r = 1$. This impact thus varies according to the actual moment and the order n.

- *Circumcircle* [45]: Another strategy is to circumscribe the image's rectangular domain by a disk as illustrated in Fig. 3.6b to overcome the disadvantages of the above incircle strategy. This strategy has the advantage of being able to map all the image's rectangular domain onto the unit disk; there is no pixel or part of pixel loss and thus all information contained in the image is used in the computation of GPCET moments. The geometric error thus does not exist in the computed GPCET moments. However, the maximum effective mapped region is only $\frac{2}{\pi}$ the whole disk due to the unused four white segments, making this type of approximation less powerful in terms of representation capability. As a result, when compared to the aforementioned incircle strategy, this strategy requires a higher order of GPCET moments in order to reconstruct images of similar quality. This is because of the scaling process to have a smaller image that can be fitted inside the unit disk: image details that can be reconstructed effectively at a certain order by using incircle require a higher order when circumcircle is used. The ratio between these orders is approximately equal to the scaling factor, that is $\sqrt{2}$.

In this work, in order to demonstrate the ability of GPCET in representing image functions inside the unit disk, the second strategy using an incircle is employed for geometric approximation. Let f be the digital image of size $M \times N$, the domain $[0, M] \times [0, N]$ of f is mapped onto $[-1, 1] \times [-1, 1]$ by

$$i \longrightarrow \frac{2i - M}{M} = i\Delta x - 1, \qquad 0 \leq i \leq M \tag{3.36}$$

$$j \longrightarrow \frac{2j - N}{N} = j\Delta y - 1, \qquad 0 \leq j \leq N \tag{3.37}$$

where $\Delta x = \frac{2}{M}$ and $\Delta y = \frac{2}{N}$. Furthermore, in order to avoid the geometric error, only the image pixels that lie entirely inside the unit disk (the yellow pixels) are used in the computation process. The values of the image function over the intersection regions between the disk and the blue/cyan pixels are then assumed to be 0. This assumption alleviates the need to evaluate the GPCET kernels over these regions and makes the computation of GPCET moments over the yellow-pixel region exact. In addition, this assumption also does not affect the computation steps that will be described in the sequel.

Numerical approximation

Numerical approximation arises naturally when GPCET moments are computed from the approximated lattice of pixels over the unit disk. Let $[i, j]$ denotes one pixel region in the domain of f having a constant value $f[i, j]$, its mapped region in the unit disk is $\left[x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2}\right] \times \left[y_j - \frac{\Delta y}{2}, y_j + \frac{\Delta y}{2}\right]$, where (x_i, y_j) are the coordinates of the mapped region's center. The condition for this mapped region to be labeled with yellow color is that its four corners lie inside the unit disk. Mathematically, the set $\mathcal{C}$ of pixels satisfying this condition is defined as

$$\mathcal{C} = \left\{ [i, j] : \left(x_i - \frac{\Delta x}{2}\right)^2 + \left(y_j - \frac{\Delta y}{2}\right)^2 \leq 1, \left(x_i - \frac{\Delta x}{2}\right)^2 + \left(y_j + \frac{\Delta y}{2}\right)^2 \leq 1, \ldots \right.$$
$$\left. \left(x_i + \frac{\Delta x}{2}\right)^2 + \left(y_j + \frac{\Delta y}{2}\right)^2 \leq 1, \left(x_i + \frac{\Delta x}{2}\right)^2 + \left(y_j - \frac{\Delta y}{2}\right)^2 \leq 1 \right\}. \tag{3.38}$$

The discrete version of the integration in Eq. (3.16) is then a discrete sum indexed by pixels having their mapped regions lying entirely inside the unit disk:

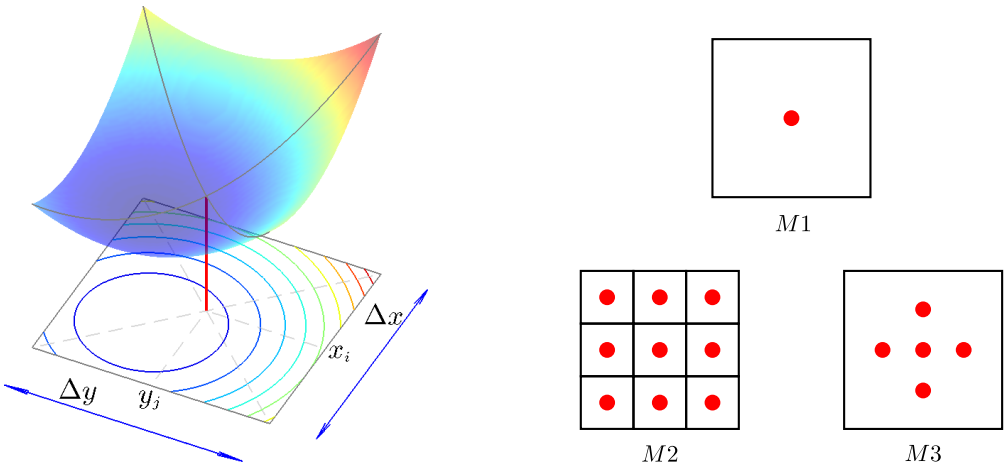
$$H_{nms} = \sum_{[i,j] \in \mathcal{C}} f[i, j]\, h_{nms}[i, j], \tag{3.39}$$

where the factor

$$h_{nms}[i, j] = \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} \int_{y_j - \frac{\Delta y}{2}}^{y_j + \frac{\Delta y}{2}} V^*_{nms}(x, y)\, \mathrm{d}x\mathrm{d}y \tag{3.40}$$

represents the contribution of V^*_{nms} over a region of size $\Delta x \times \Delta y$ representing the pixel $[i, j]$ (see Fig. 3.7a). Since V_{nms} is originally defined by the polar coordinates as in Eq. (3.1), whereas $h_{nms}[i, j]$ is to be evaluated by the Cartesian coordinates, the evaluation of $h_{nms}[i, j]$ in practice usually relies on numerical integration techniques. There exist many such techniques that can be employed for the approximate evaluation of $h_{nms}[i, j]$, of which the simplest is the rectangle formula (scheme $M1$ in Fig. 3.7b):

$$h_{nms}[i, j] \simeq V^*_{nms}(x_i, y_j)\Delta x \Delta y. \tag{3.41}$$



(a) V_{nms} over the mapped region (b) Approximation schemes

Figure 3.7: (a) Illustration of the value of V_{nms} over the mapped region of size $\Delta x \times \Delta y$ in the unit disk representing the pixel $[i, j]$. (b) Some example approximation schemes for numerical integration of 2D functions over a square region: rectangle formula ($M1$), up-sampling then rectangle formula ($M2$), and five-dimensional cubature formula ($M3$). The red dots represent the positions where the value of V_{nms} is sampled.

This approximation assumes that the value of V_{nms} over the mapped region of the pixel $[i, j]$ is constant and equal to its value at the central point (x_i, y_j). The order of approximation error in this case is $O\left((\Delta x \Delta y)^2\right)$, i.e., depending on the area of the mapped region. Therefore, a more accurate approximation could be obtained by "pseudo" up-sampling the pixel $[i, j]$ then using Eq. (3.41) on each up-sampled pixel ($L = 3 \times 3$ up-sampling is shown in the scheme $M2$ in Fig. 3.7b). Other approaches use the L-dimensional cubature formulas [75] defined as

$$h_{nms}[i, j] \simeq \sum_{k=1}^{L} w_k V^*_{nms}(u_k, v_k) \Delta x \Delta y,$$

where $\{(u_k, v_k) : 1 \leq k \leq L\}$ and $\{w_k : 1 \leq k \leq L\}$ are a set of design points that belong to the mapped region representing the pixel $[i, j]$ and a set of the corresponding weights respectively. As an example, a five-dimensional cubature formula (scheme $M3$ in Fig. 3.7b) has the following definition:

$$h_{nms}[i, j] \simeq \frac{1}{3}\left[-V^*_{nms}(x_i, y_j) + V^*_{nms}\left(x_i + \frac{\Delta x}{4}, y_j\right) + V^*_{nms}\left(x_i - \frac{\Delta x}{4}, y_j\right) \ldots \right.$$
$$\left. + V^*_{nms}\left(x_i, y_j + \frac{\Delta x}{4}\right) + V^*_{nms}\left(x_i, y_j - \frac{\Delta x}{4}\right)\right] \Delta x \Delta y.$$

The order of approximation error now reduces to $O\left((\Delta x \Delta y)^{L+1}\right)$ and, obviously, higher accuracy is obtained when a larger value of L is used. Note that the gain in accuracy has to be paid by an increase in complexity, which is not always preferred in real applications. In this work, for simplicity and for the purpose of comparison, the scheme $M1$ in Fig. 3.7b is used as a common ground for the computation of $h_{nms}[i, j]$ in GPCET and comparison methods.

The above approximation error in computing $h_{nms}[i, j]$ is often called *numerical error* in the literature. The impact of this type of error on the computed moments depends heavily on the behavior of V_{nms} over the mapped region representing each pixel and thus varies according to the

actual moment and the orders n, m. For the case of GPCET, the radial and angular kernels that constitute V_{nms} are defined based on complex exponential functions and oscillate within their respective intervals, $[0, 1]$ and $[0, 2\pi)$. This means that the value of V_{nms} will oscillate at a higher frequency as n and/or m increase. Moreover, since the aforementioned approximation schemes are based on the sampled values of V_{nms}, the computed moments are susceptible to information loss if V_{nms} has been under-sampled. This is because, in practice, the number of sampling points is fixed and determined by the size of the input image. For this reason, when a moment of a too high order n or m is considered, a fixed sampling of V_{nms} becomes insufficient.

There actually exist some proposed approaches trying to eliminate numerical error by avoiding direct approximation of $h_{nms}[i, j]$ over each mapped region representing the pixel $[i, j]$. Some stick to the Cartesian space and perform the integration in Eq. (3.40) analytically after converting V_{nms} from being defined by polar coordinates to being defined by Cartesian ones [121, 230]. This is made possible by the relationship between Zernike and geometric kernels [215] or by piecewise polynomial interpolation of V_{nms}. Another approach [236] gets rid of the integration in Eq. (3.40) and computes the moments in polar space by using bicubic interpolation to convert the digital image f from being defined as a piecewise constant function in the Cartesian space to being defined by polar coordinates. In this manner, the accurate computation of moments can be carried out in the polar space. Recently, optimization techniques have been used to improve the orthogonality of the approximated discrete kernels for more accurate image reconstruction [136]. However, these approaches to eliminate numerical error are much more computationally expensive than the above approximation schemes because interpolation makes any recurrence relation between radial kernels invalid and optimization requires iterative evaluations. Moreover, these approaches also introduce new error into the computed moments due to the interpolation or optimization process. For these reasons, in this work, these approaches are not employed to compute moments of harmonic function-based and comparison methods.

3.4.2 Computational complexity

Let $\mathcal{S} = \{(n, m) : n, m \in \mathbb{Z}\}$ be a countable set of orders of GPCET moments and assuming that the rectangle rule is employed to compute $h_{nms}[i, j]$, then the number of kernels to be computed is equal to $|\mathcal{S}|$, the cardinality of $\mathcal{S}$. Direct computation of moments using Eq. (3.39), which requires the computation of $h_{nms}[i, j]$ and then the evaluation of a discrete sum, is excessively time-consuming, especially when $|\mathcal{S}|$ is relatively large and/or the input image has high resolution. Since computation can be divided into three separate stages (radial kernels, angular kernels, and discrete sum), many strategies were proposed trying to reduce the computational complexity of one of these stages and they can be roughly classified into two groups: recursion and symmetry.

- *Recursion*: The radial kernels defined based on Jacobi polynomials often use some factorials in their definitions. Since directly computing these factorials is time consuming, most of the time taken for the computation of moments is due to the computation of radial kernels. For this reason, strategies were proposed for fast computation of Jacobi polynomial-based radial kernels using their recurrence relations as in [169] and references therein. Different forms of relations result in different implementations, each may be suitable for some particular situations. For example, the method in [118] is useful for computing Zernike moments of different orders n and the same repetition m whereas the q-recursive method [45] is more effective in cases where Zernike moments of a fixed order n and different repetitions m are needed.

- *Symmetry*: Symmetrical points of a point P_1 across the y-axis, the origin, the x-axis and

the line $y = x$ in the Cartesian coordinate system have the same radial and related angular coordinates as those of P_1. These relations between the polar coordinates of symmetrical points result in the same radial and related angular kernel values. As a result, geometrical symmetry in the distribution of pixels inside the unit disk has been utilized to reduce the need of computing the radial and angular kernels for all pixels inside the unit disk to pixels in one of the eight sectors [103]. Due to its geometrical nature, this strategy can be used in combination with any existing strategy for fast computation of radial kernels based on recurrence relations to further reduce the computational complexity.

Because the GPCET radial and angular kernels are defined based on complex exponential functions, existing recursive strategies proposed for the computation of Jacobi polynomial-based radial kernels are not applicable. However, since the geometrical symmetry holds for any kernel of the type in Eq. (3.1), it can be employed to compute GPCET moments. The remaining of this subsection will describe in more detail this geometrical symmetry for completeness along with the proposed strategies for fast computation of harmonic function-based radial and angular kernels based on their different forms of recurrence relations. These proposed strategies can again be combined with the symmetry-based strategy for a multiplication of computational gains obtained by the two combining strategies.

In the literature, there does exist another approach for fast computation of ART moments [122]. In this approach, piecewise polynomial interpolation has been used to approximate ART kernels by geometric kernels, which allow fast computation by using vertical lines and the discrete Green theorem [176], reducing the complexity from $O(N^2)$ to $O(N)$ for an image of size $N \times N$. Certainly, the proposed approach can also be employed for other types of moments defined over the unit disk. However, that approach can only be applied for binary images since it is based on the concept of left-hand and right-hand boundaries. This requisite for the existence of boundaries is the main hindrance that limits the application of methods based on the discrete Green theorem to general pattern images.

Geometrical symmetry

Assuming that P_1 is a point in the Cartesian coordinate system, its symmetrical points $P_2 - P_4$ and $Q_1 - Q_4$ are illustrated in Fig. 3.8. The Cartesian and polar coordinates of these symmetrical points, in relation with those of P_1, are easily obtained and given in Table 3.3. It is evident that the distances from the origin to all points are the same, meaning that these points have the same radial coordinate r and hence the same radial kernel value $R_{ns}(r)$. The angular coordinates of these symmetrical points can be expressed via θ, the angular coordinate of P_1, leading to the possibility of expressing the angular kernel values of these symmetrical points via that of P_1 by resorting to the following identities:

$$\mathrm{e}^{im\frac{\pi}{2}} = \begin{cases} 1, & m = 4k \\ i, & m = 4k+1 \\ -1, & m = 4k+2 \\ -i, & m = 4k+3 \end{cases} \qquad k \in \mathbb{Z},$$

$$\mathrm{e}^{-im\theta} = A_m^*(\theta).$$

Using the angular coordinates of the points $P_1 - P_4$ and $Q_1 - Q_4$ given in Table 3.3, it is thus not difficult to have the following relations between the angular kernels of symmetrical points

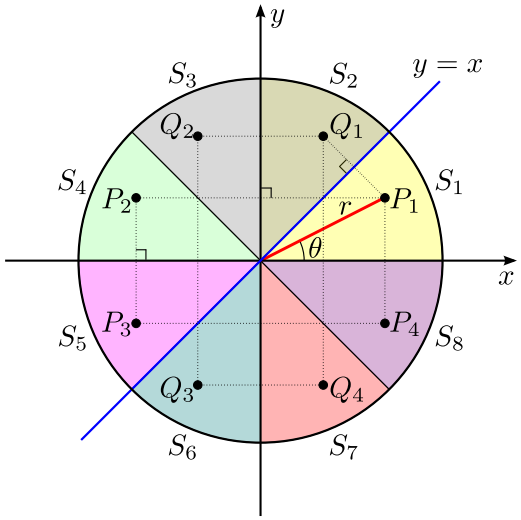


Figure 3.8: Symmetrical points $P_2 - P_4$ of a point P_1 inside the unit disk across the y-axis, the origin, and the x-axis respectively. Similarly, $Q_2 - Q_4$ are symmetrical points of Q_1, the symmetrical point of P_1 across the line $y = x$. These points have the same radial coordinate as that of P_1 whereas their angular coordinates are related to that of P_1 by relations given in the last column of Table 3.3.

and that of the original point P_1:

$$A_m(P_1) = \mathrm{e}^{im\theta}$$

$$A_m(P_2) = \mathrm{e}^{im(\pi-\theta)} = \mathrm{e}^{im\pi}\,\mathrm{e}^{-im\theta} = \begin{cases} A_m^*(P_1), & m = 2k \\ -A_m^*(P_1), & m = 2k+1 \end{cases}$$

$$A_m(P_3) = \mathrm{e}^{im(\pi+\theta)} = \mathrm{e}^{im\pi}\,\mathrm{e}^{im\theta} = \begin{cases} A_m(P_1), & m = 2k \\ -A_m(P_1), & m = 2k+1 \end{cases}$$

$$A_m(P_4) = \mathrm{e}^{-im\theta} = A_m^*(P_1)$$

$$A_m(Q_1) = \mathrm{e}^{im\left(\frac{\pi}{2}-\theta\right)} = \mathrm{e}^{im\frac{\pi}{2}}\,\mathrm{e}^{-im\theta} = \begin{cases} A_m^*(P_1), & m = 4k \\ iA_m^*(P_1), & m = 4k+1 \\ -A_m^*(P_1), & m = 4k+2 \\ -iA_m^*(P_1), & m = 4k+3 \end{cases}$$

$$A_m(Q_2) = \mathrm{e}^{im\left(\frac{\pi}{2}+\theta\right)} = \mathrm{e}^{im\pi}\,\mathrm{e}^{-im\left(\frac{\pi}{2}-\theta\right)} = \begin{cases} A_m^*(Q_1), & m = 2k \\ -A_m^*(Q_1), & m = 2k+1 \end{cases}$$

$$A_m(Q_3) = \mathrm{e}^{im\left(-\frac{\pi}{2}-\theta\right)} = \mathrm{e}^{-im\pi}\,\mathrm{e}^{im\left(\frac{\pi}{2}-\theta\right)} = \begin{cases} A_m(Q_1), & m = 2k \\ -A_m(Q_1), & m = 2k+1 \end{cases}$$

$$A_m(Q_4) = \mathrm{e}^{im\left(-\frac{\pi}{2}+\theta\right)} = \mathrm{e}^{-im\left(\frac{\pi}{2}-\theta\right)} = A_m^*(Q_1).$$

The significance of these identities is that once the angular kernel value of when of these eight points is available, that of the remaining seven points can be obtained easily by using complex conjugation, additive inverse, and multiplication by i. This observation has been exploited to restrict the computation of angular kernels to only one of the eight sectors $S_1 - S_8$, instead of

Table 3.3: Cartesian and polar coordinates of the symmetrical points of a point P_1. The coordinates of symmetrical points are expressed via those of P_1: Cartesian (x, y) and polar (r, θ).

Symmetrical point	Symmetrical axis	Cartesian coordinates	Radial coordinate	Angular coordinate
P_1		(x, y)	r	θ
P_2	y-axis	$(-x, y)$	r	$\pi - \theta$
P_3	origin	$(-x, -y)$	r	$\pi + \theta$
P_4	x-axis	$(x, -y)$	r	$-\theta$
Q_1	$y = x$	(y, x)	r	$\frac{\pi}{2} - \theta$
Q_2	$y = x$, y-axis	$(-y, x)$	r	$\frac{\pi}{2} + \theta$
Q_3	$y = x$, origin	$(-y, -x)$	r	$-\frac{\pi}{2} - \theta$
Q_4	$y = x$, x-axis	$(y, -x)$	r	$-\frac{\pi}{2} + \theta$

the whole unit disk region. Without loss of generality, assuming that S_1 $(0 \leq y \leq x)$ is going to be used as the computing sector, Euler's formula allows a rewrite of the angular kernels of P_1:

$$A_m(\theta) = \mathrm{e}^{im\theta} = \cos(m\theta) + i \sin(m\theta).$$

Eq. (3.39) can then be rewritten as

$$\begin{aligned} H_{nms} &\simeq \sum_{[i,j]\in\mathcal{C}} f[i,j] V^*_{nms}(x_i, y_j) \Delta x \Delta y \\ &= \sum_{[i,j]\in\mathcal{C}} f[i,j] R^*_{ns}(r_{ij}) A^*_m(\theta_{ij}) \Delta x \Delta y \\ &= \sum_{0\leq y_j \leq x_i, [i,j]\in\mathcal{C}} R^*_{ns}(r_{ij}) \left[A^r_m(\theta_{ij}) - i A^i_m(\theta_{ij}) \right] \Delta x \Delta y \end{aligned}$$

where (r_{ij}, θ_{ij}) are the polar coordinates of the mapped region's center (x_i, y_j); A^r_m and A^i_m have the following definitions:

$$A^r_m(\theta) = \begin{cases} \left[f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8\right] \cos(m\theta), & m = 4k \\ \left[f_1 - f_4 - f_5 + f_8\right] \cos(m\theta) + \left[f_2 - f_3 - f_6 + f_7\right] \sin(m\theta), & m = 4k+1 \\ \left[f_1 - f_2 - f_3 + f_4 + f_5 - f_6 - f_7 + f_8\right] \cos(m\theta), & m = 4k+2 \\ \left[f_1 - f_4 - f_5 + f_8\right] \cos(m\theta) + \left[-f_2 + f_3 + f_6 - f_7\right] \sin(m\theta), & m = 4k+3 \end{cases} \tag{3.42}$$

$$A^i_m(\theta) = \begin{cases} \left[f_1 - f_2 + f_3 - f_4 + f_5 - f_6 + f_7 - f_8\right] \sin(m\theta), & m = 4k \\ \left[f_1 + f_4 - f_5 - f_8\right] \sin(m\theta) + \left[f_2 + f_3 - f_6 - f_7\right] \cos(m\theta), & m = 4k+1 \\ \left[f_1 + f_2 - f_3 - f_4 + f_5 + f_6 - f_7 - f_8\right] \sin(m\theta), & m = 4k+2 \\ \left[f_1 + f_4 - f_5 - f_8\right] \sin(m\theta) + \left[-f_2 - f_3 + f_6 + f_7\right] \cos(m\theta), & m = 4k+3 \end{cases} \tag{3.43}$$

with f_1 is the value of the image function at the point (r, θ) in sector S_1 and f_k $(k = 2, \ldots, 8)$ are those of symmetrical points in the remaining seven sectors. It is clear that if the multiplicative factors of trigonometric functions cosine and sine in Eqs. (3.42) and (3.43) are pre-computed

and stored, then evaluating $\left[A^r_m(\theta_{ij}) - iA^i_m(\theta_{ij})\right]$ and $f[i,j]A^*_m(\theta_{ij})$ requires almost the same computing resource. In other words, the symmetry-based strategy leads to a reduction in computational complexity by approximately $\frac{1}{8}$. The efficiency of this strategy for computing unit disk-based moments has been verified by experiments in some previous works [103]. A further extension of this strategy for 3D patterns was also proposed recently in [238].

Recursive computation of complex exponential functions

In GPCET, both the radial and angular kernels are defined based on complex exponential functions. Direct computation of complex exponential functions is time-consuming and often constitutes a dominant part of the computation of GPCET moments due to its $O(\log^2 n)$ complexity, where n refers to the number of precision digits at which the function is to be evaluated [25]. The overall complexity may become excessively high when

- a large number of moments is needed, or
- the image has high resolution, or
- a high-precision computation is required.

Since these requirements are common in real applications, the existence of strategies for fast computation of kernels is vital for the applicability of GPCET. Fortunately, due to the following recursive definition of exponentiation:

$$\begin{aligned} \text{base case:} \quad & \mathrm{e}^{i0\alpha} = 1, \\ \text{inductive clause:} \quad & \mathrm{e}^{ik\alpha} = \mathrm{e}^{i(k-1)\alpha}\,\mathrm{e}^{i\alpha}, \quad k, \alpha \in \mathbb{Z}, \end{aligned}$$

the complex exponential functions in the definitions of GPCET radial and angular kernels can be computed recursively as

$$\begin{aligned} \mathrm{e}^{i2\pi n r^s} &= \mathrm{e}^{i2\pi(n-1)r^s}\mathrm{e}^{i2\pi r^s}, \\ \mathrm{e}^{im\theta} &= \mathrm{e}^{i(m-1)\theta}\,\mathrm{e}^{i\theta}, \end{aligned} \tag{3.44}$$

using the base cases $\mathrm{e}^{i2\pi 0 r^s} = 1$ and $\mathrm{e}^{i0\theta} = 1$ respectively.

Assuming that $\left\{\sqrt{\frac{sr^{s-2}}{2\pi}}, \mathrm{e}^{i2\pi r^s}, \mathrm{e}^{i\theta}\right\}$ has been pre-computed and stored for polar coordinates (r,θ) of all the mapped pixel regions' centers, the following recurrence relations of R_{ns} and A_m:

$$R_{ns}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}\,\mathrm{e}^{i2\pi n r^s} = R_{(n-1)s}(r)\,\mathrm{e}^{i2\pi r^s}, \tag{3.45}$$

$$A_m(\theta) = \mathrm{e}^{im\theta} = A_{m-1}(\theta)\,\mathrm{e}^{i\theta}, \tag{3.46}$$

lead to their recursive computation with the base cases $R_{0s}(r) = \sqrt{\frac{sr^{s-2}}{2\pi}}$ and $A_0(\theta) = 1$ respectively. Obviously, computing R_{ns} from $R_{(n-1)s}$ and A_m from A_{m-1} each requires only one multiplication, which is very fast when compared to exponentiation, leading to fast computation of V_{nms}. Moreover, these forms of recurrence relations are simpler than those that were discovered for Jacobi polynomial-based radial kernels [45]. By using Eq. (3.45), only one recursive computational thread is sufficient to reach every GPCET radial kernels, whereas many threads would be required to cover all Jacobi polynomial-based radial kernels. The computation flows of GPCET radial kernels R_{ns} and angular kernels A_m are illustrated in Fig. 3.9. It is evident that the method

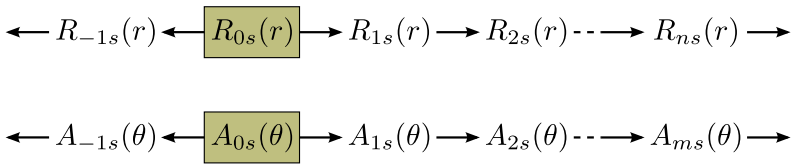


Figure 3.9: Computation of GPCET radial kernels R_{ns} and angular kernels A_m based on recursive computation of complex exponential functions in Eqs. (3.45) and (3.46) respectively. Computing R_{ns} from $R_{(n-1)s}$ and A_m from A_{m-1} each requires only one multiplication, leading to fast computation of V_{nms}.

proposed here is much faster than the one mentioned in [239] where exponentiation is required to compute complex exponential radial and angular kernels.

The above proposed recursive computation of GPCET radial and angular kernels could be employed for extremely fast computation of GPCET moments when the order set $\mathcal{S}$ composes a square region in $\mathbb{Z}^2$ and takes the origin as its center:

$$\mathcal{S} = \{(n,m) : n,m \in \mathbb{Z}^2, |n|, |m| \leq K\},$$

where K is a positive integer. By using the computational flow depicted in Fig. 3.10a, computing a GPCET moment thus requires only three multiplications, two for getting V_{nms} and one for multiplying V_{nms} by f, followed by a discrete sum of the obtained results over all the pixels $[i,j] \in \mathcal{C}$ defined in Eq. (3.39).

To further boost the computation speed, instead of letting the computational flow to visit all $(n,m) \in \mathcal{S}$ in the four quadrants in the Cartesian space, it is sufficient to visit only $(n,m) \in \mathcal{S}$ in one quadrant $(n,m > 0)$ as illustrated in Fig. 3.10b. This is possible due to the following relations:

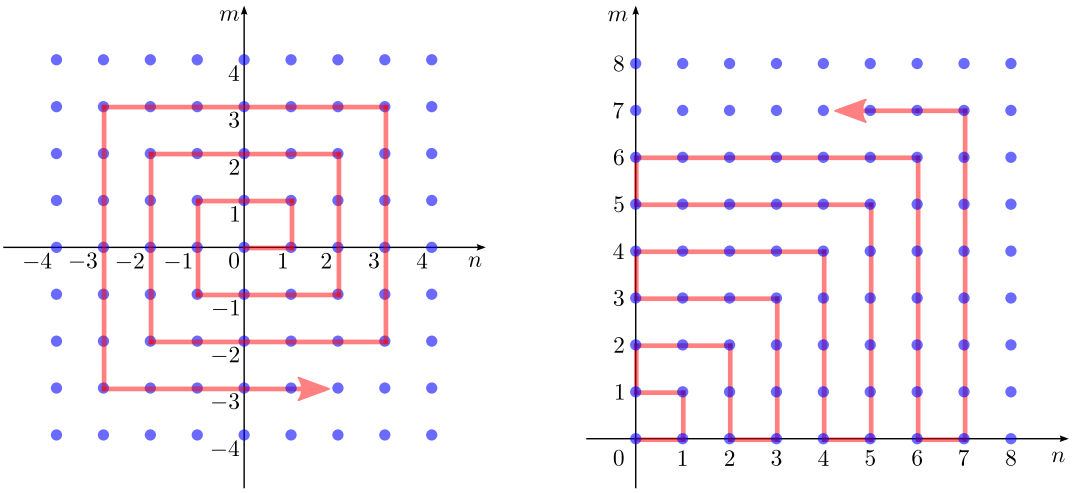
$$\begin{aligned} R_{-ns}(r) &= R^*_{ns}(r), \\ A_{-m}(\theta) &= A^*_m(\theta). \end{aligned}$$

Thus, whenever R_{ns} and A_m are available, computing the four related GPCET kernels, for which eight multiplications should be needed if the computational flow in Fig. 3.10a is used, requires only two multiplications and three conjugations:

$$\begin{aligned} V_{nms}(x,y) &= R_{ns}(r)A_m(\theta), \\ V_{-nms}(x,y) &= R^*_{ns}(r)A_m(\theta), \\ V_{-n-ms}(x,y) &= V^*_{nms}(x,y), \\ V_{n-ms}(x,y) &= V^*_{-nms}(x,y), \end{aligned}$$

leading to a $\frac{8}{3}$-time reduction in the number of multiplications.

In some situations where there is enough memory to store all the radial kernels R_{ns} $(|n| \leq K)$ and angular kernels A_m $(|m| \leq K)$, then pre-computing their values may lead to a further reduction in computational complexity by using the computational flow in Fig. 3.11. It is clear that the total number of multiplications required to update the values of R_n or A_m using Eqs. (3.45) or (3.46) for all the kernels V_{nms} $(0 \leq n,m \leq K)$ is $(K+1)^2 - 1$ whereas pre-computation needs only $2K$ multiplications. In addition, besides the memory requirement, this strategy has another disadvantage concerning the data accessing time. The pre-computed values of all R_{ns} (or A_m) have to be stored in a matrix $\mathbf{R}_s$ (or $\mathbf{A}$) indexed by r and n (or θ and m). When the kernel



(a) Four quadrants (b) One quadrant

Figure 3.10: Computation flows of GPCET kernels starting from the order $(0, 0)$. At each step, one of the orders n and m changes its value by adding 1 or -1, depending on the current direction of the flow. Computing each kernel V_{nms} thus requires two multiplications, one for computing R_n or A_m using Eqs. (3.45) or (3.46), depending on whether the value of n or m has just been changed, and the other for multiplying R_n by A_m to get V_{nms}.

V_{nms} needs to be computed, the values in the nth column of $mathbf R_s$ and mth column of $\mathbf{A}$ are retrieved and these added steps may slow down the overall computation process. However, since this strategy can be applied to all unit-disk based moments, it will not be evaluated in the experimental section.

In real situations, the less demanding computation of GPCET kernels leads to the possibility of increasing the number of GPCET moments without changing the system throughput. This increase is equivalent to an increase in the number of features or, more importantly, an increase in the number of pattern classes the system can discriminate. Another by-product of this less demanding computation is the reduction in storage space requirement. In methods based on orthogonal polynomials, to avoid the repetitive expensive computation of kernels, the common practice is to pre-compute and store them. For example, if K moments are needed then the K corresponding kernels need to be stored. In the case of GPCET, there merely needs to store the set $\left\{\sqrt{\frac{sr^{s-2}}{2\pi}}, \mathrm{e}^{i2\pi r^s}, \mathrm{e}^{i\theta}\right\}$ for polar coordinates (r, θ) of all the mapped regions' centers, regardless the required number of moments K since the kernel of any order can be computed without much cumbersome. This, of course, leads to a reduction in storage space requirement by $\frac{2K}{3}$.

Computation of trigonometric functions

Unlike those of GPCET, the radial kernels of GRHFM, GPCT, and GPST are defined based on trigonometric cosine and sine functions. It is well-known that cosine and sine functions are equivalent to complex exponential function in terms of computation complexity due to the following identities:

$$\cos(\pi n r^s) = \mathrm{Re}\{\mathrm{e}^{i\pi n r^s}\} = \frac{\mathrm{e}^{i\pi n r^s} + \mathrm{e}^{-i\pi n r^s}}{2},$$

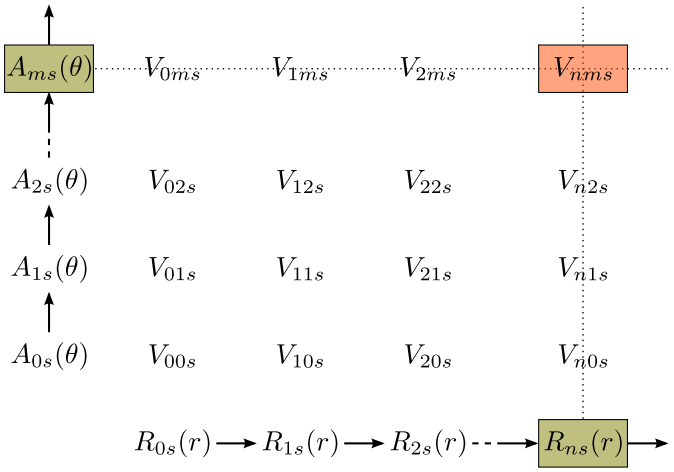


Figure 3.11: Computation of GPCET kernels V_{nms} from the pre-computed and stored values of the radial kernels R_{ns} and angular kernels A_m $(0 \leq n, m \leq K)$. Pre-computation of R_{ns} and A_m needs only $2K$ multiplications, whereas their step-by-step updates using Eqs. (3.45) or (3.46) require a total of $(K+1)^2 - 1$ multiplications.

$$\sin(\pi n r^s) = \mathrm{Im}\{\mathrm{e}^{i\pi n r^s}\} = \frac{\mathrm{e}^{i\pi n r^s} - \mathrm{e}^{-i\pi n r^s}}{2i}.$$

By using the above identities, the formulation of recursive computation of radial and angular kernels of GPCET can also be employed for fast computation of radial kernels of GRHFM, GPCT, and GPST. In other words, implementations of GRHFM, GPCT, and GPST can resort to that of GPCET for low computational complexity. In this way, all the aforementioned computational gains claimed for GPCET moments by using recursive computation are also valid for GRHFM, GPCT, and GPST moments.

Apart from relying on complex exponential functions, cosine and sine functions could also be fast computed by using the following recurrence relations:

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{n+1}(x) &= 2x\,T_n(x) - T_{n-1}(x), \quad \text{for } n \geq 1, \end{aligned} \tag{3.47}$$

and

$$\begin{aligned} U_0(x) &= 1 \\ U_1(x) &= 2x \\ U_{n+1}(x) &= 2x\,U_n(x) - U_{n-1}(x), \quad \text{for } n \geq 1 \end{aligned} \tag{3.48}$$

with

$$\begin{aligned} T_n(\cos(\pi r^s)) &= \cos(\pi n r^s), \\ U_n(\cos(\pi r^s)) &= \frac{\sin(\pi(n+1)r^s)}{\sin(\pi r^s)} \Rightarrow \sin(\pi n r^s) = U_{n-1}(\cos(\pi r^s))\sin(\pi r^s), \end{aligned}$$

where T_n and U_n are the Chebyshev polynomials of the first and second kinds respectively [120]. Accordingly, the computations of $\cos(\pi n r^s)$ and $\sin(\pi n r^s)$ can also be carried out recursively

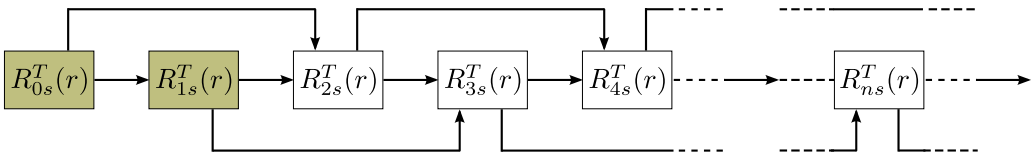


Figure 3.12: Computation of GRHFM, GPCT, and GPST radial kernels based on recursive computation of cosine and sine functions by means of Chebyshev polynomials in Eqs. (3.47) and (3.48) respectively. Computing $\cos(\pi n r^s)$ and $\sin(\pi n r^s)$ each requires only one multiplication and one substraction, leading to fast computation of V^H_{nms}, V^C_{nms}, and V^S_{nms}.

with the base cases $\{1, \cos(\pi r^s)\}$ for Eq. (3.47) and $\{1, 2\cos(\pi r^s)\}$ for Eq. (3.48), providing that $2\cos(\pi r^s)$ and $\sin(\pi r^s)$ are available throughout the computation processes. The computation flows of GRHFM, GPCT, and GPST radial kernels are illustrated in Fig. 3.12. It is clear that computing $\cos(\pi n r^s)$ and $\sin(\pi n r^s)$ each requires only one multiplication and one substraction, which is almost equivalent to one multiplication required to compute $e^{i2\pi n r^s}$ as in Eq. (3.44).

Before closing this subsection on computational complexity, it should be noted here that the above proposed methods for recursive and fast computation of complex exponential and trigonometric functions can also be employed for ART [22] and GFD [243] since their radial kernels are also defined based on harmonic functions. In addition, since the methods proposed in this subsection are orthogonal to the one based on geometrical symmetry, their combination will, in theory, multiply the computational gains obtained by the two combining methods. This observation will have experimental evidence in Section 3.5.

3.4.3 Numerical stability

Accuracy is another concern when moments are computed numerically in digital systems. Since moments are originally defined by using a double continuous integral over a unit disk domain, the discrete approximation of the integral will incur error in the computation. Another type of error comes from the digital nature of computing systems where numbers can only be correctly represented in a certain range and to a certain precision. These two types of errors will be discussed in detail in this subsection.

Approximation error

The two types of discrete approximations discussed in Subsection 3.4.1 naturally correspond to two types of approximation errors [135]: geometric error and numerical error. Geometric error occurs when the domain of integration does not exactly cover the unit disk due to the geometric difference between circular and rectangular domains. This type of error, however, could be "avoided" if only the pixels that lie entirely inside the unit disk are used and the values of the image function over the remaining regions of the unit disk are assumed to be 0. This means that the computed moments only reflect the values of the yellow pixels in Fig. 3.6. Since this strategy will be used to compute the moments of harmonic function-based and comparison methods, geometric error hence does not exist.

Numerical error arises when $h_{nms}[i, j]$ in Eq. (3.40) is computed by a numerical integration technique. Because the numerically computed value of $h_{nms}[i, j]$ is just an approximation to its analytical value, this type of error cannot be avoided in any way if moments are computed by numerical approximation. The magnitude of this type of error, however, could be reduced if only a highly accurate numerical integration technique is employed (e.g., "pseudo" sub-sampling or

cubature). Due to Eqs. (3.36) and (3.37), it is clear that the impact of numerical error on the computed moments depends on the image size: a smaller-sized image has a greater error, and vice versa. The impact of numerical error on all unit disk-based moments will be demonstrated experimentally by means of reconstruction error in the experimental section.

Representation error

In today's numerical computing systems, a real number is in general approximately represented in floating-point format in order to allow reasonable storage requirement and relatively quick calculations. The typical number that can be represented exactly is of the form

$$\text{Significand} \times \text{base}^{\text{exponent}},$$

where *significand* denotes a signed digit string of a given length in a given *base* and *exponent* is a signed integer which modifies the magnitude of the number. Since computing systems are binary in nature, floating-point numbers are normalized for representation as

$$\pm(1+f) \times 2^e,$$

where f is the fraction or mantissa ($0 \leq f < 1$) and e is the exponent. In 32-bit systems, under the IEEE 754 standard, *double* precision floating numbers use two storage locations, or 64 bits, to store the value of f, e, and the number's sign: 52 bits for f, 11 bits for $e+1023$, and 1 bit for the sign. A double number v thus can only be represented with the relative accuracy of one-half the *machine epsilon*, or $\frac{1}{2} \times eps = \frac{1}{2} \times 2^{-52} \simeq 1.1102 \times 10^{-16}$. This means that, when represented in the ordinary decimal numeral system, only the first 15 left-most digits of v are significant. In addition, due to the limited range of e, the absolute values of double numbers are also limited in the range $2^{-1022} \div (2 - eps)\, 2^{1023}$, or approximately $2.2251 \times 10^{-308} \div 1.7977 \times 10^{308}$. This finite set of double numbers with finite precision leads to the phenomena of *underflow*, *overflow*, and *roundoff* in computing systems. Due to their nature, it has been known in the literature that Jacobi polynomial-based methods suffer from all three types of errors [170] as pointed out below:

- Underflow error occurs when a computed quantity has a value under the range of its data type. Jacobi polynomial-based methods has this type of error due to the use of powers of r in their definition. At the radial coordinate close to zero $r = 0.001$, $r^{102} = 1.0000 \times 10^{-306}$ and $r^{103} = 1.0000 \times 10^{-309}$ then any computation that involves r to the power greater than 102 will cause underflow error. Obviously, this type of error depends on the size of images: a larger-sized image starts to have this error at a smaller order n. As an example, for an input image of size 1024×1024, the smallest value of r in the computation is $\frac{1}{2} \times \frac{1}{1024} = 2^{-11} = 4.8828 \times 10^{-4}$, then underflow error starts to occur at $n = 93$ onwards for all Jacobi polynomial-based methods.

- Overflow error occurs when a computed quantity has a value above the range of its data type. Jacobi polynomial-based methods has this type of error due to the use of factorial in their definition. Since $170! = 7.2574 \times 10^{306}$ and $171! = 1.2410 \times 10^{309}$, any computation that involves factorial of a number greater than 170 will cause overflow error. From the definitions of radial polynomials, it is straightforward to verify that ZM, PZM, OFMM, CHFM, and PJFM start to have this type of error at $n = 171, 85, 85, 171,$ and 84 onwards respectively.

- Roundoff error is the difference between the approximation of a number and its exact (i.e., correct) value. Because of the finite precision in computing systems, this type of

Table 3.4: The radial orders of Jacobi polynomial-based methods from which underflow, overflow, and roundoff errors start to occur in 32-bit computing systems. These methods have the same order for underflow error and different orders for overflow and roundoff errors.

Error type	ZM	PZM	OFMM	CHFM	PJFM
Underflow[3]	93	93	93	93	93
Overflow	171	85	85	171	84
Roundoff[4]	46	23	23	79	21

error occurs in almost all numerical computation steps. However, different from the other methods, Jacobi polynomial-based methods face the problem of excessively large coefficients in the definitions of radial polynomials. These coefficients are sometimes larger than 2^{52} and thus, for the commonly 15-digit precision, computing radial kernels produces error of the order of unity or larger. It is not difficult to determine the orders from which Jacobi polynomial-based methods have this type of error; they are $n = 46, 23, 23, 79,$ and 21 for ZM, PZM, OFMM, CHFM, and PJFM respectively.

For all Jacobi polynomial-based methods, the starting orders for each type of error are summarized in Table 3.4. Due to their distinct definitions, different methods have different orders for overflow and roundoff errors. For underflow error, Jacobi polynomial-based methods have the same order because of the same polynomial order in their radial kernels of the same order. It can be seen that, among these three types of errors, roundoff error occurs at the smallest order for all Jacobi polynomial-based methods. As a result, the roundoff error is the main concern in moment computation.

From the above definitions of three types of representation errors, it appears that eigenfunction-based and harmonic function-based methods do not suffer from underflow and overflow errors, they do have roundoff error because of the nature of numerical computing systems. However, the impact of roundoff error on their computed moments is not as severe as that on the computed moments of Jacobi polynomial-based methods because of the non-existence of large-valued coefficients in their radial kernel definitions. As will be shown experimentally in the next section, this impact causes serious problems in Jacobi polynomial-based methods. Nevertheless, any of the aforementioned error types is undesirable since it alters the computed moments, compromises the orthogonality of moments/kernels, and finally corrupts the application's performance.

3.5 Experimental results

The effectiveness of the proposed harmonic function-based moments will be demonstrated in comparison with existing moments of the same nature, i.e., unit disk-based orthogonal moments, through three types of experiments: computational complexity, representation capability, and discrimination power. The first one evaluates how fast the computation of harmonic function-based kernels/moments is, using the proposed recursive schemes in combination with and without the geometrical symmetry-based method. The second type of experiments deals with the capability of harmonic function-based moments in representing image functions and this is done via image

[3]For an input image of size 1024×1024.
[4]Roundoff error of the order of unity.

Table 3.5: The constraints on the moment orders (n, m) of comparison methods for a fixed value of K in the experiments on computational complexity. All (n, m) satisfying these conditions are used in the computation and the elapsed times are averaged out over all feasible orders.

Method	Order range
ZM	$\lvert m\rvert \leq n \leq K$, $n - \lvert m\rvert =$ even
PZM	$\lvert m\rvert \leq n \leq K$
OFMM/CHFM/PJFM	$0 \leq \lvert m\rvert, n \leq K$
FBM/BFM/DHC	$0 \leq \lvert m\rvert, n \leq K$
GPCET	$\lvert m\rvert, \lvert n\rvert \leq K$
GRHFM	$\lvert m\rvert \leq K$, $0 \leq n \leq 2K$
GPCT	$0 \leq \lvert m\rvert, n \leq K$
GPST	$\lvert m\rvert \leq K$, $1 \leq n \leq K$

reconstruction. The third type of experiments is on the applicability of harmonic function-based moments in rotation-invariant pattern recognition problems at different levels of noise.

3.5.1 Computational complexity

The computational complexity is evaluated in terms of the elapsed time taken to compute the kernels/moments of comparison methods from an image of size 128×128. For this image, the incircle in Fig. 3.6a contains 12596 yellow pixels. Experiments are performed on a PC with a 2.33GHz CPU, 4GB RAM running Linux kernel 2.6.38; MATLAB version 7.7 (R2008b) is used as the programming environment. Let K be some integer constant, all the kernels/moments of orders (n, m) of each method that satisfy the conditions in Table 3.5 are computed and the averaged elapsed time over all feasible orders is taken as the kernel/moment computation time at that value of K. The value of K is varied in the range $0 \leq K \leq 20$ in all experiments on computational complexity in order to study the trends in the dependance of kernel/moment computation time on the maximal kernel/moment order K. In addition, for more reliable results, all the running times indicated in this subsection are averaged over 100 trials.

Direct computation: Fig. 3.13 provides the computation times per kernel in milliseconds of all unit disk-based orthogonal moments. The kernels are computed using their corresponding definitions, no recursive strategy is used. It is observed from the figure that

- Jacobi polynomial-based methods (ZM, PZM, OFMM, CHFM, PJFM) have kernel computation times that increase almost linearly with the increase in K, meaning that longer times are needed to compute kernels of higher orders. This is because of the evaluation of factorials of larger integers and of the computation of more additive terms in the final summations. Among these methods, OFMM and PJFM have the highest and similar complexity while ZM has the lowest. This relative complexity ranking of these methods is consistent with the ranking in the number of multiplications required to compute their radial kernels.

- Eigenfunction-based methods (FBM, BFM, DHC) require the longest times to compute their kernels over comparison methods. Among these methods, FBM and DHC have the same complexity whereas that of BFM is slightly less. The reasons for these observations

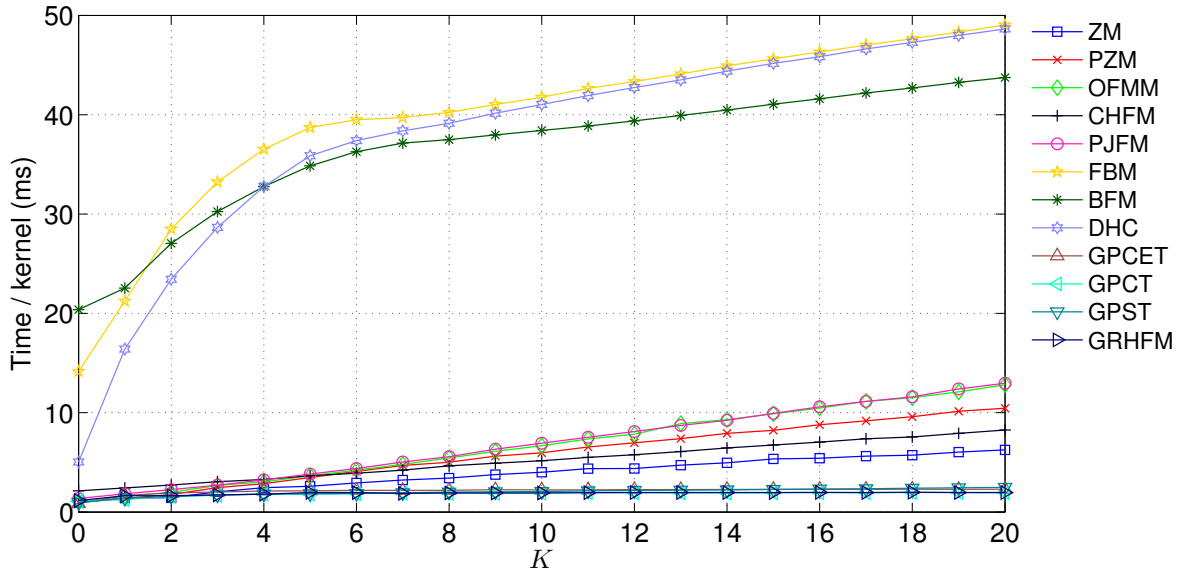


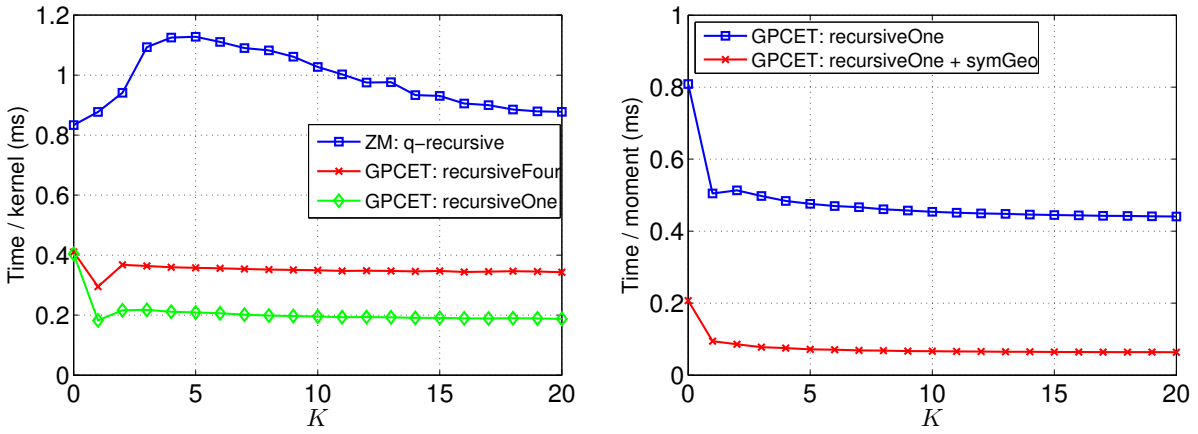
Figure 3.13: Kernel computation times of comparison methods by direct computation at different values of K. For each method and at a specific value of K, the kernel orders used in the computation should satisfy the conditions described in Table 3.5 and the averaged elapsed time over all feasible orders is taken as the kernel computation time.

are threefold. Firstly, these methods initially need to find the zeros of (or the derivative of) Bessel functions in order to define their radial kernels. Secondly, there exists no approximation that allows fast computation for Bessel functions. And finally, FBM and DHC use Bessel functions of different orders while BFM only uses a fixed-order Bessel function. In these experiments, the evaluation of Bessel functions is facilitated by the MATLAB built-in function `besselj`.

- Harmonic function-based methods (GPCET, GRHFM, GPCT, GPST) each requires an almost-constant time to compute its kernels of different orders. This is because a change in the kernel orders corresponds only to a change in the input to the complex exponential function (GPCET) or cosine/sine functions (GRHFM, GPCT, GPST) and, as a result, does not affect the kernel computation time. Moreover, the kernel computation times of these methods are nearly the same since exponential, cosine, and sine functions are equivalent in terms of computational complexity [25].

From the above observations, it can be concluded that the simple, resembling, and relating definitions of harmonic function-based kernels have resulted in an almost-constant kernel computation time, regardless of the maximal kernel order K. This makes a strong contrast with Jacobi polynomial-based and eigenfunction-based methods where a higher kernel order means a longer kernel computation time. As a consequence, harmonic function-based methods should be the preferred methods in terms of computational complexity when kernel/moments of high orders are needed.

Recursive computation: The proposed recursive strategies for fast computation of GPCET kernels are also evaluated and compared with those for fast computation of ZM kernels. The reason for comparing only to ZM is twofold: the lack of benchmarks on fast computation strategies for other methods and the popularity of recursive strategies for ZM in the literature. In this



(a) Without geometrical symmetry

(b) With geometrical symmetry

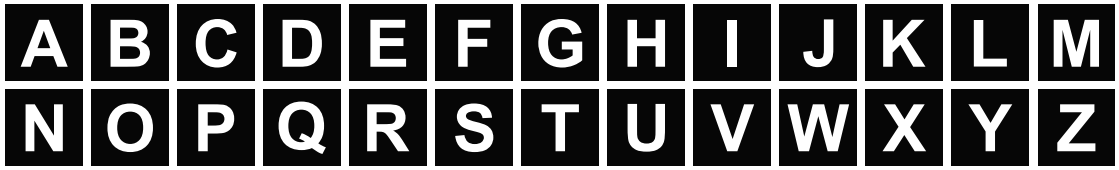
Figure 3.14: (a) ZM's and GPCET's kernel computation times at different values of K using the q-recursive and different recursive strategies for the computation of complex exponential functions described in Subsection 3.4.2. (b) GPCET's moment computation times at different values of K without and with geometrical symmetry.

comparison, ZM kernels are computed using the current state-of-the-art q-recursive strategy [45]. The comparison results are given in Fig. 3.14a where the legends *recursiveFour* and *recursiveOne* denote recursive computation of GPCET kernels using the computational flows in Figs. 3.10a and 3.10b respectively. It is observed that:

- ZM kernel computation time by q-recursive increases in the range $K = 0 \rightarrow 5$ and gradually decreases when $K > 5$. This is because the q-recursive strategy, which requires the pre-computation of the radial kernels $R_{nn}(r)$ and $R_{nn-2}(r)$ for each order n, is applicable only when $K \geq 4$ and is profitable when $K \geq 6$. Moreover, as n increases, more radial kernels are to be computed by recursion and thus the proportion of directly computed radial kernels decreases. This finally leads to a decrease in the averaged computation time of radial kernels as n increases.

- Using *recursiveFour* and *recursiveOne* to compute GPCET kernels leads to an almost-constant computation time regardless of the maximal kernel order K and *recursiveOne* is almost two-time faster than *recursiveFour*. The only exception is at $K = 1$ where the computation time suddenly drops. This is because MATLAB optimizes by simply copying the pre-computed values of $e^{i\theta}$ into $A_1(\theta)$ since $A_0(\theta) = 1$, instead of the more complex multiplication. A constant computation time is due to the fact that the recurrence relations in Eqs. (3.45) and (3.46) do not depend on the kernel orders and there is no need for the pre-computation of GPCET radial kernel of any order as in the ZM's q-recursive strategy. The "purely" recursive computation of radial and angular kernels is a distinct characteristic of harmonic function-based methods.

Taking *recursiveOne* as the selected strategy for recursive fast computation of GPCET kernels, recursive computation of GPCET kernels by *recursiveOne* is, on average, approximately 10-time faster than direct computation of GPCET kernels and five-time faster than recursive computation of ZM kernels by q-recursive. Furthermore, due to the equivalence in computational complexity of harmonic function-based kernels/moments as demonstrated in Subsection 3.4.2, harmonic



(a) Vector character images

(b) 16×16

(c) 32×32

(d) 64×64

(e) 128×128

(f) 256×256

(g) 512×512

Figure 3.15: (a) The vector character images used to generate the six character datasets for the reconstruction experiments by sampling them to have the sizes of 16×16, 32×32, 64×64, 128×128, 256×256, 512×512, and then quantizing them to bilevels. (b)–(g) Some sampled and quantized images from the six datasets.

function-based methods are promising replacements of ZM in image analysis applications where low computational complexity is an important method-selection criteria.

In computing GPCET moments, *recursiveOne* could be combined with the strategy based on geometrical symmetry [103] to further reduce the moment computation time. Fig. 3.14b provides the average elapsed times to compute one GPCET moment using *recursiveOne* without and with geometrical symmetry (*symGeo*). It is observed that, on average, the combination is seven-time faster than using *recursiveOne* alone, leading to a reduction in the computation time per moment from 0.4406 to 0.0634 milliseconds at $K = 20$. These results clearly demonstrate that the strategies for fast computation of GPCET moments based on recursive computation of complex exponential functions and on geometrical symmetry are orthogonal. Combining them definitely leads to a multiplication of the computational gains obtained individually by the two strategies.

Finally, it should also be noted here that the value of the parameter s only slightly affects the computation of GPCET kernels/moments by direct computation and does not affect at all the proposed recursive computation strategies. This is because s has no role in the recurrence relations in Eqs. (3.45) and (3.46) even though it appears in the definition of $R_{ns}(r)$. As a result, all the conclusions drawn above hold for every s.

3.5.2 Representation capability and numerical stability

The capability of harmonic function-based moments in representing image functions is demonstrated via image reconstruction. In the following experiments, a set of six character datasets has been generated by sampling 26 vector images of Latin characters in Arial bold font (shown in Fig. 3.15a) to have the sizes of 16×16, 32×32, 64×64, 128×128, 256×256, 512×512, and then quantizing them to bilevels. Images in each of these six datasets are of the same size with some samples are given in Fig. 3.15b–3.15g for the six datasets. The purpose of using datasets of images of different sizes generated from the same source is to investigate the influence of approximation error discussed in Subsection 3.4.3 on the computed moments of comparison methods. The

representation error, which exists in Jacobi polynomial-based methods, will become apparent when moments of high-enough radial orders are involved. Some samples of reconstructed images from the character image "E" of size 64×64 by harmonic function-based methods (GPCET, GPCT, GPST) are given in Fig. 3.16; those by Jacobi polynomial-based (ZM, PZM, OFMM, CHFM, PJFM) and eigenfunction-based (FBM, BFM, DHC) methods are given in Fig. 3.17. In each sub-figure and at each value of K, all moment orders (n, m) satisfying the conditions in Table 3.5 are used for the reconstruction. These conditions are selected so that the moments that capture the lowest-frequency information are used first for the reconstruction. It can be seen from these sub-figures that

- Generally, as more moments are used in the reconstruction process, the reconstructed images get closer to the original ones. However, in the cases of PZM, OFMM, and PJFM, the quality of their reconstructed images deteriorates quickly at $K = 23$, 23, and 21 onwards respectively. Similar phenomena also exist in other Jacobi polynomial-based methods but at higher values of K (46 for ZM and 79 for CHFM).

- Harmonic function-based methods have difficulty in reconstructing the inner region of the images when $s = 2, 4$ with more difficulty at $s = 4$. On the contrary, they have difficulty with the images' outer region when $s = 0.5$. This is the experimental evidence for the information suppression problem caused by the biased distributions of zeros that has been discussed in Subsection 3.3.4.

- Among harmonic function-based methods and at a specific value of s, GPCET has better reconstructed images when K is small. At large values of K, images reconstructed by GPCT/GPST are closest/farthest to the original images at the corresponding values of K. This means that GPCT/GPST require the smallest/largest numbers of moments in order to reconstruct images of similar quality. These superiority/inferiority of GPCT/GPST can be easily observed at boundary regions where $r \simeq 0$ and $r \simeq 1$.

- Harmonic function-based and eigenfunction-based methods capture the image information, especially the edges, better than Jacobi polynomial-based methods.

It thus can be concluded here that the more deviation the value of s from 1 is, the more difficulty harmonic function-based methods will have to reconstruct the inner (when $s > 1$) or the outer (when $s < 1$) region of images. Conversely, harmonic function-based methods can reconstruct quickly the inner or outer region of images when $s < 1$ or $s > 1$ respectively. In other words, the parameter s could be used to control the representation capability of harmonic function-based methods: more emphasis could be placed on certain image regions of interest.

The gauge of reconstruction capability is measured by how well the reconstructed image is in terms of its similarity to the ground-truth one. For this purpose, the reconstruction error between an image and its reconstructed version defined in Eq. (3.35) is considered to be a good measure. In order to compute this measure, a finite set of moments is first calculated and, from this set of moments, images are then reconstructed using Eq. (3.34). Since this process involves the computation of moment kernels in both the decomposition and then reconstruction steps, this measure can additionally be utilized for the investigation of the numerical stability of comparison methods. Let $\mathcal{S}(K)$ be the order set containing all (n, m) that satisfy the conditions stated in Table 3.5 at a specific value of K, Table 3.6 provides the cardinality of $\mathcal{S}(K)$, $|\mathcal{S}(K)|$, of comparison methods. For an image function f defined over the unit disk region, its reconstructed

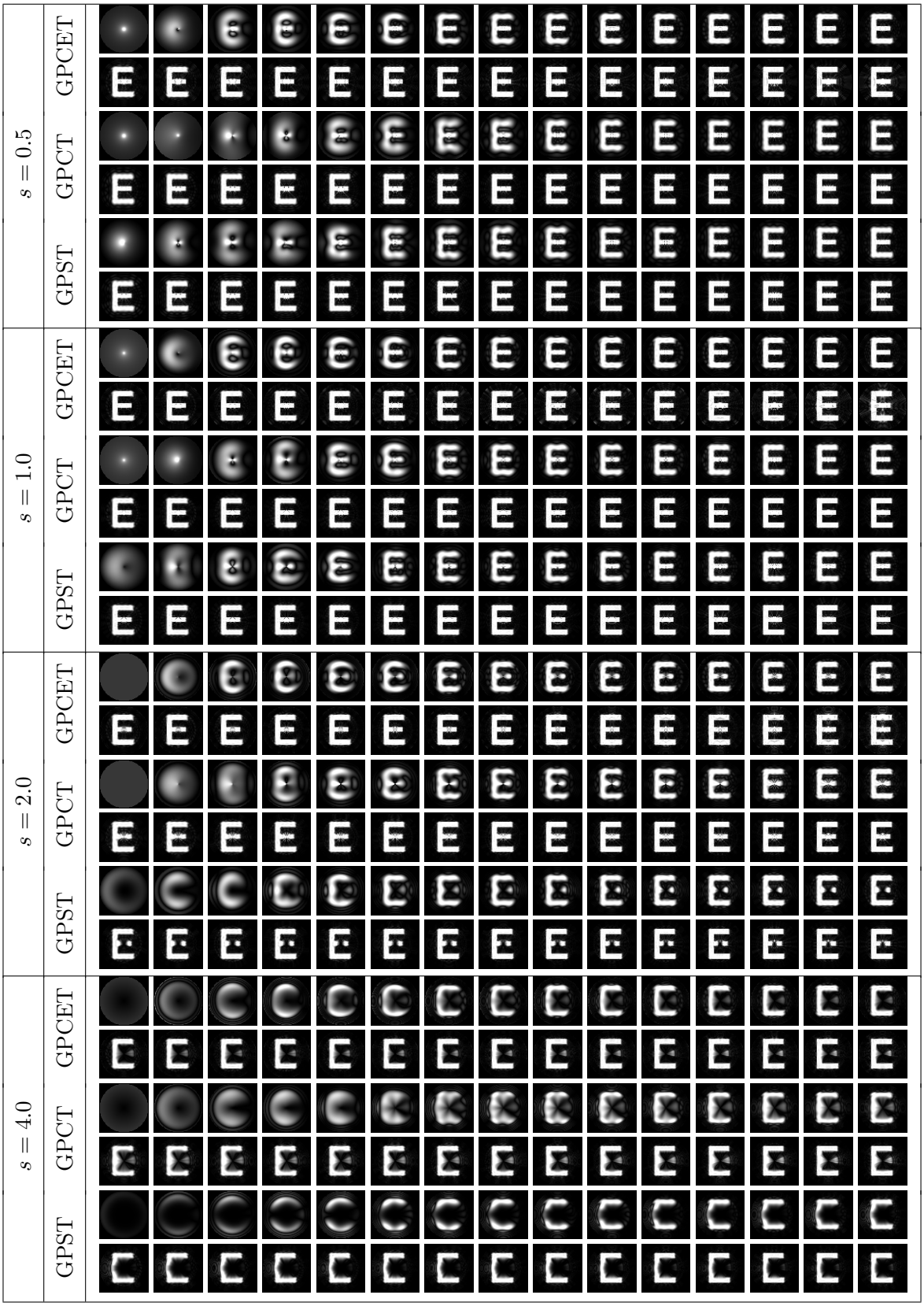
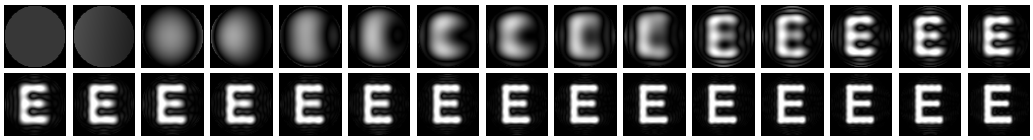
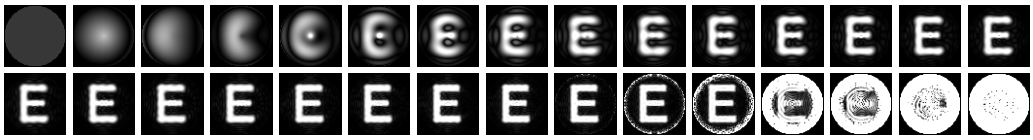


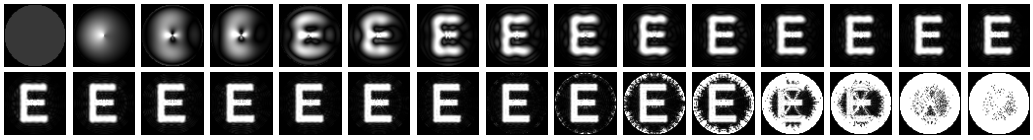
Figure 3.16: Some samples of reconstructed images from the character image "E" of size 64×64 by harmonic function-based methods at $s = 0.5, 1, 2, 4$ for $K = 0, 1, \ldots, 29$ (GPCET, GPCT) and $K = 1, 2, \ldots, 30$ (GPST) (from left to right, top to bottom).



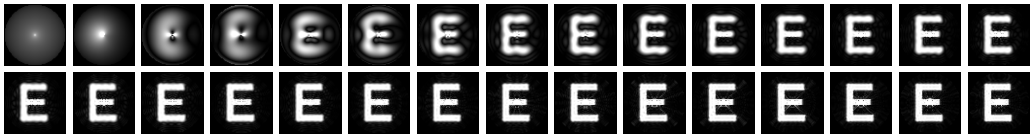
(a) ZM



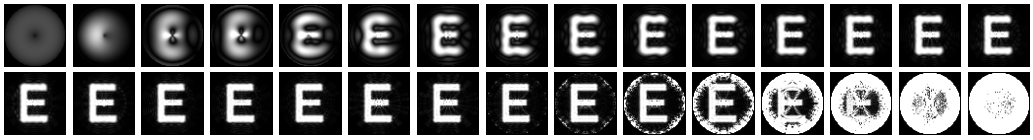
(b) PZM



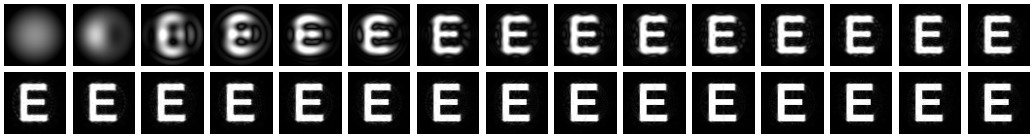
(c) OFMM



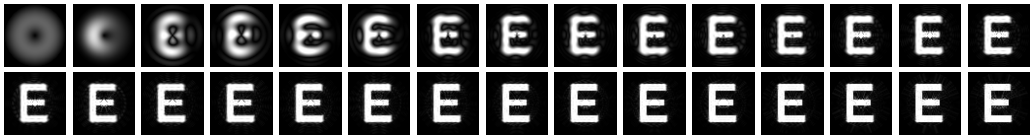
(d) CHFM



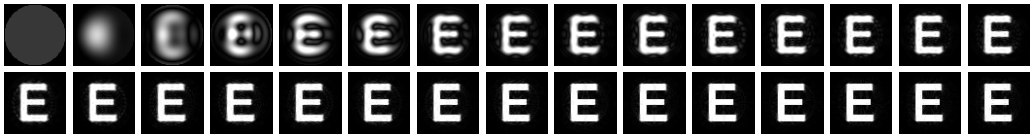
(e) PJFM



(f) FBM



(g) BFM



(h) DHC

Figure 3.17: Some samples of reconstructed images from the character image "E" of size 64×64 by Jacobi polynomial-based (ZM, PZM, OFMM, CHFM, PJFM) and eigenfunction-based (FBM, BFM, DHC) methods for $K = 0, 1, \ldots, 29$ (from left to right, top to bottom).

Table 3.6: The cardinality of the order set $\mathcal{S}(K) = \{(n,m) : n,m \in \mathbb{Z}\}$, $|\mathcal{S}(K)|$, of comparison methods having (n,m) satisfy the conditions stated in Table 3.5 at a specific value of K. All the moments of orders $(n,m) \in \mathcal{S}(K)$ are used to reconstruct the original images.

Method	$\lvert\mathcal{S}(K)\rvert$
ZM	$\frac{(K+1)(K+2)}{2}$
PZM	$(K+1)^2$
OFMM/CHFM/PJFM	$(K+1)(2K+1)$
FBM/BFM/DHC	$(K+1)(2K+1)$
GPCET	$(2K+1)^2$
GRHFM	$(2K+1)^2$
GPCT	$(K+1)(2K+1)$
GPST	$K(2K+1)$

version by using all moments of orders $(n,m) \in \mathcal{S}(K)$ is denoted as $\hat{f}_s^K$. The reconstruction error, normalized by the total image energy, is then defined as

$$\bar{\epsilon}_s^2(K) = \frac{\iint\limits_{x^2+y^2\leq 1} \left[f(x,y) - \hat{f}_s^K(x,y)\right]^2 \mathrm{d}x\mathrm{d}y}{\iint\limits_{x^2+y^2\leq 1} f^2(x,y)\,\mathrm{d}x\mathrm{d}y}.$$

When a set of images is going to be used for the evaluation of representation capability, the above measure is slightly modified and called the mean-square reconstruction error MSRE [215], which has the following definition:

$$\mathrm{MSRE}(K) = \frac{E\left\{\iint\limits_{x^2+y^2\leq 1} \left[f(x,y) - \hat{f}_s^K(x,y)\right]^2 \mathrm{d}x\mathrm{d}y\right\}}{E\left\{\iint\limits_{x^2+y^2\leq 1} f^2(x,y)\,\mathrm{d}x\mathrm{d}y\right\}},$$

where $E\{\cdot\}$ is the expectation in ensemble averaging. In digital computing systems, $\mathrm{MSRE}(K)$ is numerically approximated by the following formula:

$$\mathrm{MSRE}(K) = \frac{E\left\{\sum_{[i,j]\in\mathcal{C}} \left(f[i,j] - \sum_{(n,m)\in\mathcal{S}(K)} H_{nms} h_{nms}[i,j]\right)^2\right\}}{E\left\{\sum_{[i,j]\in\mathcal{C}} \left(f[i,j]\right)^2\right\}}.$$

It is straightforward that, theoretically, $0 \leq \mathrm{MSRE}(K) \leq 1$ and the lower (upper) bounds of $\mathrm{MSRE}(K)$ are reached when $|\mathcal{S}(K)|$ reaches its limits ($|\mathcal{S}(K)| = 0$ or $|\mathcal{S}(K)| = \infty$). However, because of the existence of approximation/representation errors and the unreachable theoretical point $|\mathcal{S}(K)| = \infty$, the statement $0 \leq \mathrm{MSRE}(K) \leq 1$ does not hold anymore; instead, one can only assert that $\mathrm{MSRE}(K) > 0$. In this experiment, a smaller value of $\mathrm{MSRE}(K)$ means that the reconstructed image $\hat{f}_s^K$ is more similar to f or, in other words, a better reconstruction. In

addition, due to Eq. (3.35), MSRE(K) should have a smaller value when more moments are used in the reconstruction process, regardless of their orders.

MSRE curves of harmonic function-based methods on the six character datasets and at different values of s, from 0.1 to 6 with increment of 0.1, are given in Figs. 3.18–3.21. In each of the sub-figures and at a specific value of s in the horizontal axis, there is an MSRE curve with the values of $|\mathcal{S}(K)|$ and MSRE(K) illustrated as the ordinate and the color of the grid points having abscissa s. The values of MSRE(K) which are outside the color display range $[0, 1]$ will be assigned the red color. A red color in MSRE(K) clearly means that the reconstructed image $\hat{f}_s^K$ does not reflect at all f. It is observed from the color patterns of the sub-figures that

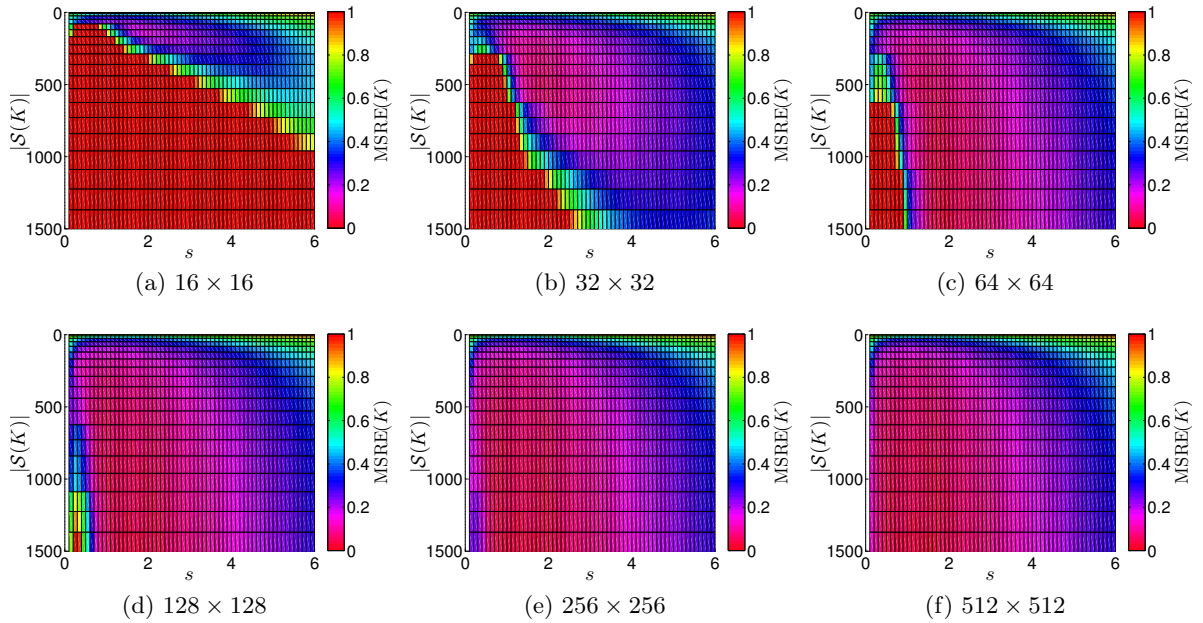
- The color patterns of GPCET are exactly the same as those of GRHFM for all the six character datasets, meaning that the reconstructed images by GPCET and GRHFM are the same. This provides experimental evidence for the equivalence between the radial kernels of GPCET and GRHFM that has been disclosed in Subsection 3.2.1. For the purpose of representation and/or compression, GPCET and GRHFM moments can thus be used interchangeably without any gain or loss in performance. For this reason, in the remaining of this subsection on representation capability and numerical stability, GPCET can be used on behalf of GRHFM in discussions and comparisons with other methods.

- Among GPCET, GPCT, and GPST, a closer resemblance between the color patterns of GPCET and those of GPCT is observed. Moreover, at a specific image size and at the same values of s and $|\mathcal{S}(K)|$, MSRE(K) generally has its highest and lowest values in the case of GPST and GPCT respectively. This means that, in general, among harmonic function-based methods, GPCT has the highest representation capability whereas GPST has the lowest. It should be noted that similar observations are observed in other applications: for compression, cosine functions are much efficient than complex exponential and sine functions; for differential equations, the cosines express a particular choice of boundary conditions.

- For each of the harmonic function-based methods and at a specific value of s, increasing the image size leads to a decrease in the value of MSRE(K) at the same $|\mathcal{S}(K)|$, meaning that the reconstructed image $\hat{f}_s^K$ is more similar to f. The difference between the values of MSRE(K) at different image sizes indicates the existence of approximation error in the computed moments and this provides experimental evidence for the impact of image size on this type of error that has been discussed in Subsection 3.4.3: a smaller image size leads to a higher approximation error, and vice versa. However, the small difference in MSRE(K) between image sizes of 256×256 and 512×512 suggests that the impact of approximation error becomes negligible for large-sized images.

- For each of the harmonic function-based methods and at a specific image size, changing the value of s also leads to a change in the value of MSRE(K) at the same $|\mathcal{S}(K)|$. The value of MSRE(K) decreases slowly when s has a too small or a too large value. This is due to the negligence of the extracted moments on certain regions of the images as discussed in Subsection 3.3.4. Additionally, when $s < 2$, $\sqrt{\frac{sr^{s-2}}{2\pi}} = \sqrt{\frac{s}{2\pi r^{2-s}}}$ and there exists a problem of numerical instability due to the existence of the term r in the denominator of the radial kernels of harmonic function-based methods. The existence of r in the denominator causes very high kernel values near the origin.

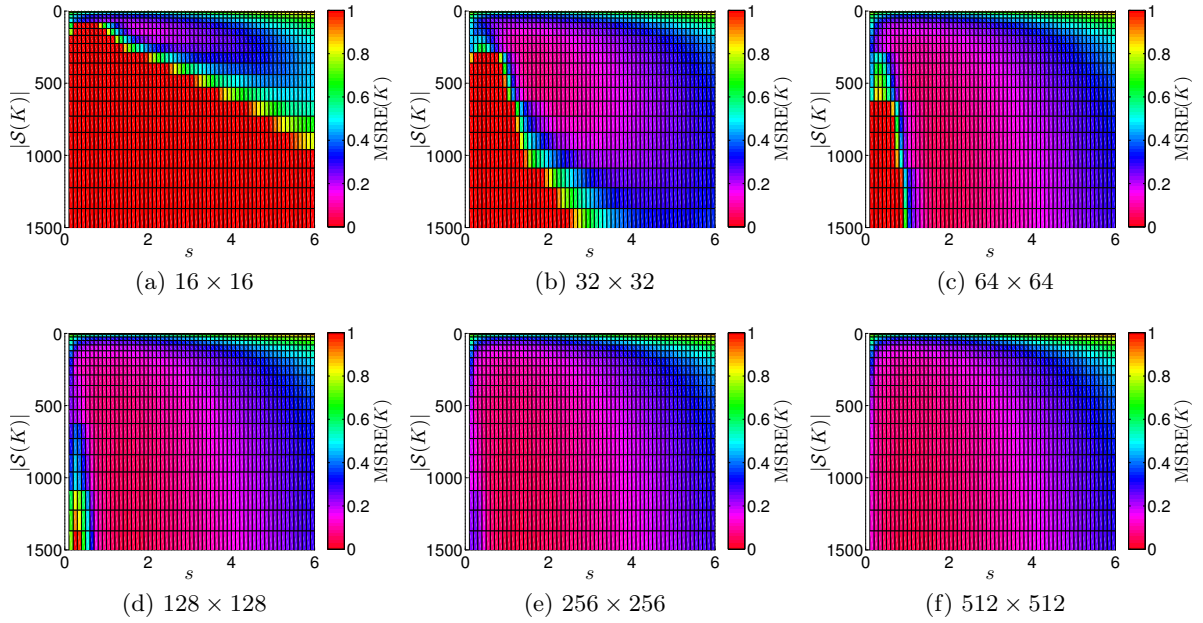
For better visualization and for the purpose of comparison, Fig. 3.22 provides MSRE curves of harmonic function-based methods (GPCET, GPCT, GPST) at selected values of $s = 0.5, 1, 2, 4$



(a) 16×16 (b) 32×32 (c) 64×64

(d) 128×128 (e) 256×256 (f) 512×512

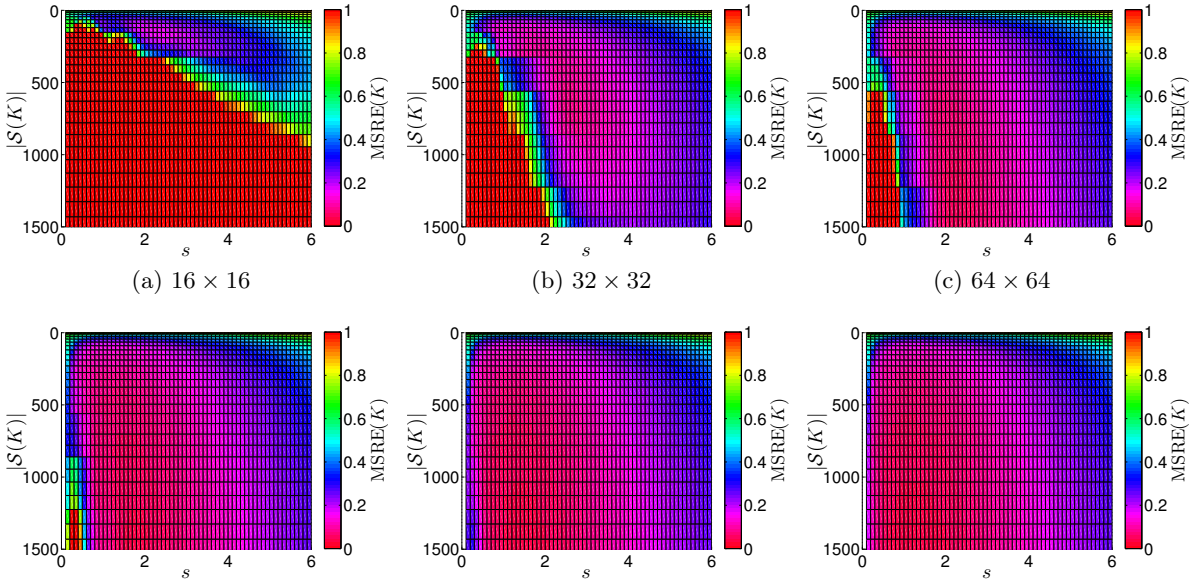
Figure 3.18: MSRE curves of GPCET on the six character datasets and at different values of s. In each sub-figure and at a specific value of s in the horizontal axis, there is an MSRE curve with the values of $|\mathcal{S}(K)|$ and $\mathrm{MSRE}(K)$ illustrated as the ordinate and the color of the grid points having abscissa s.



(a) 16×16 (b) 32×32 (c) 64×64

(d) 128×128 (e) 256×256 (f) 512×512

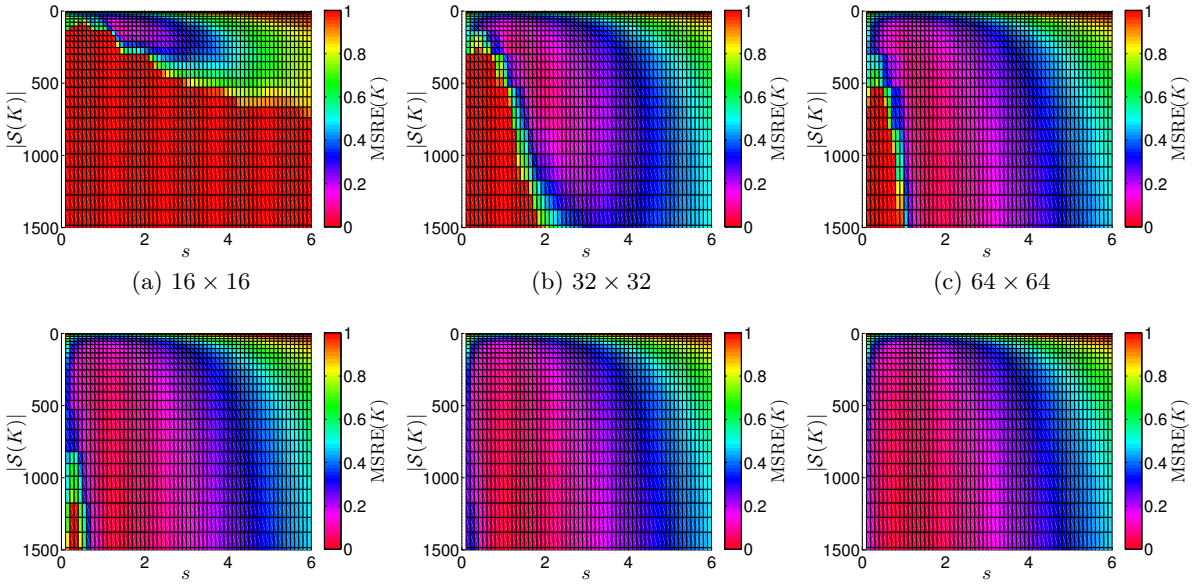
Figure 3.19: MSRE curves of GRHFM on the six character datasets at different values of s. In each sub-figure and at a specific value of s in the horizontal axis, there is an MSRE curve with the values of $|\mathcal{S}(K)|$ and $\mathrm{MSRE}(K)$ illustrated as the ordinate and the color of the grid points having abscissa s.



(a) 16×16 (b) 32×32 (c) 64×64

(d) 128×128 (e) 256×256 (f) 512×512

Figure 3.20: MSRE curves of GPCT on the six character datasets at different values of s. In each sub-figure and at a specific value of s in the horizontal axis, there is an MSRE curve with the values of $|\mathcal{S}(K)|$ and MSRE(K) illustrated as the ordinate and the color of the grid points having abscissa s.



(a) 16×16 (b) 32×32 (c) 64×64

(d) 128×128 (e) 256×256 (f) 512×512

Figure 3.21: MSRE curves of GPST on the six character datasets at different values of s. In each sub-figure and at a specific value of s in the horizontal axis, there is an MSRE curve with the values of $|\mathcal{S}(K)|$ and MSRE(K) illustrated as the ordinate and the color of the grid points having abscissa s.

plotted in the traditional 2D Cartesian coordinate system with $|\mathcal{S}(K)|$ and MSRE(K) illustrated as the abscissa and ordinate respectively. From the six sub-figures that correspond to the six character datasets, all the above observations on harmonic function-based methods can be verified with relative ease.

Comparison of GPCET with Jacobi polynomial-based and eigenfunction-based methods using MSRE curves computed from the six character datasets is given in Fig. 3.23. It can be seen from the sub-figures that

- For all methods, approximation error causes MSRE(K) at the same $|\mathcal{S}(K)|$ to have a higher value at a smaller image size, similar to the phenomenon observed in the comparison among harmonic function-based methods carried out above. This provides experimental evidence for the theoretical arguments on this type of error discussed in Subsection 3.4.3: a smaller image size leads to a higher approximation error, and vice versa. However, the small difference in MSRE(K) between image sizes of 256×256 and 512×512 suggests that the impact of approximation error becomes negligible for large-sized images.

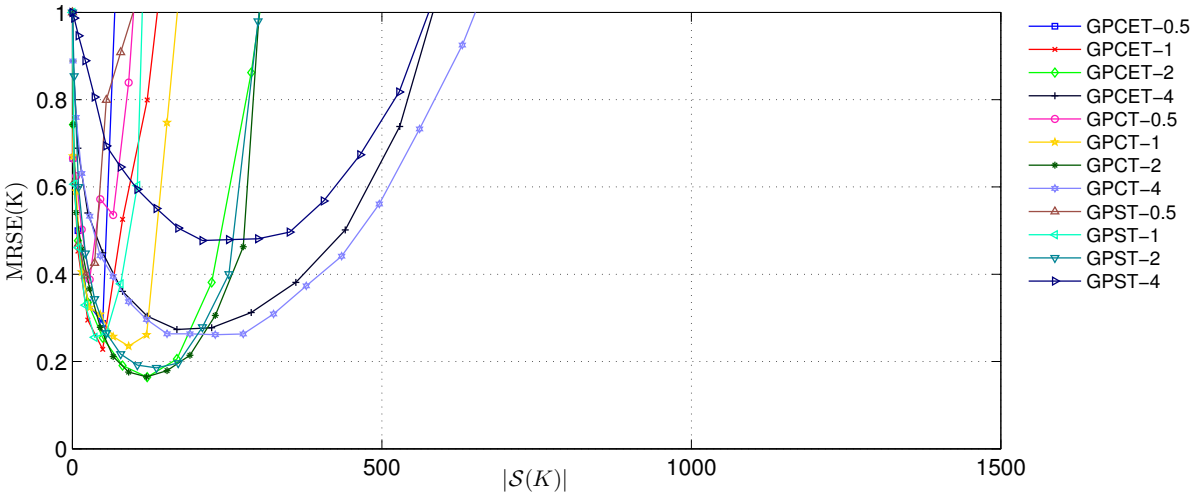
- Numerical stability of Jacobi polynomial-based methods breaks down when K is increased up to certain values. The rapid deterioration in the images reconstructed by Jacobi polynomial-based methods observed in Fig. 3.17 is exhibited here by the sudden upturns in their corresponding MSRE curves at $K = 46$, 21, 23, and 23 for ZM, PZM, OFMM, and PJFH respectively. The MSRE curve of CHFM breaks down later at $K = 79$ (not shown in the figure). These phenomena conform with the theoretical arguments on representation error discussed in Subsection 3.4.3: the starting values of K that cause deterioration here are equal to the starting radial orders that cause roundoff error of the order of unity in Jacobi polynomial-based methods (Table 3.4).

- For large-sized images, except for GPCET ($s = 4$) and the sudden upturns of Jacobi polynomial-based methods, all comparison methods have similar performances with the lowest curves come from eigenfunction-based methods. For small-sized images, ZM has the highest representation capability, followed by GPCET ($s = 4$).

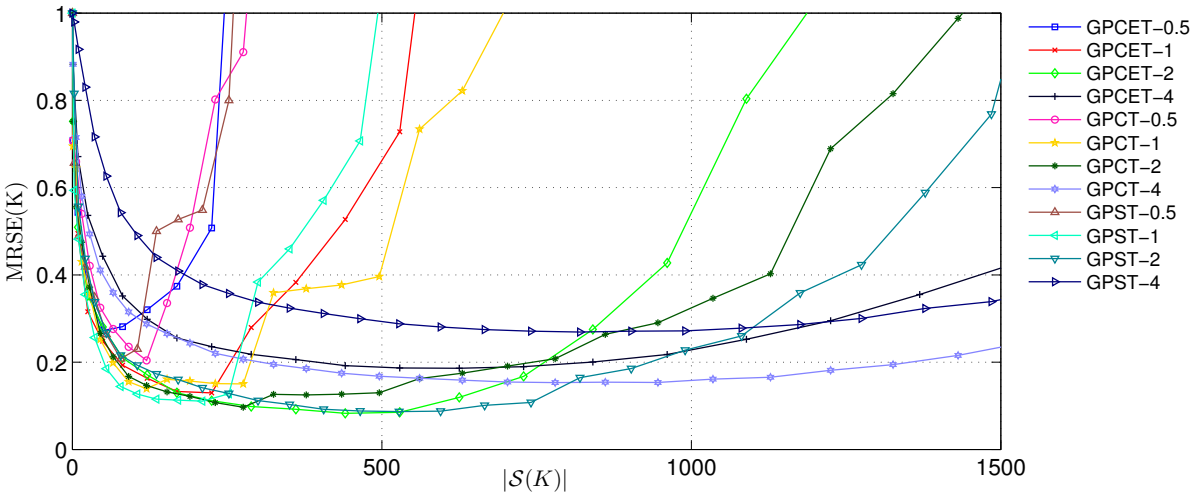
From the experiments carried out in this subsection, it is now clear that approximation and representation errors each affects the computed moments in a different way. Approximation error causes a slightly change in the computed moments. On the contrary, a sudden upturn in the MSRE curve caused by representation error means that the computed moments from that point are totally unreliable and they should not be used in other applications, such as image compression or pattern recognition.

3.5.3 Pattern recognition

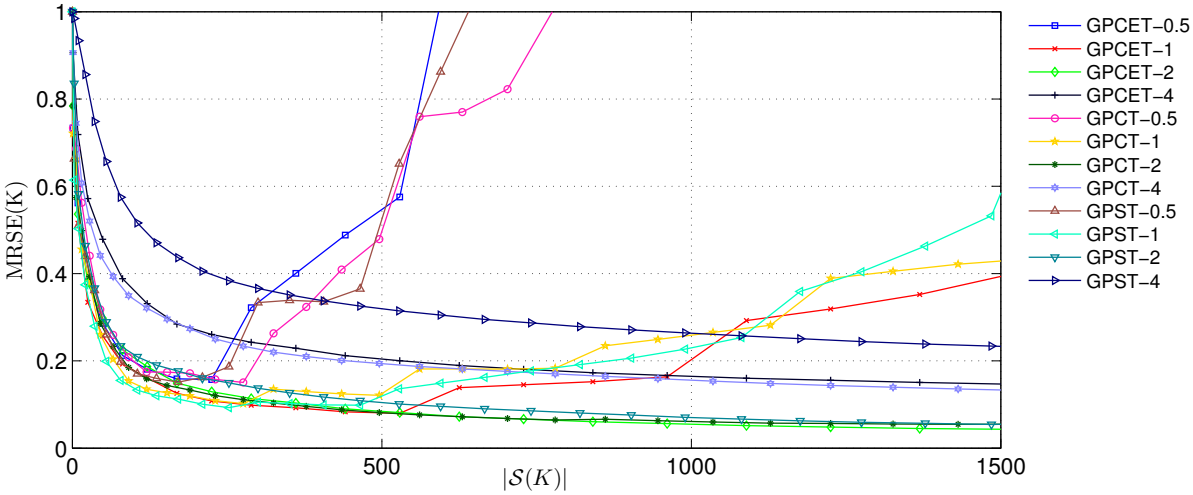
Experiments have been designed to gauge the applicability of harmonic function-based moments in rotation-invariant pattern recognition problems at different levels of additive noise. The experimental images are from the COREL photograph dataset [225]: 100 images are selected, cropped, and scaled to a standard size of 128×128. These 100 images are the training images, their computed moments are taken as the ground-truth for comparison with those of the testing images. Some samples of these training images are given in Fig. 3.24. For these images, only the pixels $[i, j] \in \mathcal{C}$ with $\mathcal{C}$ defined in Eq. (3.38) keep their original intensity values. The remaining pixels, which are irrelevant to the experiments since they are not entirely lying inside the incircle, have their intensity values set to zero.



(a) 16×16

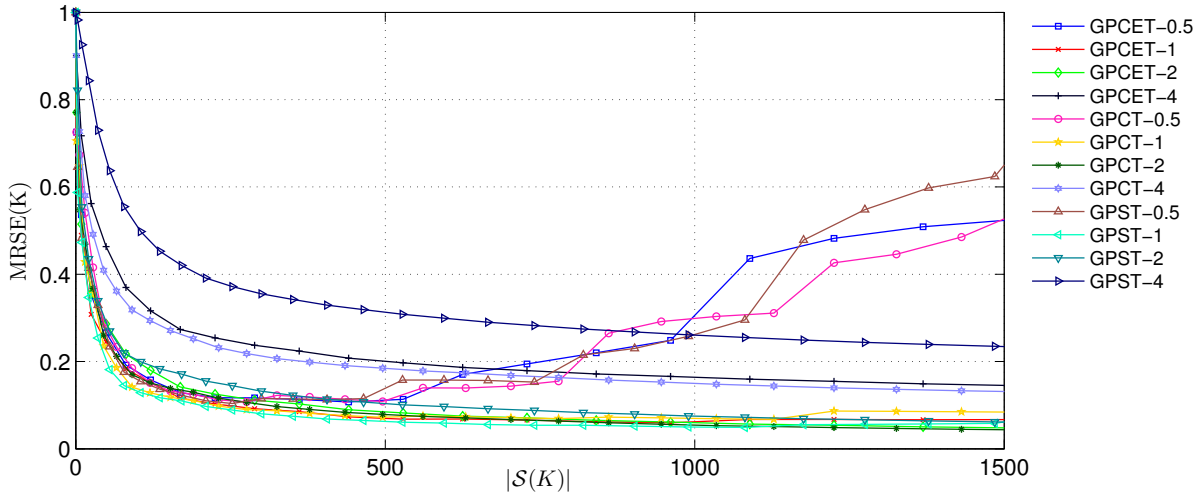


(b) 32×32

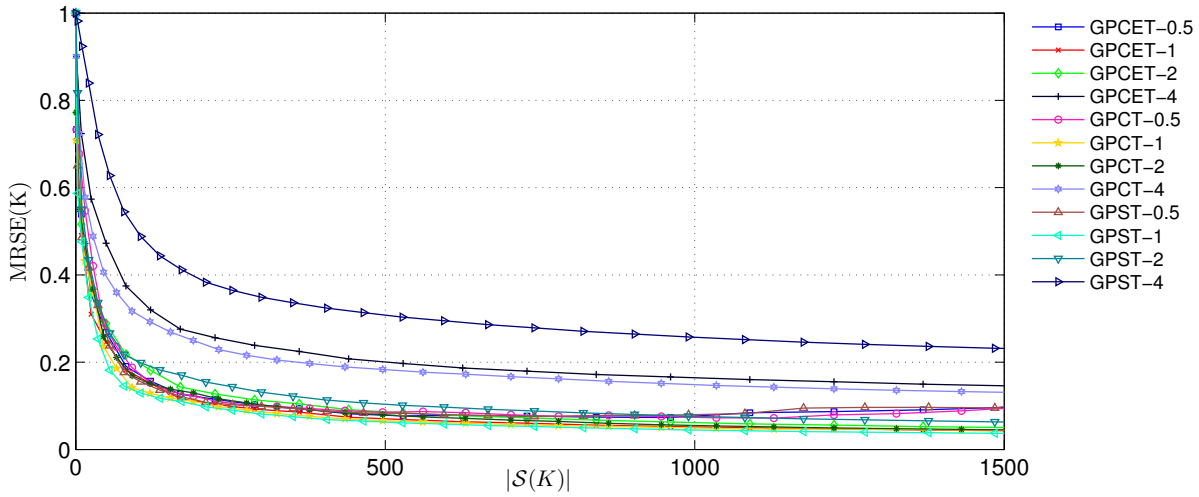


(c) 64×64

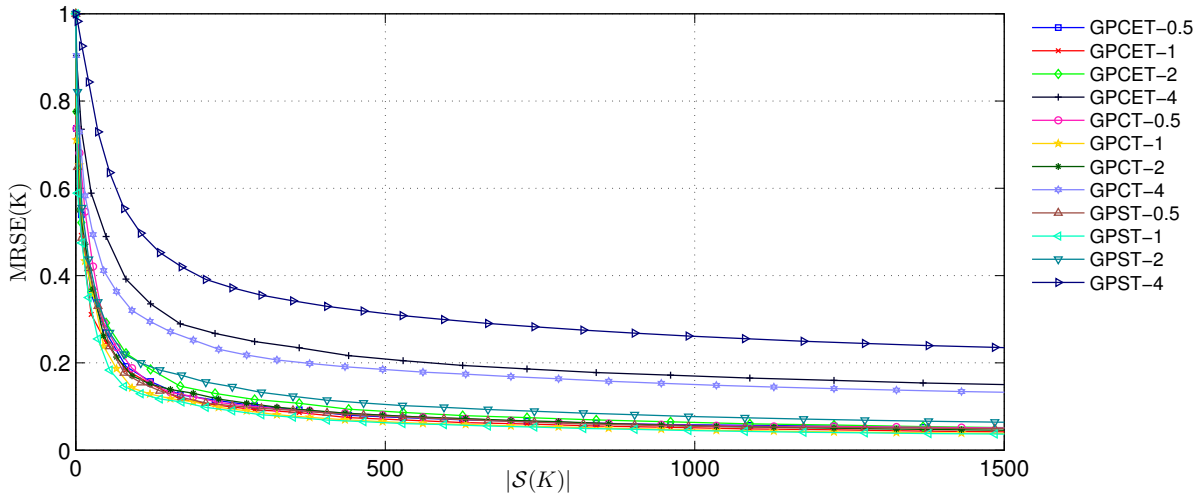
Figure 3.22: MSRE curves of harmonic function-based methods (GPCET, GPCT, GPST) at $s = 0.5, 1, 2, 4$ on the six character datasets (to be continued on the next page).



(d) 128×128

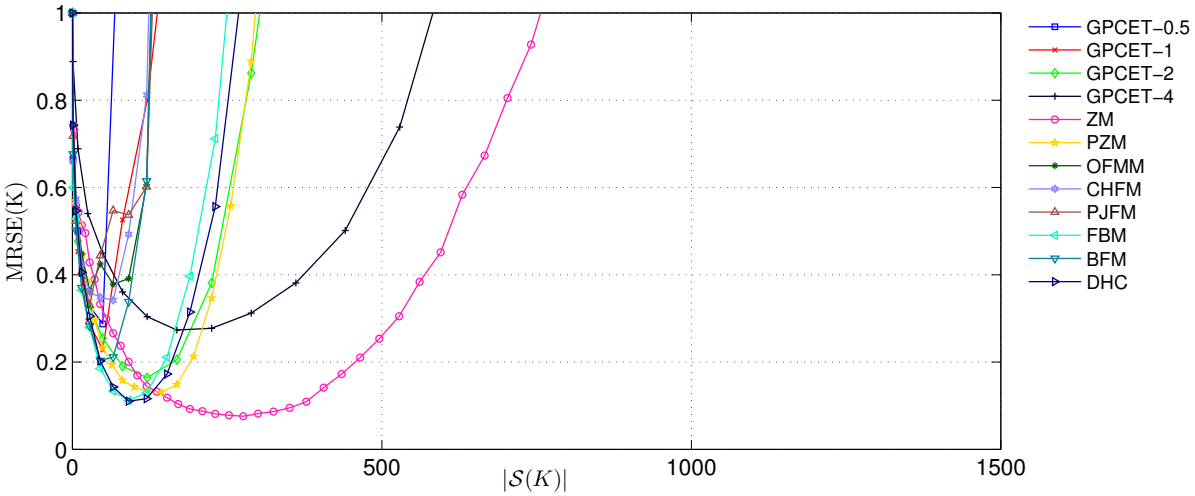


(e) 256×256

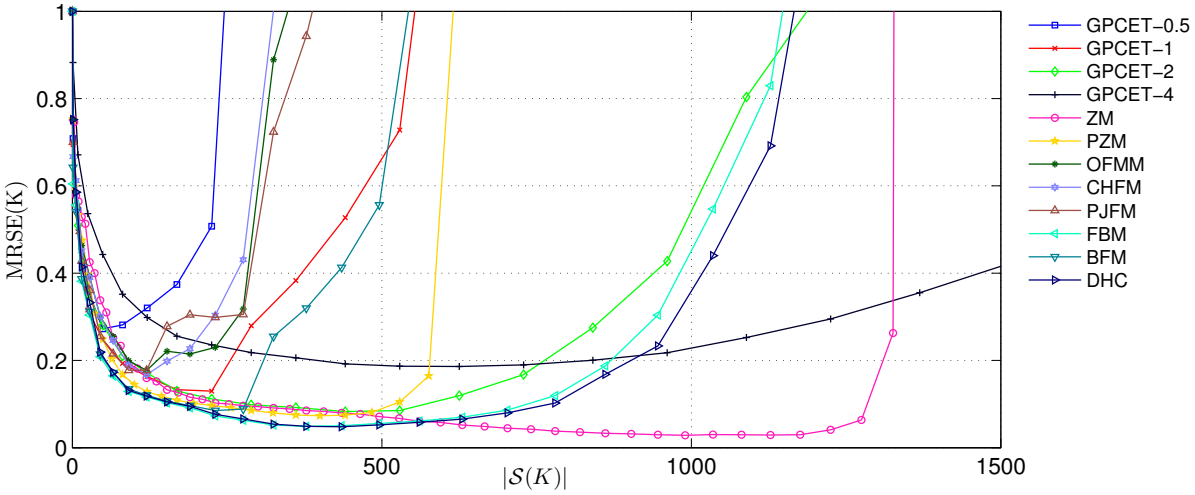


(f) 512×512

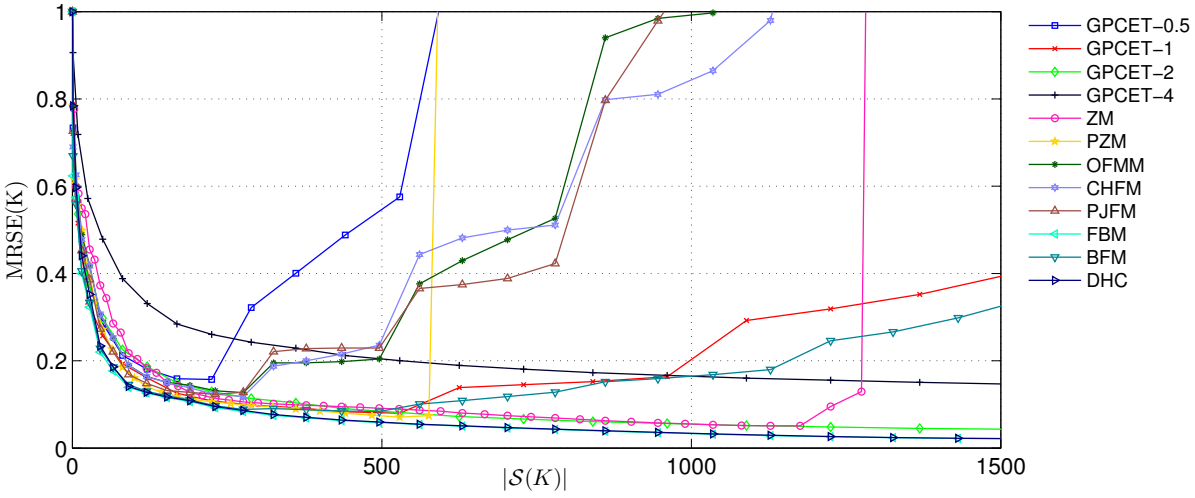
Figure 3.22: MSRE curves of harmonic function-based methods (GPCET, GPCT, GPST) at $s = 0.5, 1, 2, 4$ on the six character datasets.



(a) 16×16

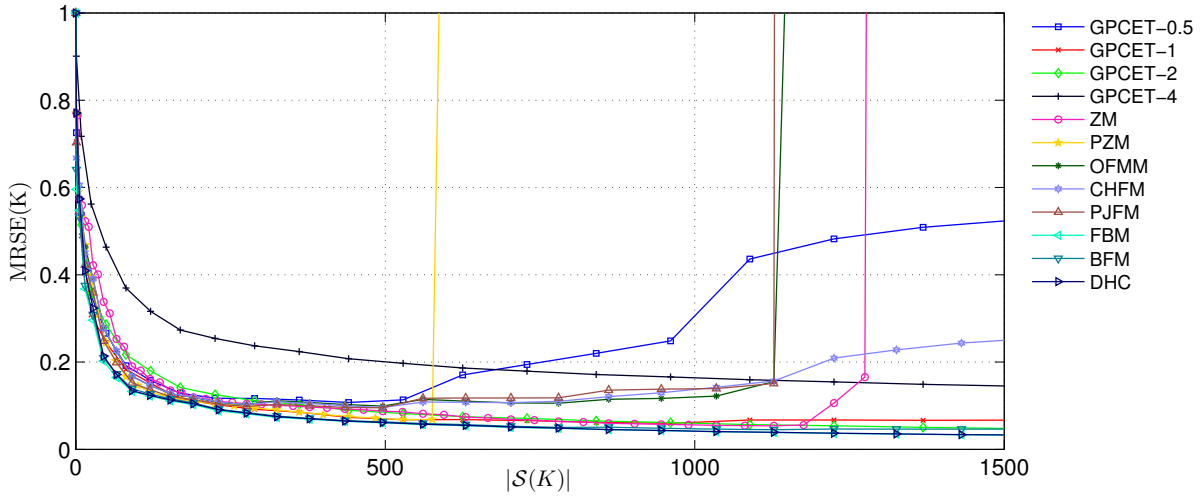


(b) 32×32

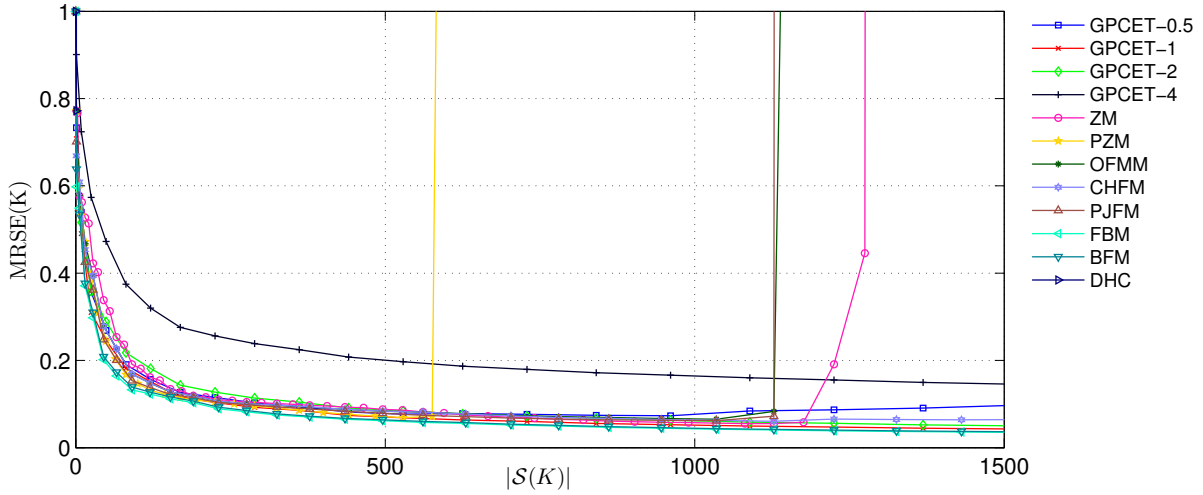


(c) 64×64

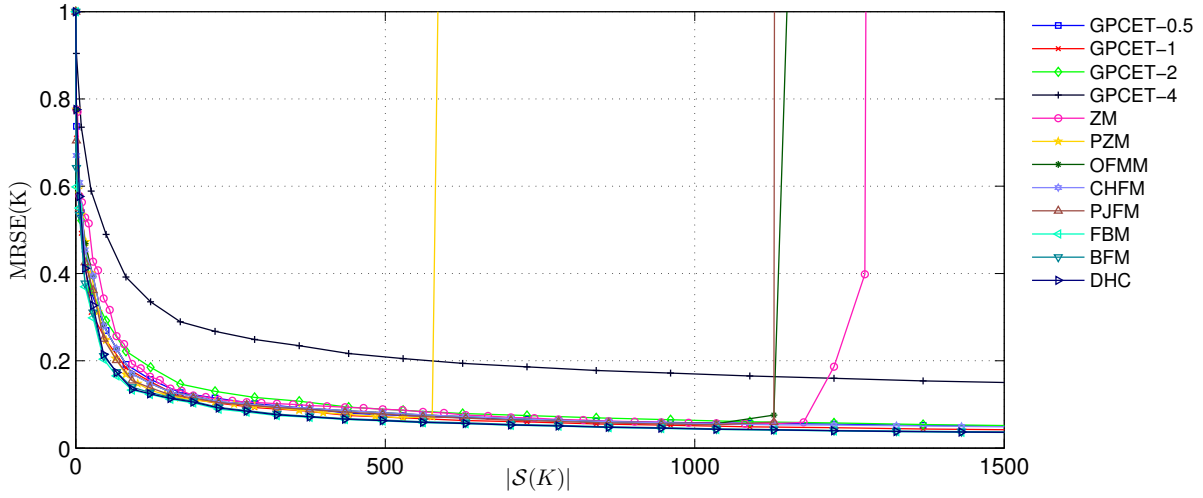
Figure 3.23: MSRE curves of GPCET at $s = 0.5, 1, 2, 4$, Jacobi polynomial-based (ZM, PZM, OFMM, CHFM, PJFM), and eigenfunction-based (FBM, BFM, DHC) methods on the six character datasets (to be continued on the next page).



(d) 128×128



(e) 256×256



(f) 512×512

Figure 3.23: MSRE curves of GPCET at $s = 0.5, 1, 2, 4$, Jacobi polynomial-based (ZM, PZM, OFMM, CHFM, PJFM), and eigenfunction-based (FBM, BFM, DHC) methods on the six character datasets.

Figure 3.24: Ten sample images out of 100 images from the COREL photograph dataset used in the rotation-invariant pattern recognition experiments. These images are used as the models to generate testing images by rotating and then adding noise of different levels to them.

The testing images are generated from the training images by rotating them with rotation angles $\phi = 0°, 30°, \ldots, 330°$ and then contaminating them with Gaussian white noise of variances $\sigma^2 = 0.00, 0.05, 0.10, 0.15, 0.20$[5]. Furthermore, in order to investigate the role of the parameter s in the recognition performance, three different testing datasets are generated separately by restricting the noise to be added to the whole image (NoiseAll), the inner region (NoiseInner), and the outer region (NoiseOuter). The inner and outer regions ensemble form the whole unit-disk region and the boundary between them is the circle of radius 0.5 (i.e., 32 in pixel unit) having the same center with the image. In this way, for each training image, a total of $12 \times 5 \times 3 = 180$ testing images are generated from it, making a total of $100 \times 180 = 18 \times 10^3$ images needed to be classified according to their computed moments. As an example, sample testing images of variance $\sigma^2 = 0.1$ at rotation angles $\phi = 0°, 30°, \ldots, 150°$ from the three different testing datasets (NoiseAll, NoiseInner, and NoiseOuter) that are generated from a single training image are given in Fig. 3.25.

For the purpose of classification, each image of the training and testing datasets is represented by a feature vector, which is the magnitude of its computed moments. Classification is then carried out based on the ℓ_2-norm distances (defined in Eq. (2.30)) between the feature vector of the testing image and those of all training images. It is quite obvious that when the testing images are not contaminated by noise, all the methods (i.e., unit-disk based moments) theoretically produce 100% classification rate on rotation-invariant pattern recognition problems. However, due to the digital nature of the imagery (sampling and quantization errors) and digital computation (approximation and representation errors), moments computed in digital systems are not truly invariant [214]. For this reason and in order to investigate the impact of the size of the feature vector on the classification rate, a set of K values has been used on each dataset as follows:

- NoiseAll: $K = 3, 6, 9, 12, 15$,
- NoiseInner: $K = 1, 2, 3, 4, 5$,
- NoiseOuter: $K = 2, 4, 6, 8, 10$.

The reason for using a different set of K values on each dataset is the difference in the amount of discriminative information that remains in the images after contaminating them with Gaussian white noise. In the presence of noise, the larger the image region contaminated by noise is, the

[5]The variances are normalized values, corresponding to image's intensity values ranging from 0 to 1.

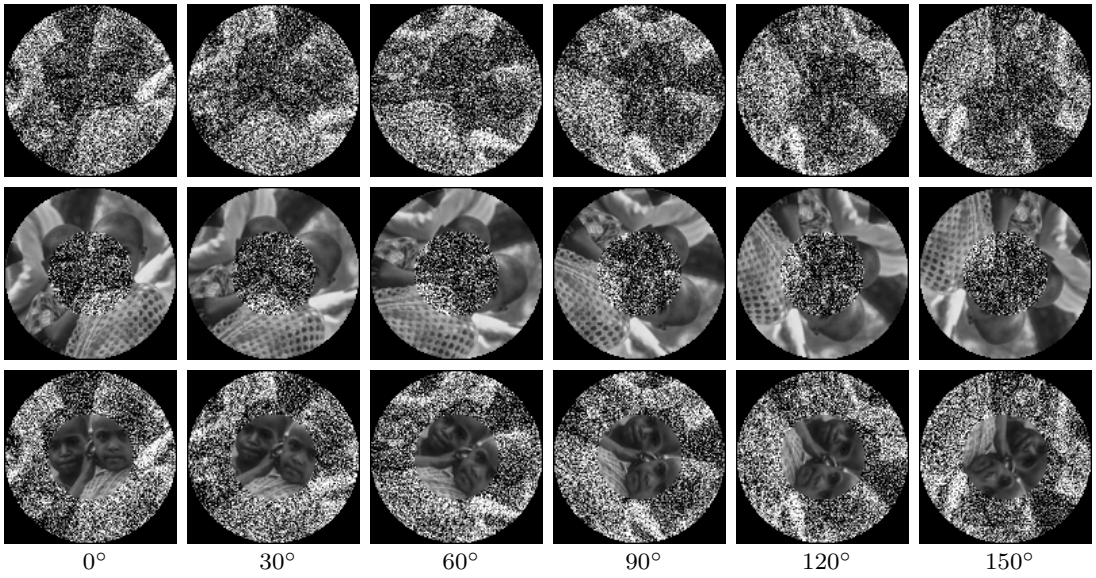


Figure 3.25: Sample noisy images of variance $\sigma^2 = 0.1$ at rotation angles $\phi = 0°, 30°, \ldots, 150°$ (left to right) from the three different testing datasets. *Top row*: noise is added to the whole image (NoiseAll). *Middle row*: noise in the image's inner region (NoiseInner). *Bottom row*: noise in the image's outer region (NoiseOuter).

less the discriminative information remains in the image. It is thus straightforward that, in order to maintain the same classification performance on the three testing datasets, K should take the largest value on NoiseAll and the smallest value on NoiseInner.

The classification rates for harmonic function-based methods (GPCET, GRHFM, GPCT, GPST) at $s = 0.5, 1, 2, 4$ on NoiseAll, NoiseInner, and NoiseOuter datasets using their corresponding sets of K values are given in Tables 3.7, 3.8, and 3.9 respectively. From these tables, it can be observed for all harmonic function-based methods that

- When the testing images get noisier, meaning an increase in σ^2, the classification rates on the same dataset and at the corresponding values of K decrease.

- The classification rates at the corresponding noise levels σ^2 and on the same dataset increase along with the increase in K, or, in other words, increase when more moments are employed in the feature vector.

- On NoiseAll, at the corresponding values of K and σ^2, the classification rates of GRHFM, GPCT, GPST have their peaks at $s = 1$ and decrease as s goes away from 1. However, the classification rate of GPCET does not have a clear trend, it seems to have its minimum value at $s = 2$ and increases as s goes away from 2.

- On NoiseOuter and NoiseInner, as s increases from 0.5 to 4 and at the corresponding values of K and σ^2, the classification rate decreases on NoiseOuter and increases on NoiseInner.

- On average, at the corresponding values of K and σ^2 and on the same dataset, GRHFM has the best classification performance.

The change in classification performance due to changes in the values of σ^2 and/or K is predictable. Additionally, the dependance of performance on the value of s could be explained by

the theoretical arguments in Subsection 3.3.4. The peak performances of harmonic function-based methods on NoiseInner/NoiseOuter at $s = 4/s = 0.5$ are due to the bias of the distributions of zeros of their radial kernels towards $r = 1/r = 0$ at $s = 4/s = 0.5$. This means the discriminative information contained in the computed moments is from the outer/inner region of the images where the noise is not present. Similarly, the peak performances on NoiseAll of GRHFM, GPCT, and GPST at $s = 1$ are due to the uniform distributions of zeros of their radial kernels over $0 \leq r \leq 1$ at $s = 1$. The abnormal trend observed from the classification rates of GPCET could be explained by the complex nature of its radial kernels. Although the zeros of their real and imaginary parts are clearly defined, GPCET radial kernels themselves do not have zeros due to the employed complex exponential functions. The dominance of GRHFM over the other harmonic function-based methods has the following explanations:

- GRHFM has been shown to be a variant of GPCET in terms of representation, similar to the equivalence between different forms of Fourier series. However, the trigonometric function-based radial kernels of GRHFM do not contain phase information, unlike the exponential function-based radial kernels of GPCET. Accordingly, GRHFM suffers less from the loss of phase information [116] when a magnitude operator is used to compute rotation-invariant feature vectors, resulting in its better performance over GPCET.

- GPCT and GPST are respectively defined based on the cosine and sine series, which are the so-called half-range expansions of a function. They are special cases of the Fourier series, arising naturally when attempting to decompose even/odd functions. Due to this interpretation, many of the properties of cosine and sine series are less elegant and more involved than the corresponding ones of the Fourier series [240] and this may explain for the inferiority of GPCT and GPST to GRHFM in terms of classification rate.

Taking GRHFM as the representative of harmonic function-based methods, comparison of GRHFM at $s = 0.5, 1, 2, 4$ with non-orthogonal (ART, GFD, RM), Jacobi polynomial-based (ZM, PZM, OFMM, CHFM, PJFM), and eigenfunction-based (FBM, BFM, DHC) methods on NoiseAll, NoiseInner, and NoiseOuter datasets are given in Tables 3.10, 3.11, and 3.12 respectively. Besides similar trends in the dependance of the classification rates on the values of K and σ^2, it can also be seen from these tables that

- Non-orthogonal methods have smaller classification rates than those of orthogonal ones on the three testing datasets at the corresponding values of K and σ^2. These inferior results demonstrate clearly that non-orthogonal methods are less effective than orthogonal ones.

- Jacobi polynomial-based methods have lower performance than that of GRHFM at its peaks (i.e., $s = 1$ on NoiseAll, $s = 4$ on NoiseInner, and $s = 0.5$ on NoiseOuter) at the corresponding values of K and σ^2 and on the same dataset, except for OFMM on NoiseAll.

- Eigenfunction-based methods perform better than GRHFM on NoiseAll ($s = 1$) when the value of K is large enough (i.e., $K \geq 6$). They perform worse than GRHFM on NoiseInner ($s = 4$) and have comparable performance with GRHFM on NoiseOuter ($s = 0.5$).

From the experiments carried out in this subsection, it now can be concluded that harmonic function-based moments could be used to define region-based feature vectors in rotation-invariant pattern recognition problems. They outperform non-orthogonal and Jacobi polynomial-based moments and have comparable performances with those of eigenfunction-based moments on the carefully-designed experimental datasets. Moreover, the decisive role of s in the recognition performance, as theoretically argued in Subsection 3.3.4, has also been confirmed experimentally.

3.6 Conclusions

In this chapter, the generalizations of existing unit disk-based orthogonal moments using harmonic functions have been pursued with the radial kernels are defined based on

- GPCET: Fourier series using complex exponential functions.
- GRHFM: Fourier series using trigonometric functions.
- GPCT: cosine series.
- GPST: sine series.

The sets of orthogonal kernels of harmonic function-based moments have been proven to be complete in a Hilbert space of square-integrable continuous complex-valued functions. Moreover, the use of a parameter s in the definition results in four classes of moments that have beneficial properties of the original moments (PCET, RHFM, PCT, and PST) while giving more flexibility in their definitions. This flexibility has been demonstrated to be useful both theoretically and experimentally in some particular applications, especially in image compression and pattern recognition problems.

The simple, resembling, and relating definitions of harmonic function-based kernels have resulted in an almost-constant kernel computation time, regardless of the maximal kernel order. This makes a strong contrast with Jacobi polynomial-based and eigenfunction-based methods where a higher order means a longer kernel computation time. Recursive strategies for fast computation of harmonic function-based kernels have also been proposed by exploiting the recurrence relations between harmonic functions, leading to a method that is approximately 10-time faster than direct computation. When compared with the current state-of-the-art strategy for fast computation of ZM kernels, the proposed method is also approximately five-time faster. Moreover, combination of the proposed method with the method based on geometrical symmetry leads to a multiplication of the computational gains obtained individually by the two combining methods.

In terms of representation capability, like all other methods, harmonic function-based methods suffer from approximation error and, unlike Jacobi polynomial-based methods, do not suffer from representation error. As a result, the numerical instability that is common in Jacobi polynomial-based methods does not exist in harmonic function-based methods. Apart from this numerical instability, the representation capabilities of all unit disk-based orthogonal moments are comparable. However, the ability to control the representation capability according to image regions by changing the value of s draws a distinction between harmonic function-based methods and the others. Based on this ability, it is possible to have a faster reconstruction of the image function in certain image regions of interest, leading to potential applications in image compression.

Finally, in rotation-invariant pattern recognition problems, harmonic function-based methods have been shown to generally perform better than non-orthogonal and Jacobi polynomial-based methods while having comparable performance with that of eigenfunction-based methods. For this reason, harmonic function-based moments could be used to define region-based feature vectors in rotation-invariant pattern recognition problems. Moreover, the decisive role of s in the recognition performance has been confirmed experimentally and s can be used to place emphasis of the feature vector to be extracted on certain image regions that contain discriminative information, leading to potential applications in pattern recognition. This ability is also a distinct characteristic of harmonic function-based methods that the others do not have.

Table 3.7: Classification rates of harmonic function-based methods (GPCET, GRHFM, GPCT, GPST) at $s = 0.5, 1, 2, 4$ on NoiseAll dataset under different levels of Gaussian noise $\sigma^2 = 0.00, 0.05, 0.10, 0.15, 0.20$ and at different values of $K = 3, 6, 9, 12, 15$.

K	σ^2	GPCET				GRHFM				GPCT				GPST			
		s=0.5	s=1	s=2	s=4	s=0.5	s=1	s=2	s=4	s=0.5	s=1	s=2	s=4	s=0.5	s=1	s=2	s=4
3	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.00	98.00	97.08	96.08	98.00	97.92	97.75	97.00	96.92	97.67	96.75	94.67	95.75	96.58	95.67	87.92
	0.10	87.67	81.67	75.25	78.00	86.58	87.08	81.58	81.83	75.67	83.83	75.08	75.92	72.58	78.75	70.50	70.17
	0.15	57.00	56.00	47.75	49.42	59.50	65.42	56.50	50.42	46.33	53.75	46.67	47.00	45.33	50.75	42.00	41.33
	0.20	34.75	35.08	27.67	28.25	38.42	41.25	33.25	32.08	28.33	33.08	27.42	28.33	28.17	32.00	22.08	23.00
6	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.00	98.00	98.00	97.92	98.00	98.00	98.00	97.92	98.00	98.00	98.00	97.92	98.17	98.00	98.00	96.00
	0.10	93.42	91.83	84.42	86.17	92.75	92.50	88.25	88.83	88.67	91.08	87.42	86.50	90.58	91.92	87.83	82.92
	0.15	73.33	68.17	60.50	66.67	74.00	76.25	67.42	65.83	62.25	71.17	63.58	62.67	67.92	73.25	62.50	64.67
	0.20	43.67	42.17	37.33	43.33	45.58	49.50	44.83	43.83	37.42	42.92	39.92	38.50	42.83	46.92	38.00	40.58
9	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.33	98.00	98.00	96.83
	0.10	94.33	93.08	88.33	89.50	94.00	93.25	91.42	90.17	91.58	92.33	91.17	88.08	93.33	92.67	91.58	86.92
	0.15	79.92	77.17	67.92	73.83	79.50	81.33	72.83	74.00	69.83	77.33	70.92	69.83	74.58	80.00	71.25	72.42
	0.20	50.83	49.00	44.42	50.50	51.50	56.75	52.58	51.00	45.42	49.25	48.17	46.25	46.92	52.33	46.42	48.75
12	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.08	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.50	98.00	98.00	97.00
	0.10	95.00	94.58	90.33	89.75	95.00	94.58	93.25	91.42	93.00	93.17	92.58	89.58	95.33	93.25	92.92	88.17
	0.15	84.17	80.58	72.58	76.67	83.42	84.75	76.25	78.00	76.08	81.08	75.50	73.58	79.08	83.50	74.08	74.67
	0.20	57.17	55.33	50.75	58.08	58.42	62.33	57.67	57.08	49.83	54.75	53.42	50.75	51.67	58.25	51.83	52.42
15	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.17	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.00	98.75	98.00	98.00	97.08
	0.10	96.17	95.67	92.00	90.75	95.58	95.33	94.25	92.25	94.25	93.92	93.33	90.17	96.00	93.83	93.25	88.83
	0.15	86.92	83.08	75.92	79.08	85.75	86.50	79.67	79.58	78.50	83.58	78.33	76.42	82.17	85.33	76.83	76.33
	0.20	61.67	60.00	55.58	62.25	63.33	66.17	61.17	62.33	52.92	59.00	57.92	56.17	56.25	61.92	55.67	55.83

Table 3.8: Classification rates of harmonic function-based methods (GPCET, GRHFM, GPCT, GPST) at $s = 0.5, 1, 2, 4$ on NoiseInner dataset under different levels of Gaussian noise $\sigma^2 = 0.00, 0.05, 0.10, 0.15, 0.20$ and at different values of $K = 1, 2, 3, 4, 5$.

K	σ^2	GPCET				GRHFM				GPCT				GPST			
		s=0.5	s=1	s=2	s=4	s=0.5	s=1	s=2	s=4	s=0.5	s=1	s=2	s=4	s=0.5	s=1	s=2	s=4
1	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.33	98.67	99.33	97.33
	0.05	91.50	92.58	97.33	99.08	94.00	97.67	97.67	99.00	86.42	86.00	87.75	98.67	7.58	16.67	43.83	83.08
	0.10	75.08	77.42	84.83	97.58	79.08	83.00	91.83	98.92	63.50	69.58	70.50	92.08	4.50	8.67	28.58	73.25
	0.15	63.08	66.50	76.50	95.17	67.50	71.08	82.83	97.33	45.50	53.08	57.33	84.67	3.42	6.08	21.00	66.67
	0.20	52.00	56.33	67.08	93.50	56.17	61.75	69.42	94.83	37.33	44.00	48.33	79.33	3.33	4.83	17.58	63.33
2	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.83	98.92	100.00	100.00	99.25	99.75	100.00	100.00	99.00	99.67	100.00	100.00	93.67	98.58	100.00	100.00
	0.10	96.75	96.50	99.00	100.00	96.17	98.67	100.00	100.00	89.75	96.33	97.83	100.00	76.00	87.17	99.50	100.00
	0.15	91.92	92.92	98.50	100.00	92.25	94.67	98.42	100.00	80.67	89.67	94.42	100.00	62.67	77.17	94.67	100.00
	0.20	86.00	88.50	93.75	100.00	86.17	90.50	94.92	99.83	70.08	81.25	87.67	99.50	49.75	65.67	92.83	100.00
3	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	100.00	100.00	100.00	100.00	99.58	99.92	100.00	100.00	99.58	99.75	100.00	100.00	99.00	99.17	100.00	100.00
	0.10	98.83	98.83	99.08	100.00	98.50	98.92	100.00	100.00	97.17	98.83	100.00	100.00	93.58	98.58	100.00	100.00
	0.15	96.58	96.83	99.00	100.00	97.08	97.25	99.17	100.00	92.92	97.17	98.75	100.00	82.00	91.92	100.00	100.00
	0.20	95.75	95.92	98.42	100.00	93.92	95.33	98.58	99.92	84.67	92.33	97.83	99.83	70.08	88.25	99.92	100.00
4	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	100.00	100.00	100.00	100.00	99.67	99.83	100.00	100.00	99.67	99.75	100.00	100.00	99.00	99.33	100.00	100.00
	0.10	99.00	98.92	99.75	100.00	98.67	99.00	100.00	100.00	98.25	99.00	100.00	100.00	95.50	98.83	100.00	100.00
	0.15	97.42	97.17	99.00	100.00	97.92	97.67	100.00	100.00	96.42	98.00	99.75	100.00	90.83	95.50	100.00	100.00
	0.20	96.75	96.92	98.83	100.00	96.92	96.92	99.42	99.83	92.17	97.00	98.33	99.08	83.67	93.92	99.58	100.00
5	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	100.00	100.00	100.00	100.00	99.92	100.00	100.00	100.00	99.92	100.00	100.00	100.00	99.00	99.92	100.00	100.00
	0.10	99.00	99.00	100.00	100.00	98.92	99.00	100.00	100.00	98.58	99.00	100.00	100.00	97.17	99.00	100.00	100.00
	0.15	97.58	97.58	99.42	100.00	98.00	98.00	100.00	100.00	97.42	98.25	100.00	100.00	93.17	97.75	100.00	100.00
	0.20	97.08	97.08	98.92	100.00	98.00	97.17	99.58	99.92	95.50	97.33	99.17	99.17	88.00	96.08	99.50	100.00

Table 3.9: Classification rates of harmonic function-based methods (GPCET, GRHFM, GPCT, GPST) at $s = 0.5, 1, 2, 4$ on NoiseOuter dataset under different levels of Gaussian noise $\sigma^2 = 0.00, 0.05, 0.10, 0.15, 0.20$ and at different values of $K = 2, 4, 6, 8, 10$.

K	σ^2	GPCET				GRHFM				GPCT				GPST			
		s=0.5	s=1	s=2	s=4	s=0.5	s=1	s=2	s=4	s=0.5	s=1	s=2	s=4	s=0.5	s=1	s=2	s=4
2	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.00	97.17	97.08	94.25	97.42	98.08	96.08	94.08	94.33	95.08	92.92	93.25	95.67	92.58	80.17	79.58
	0.10	95.42	86.25	75.58	75.83	92.92	91.00	82.58	79.08	76.83	85.83	72.17	70.08	79.25	69.75	53.17	47.42
	0.15	79.67	73.00	57.42	46.17	82.75	78.33	65.25	52.92	57.67	64.08	55.83	45.42	64.33	46.67	32.17	25.08
	0.20	65.75	59.58	42.83	29.50	67.92	65.25	52.25	34.92	46.58	51.00	40.08	31.08	49.00	33.92	20.92	16.33
4	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.00	98.83	98.00	97.92	98.42	99.00	98.00	97.92	98.00	98.83	98.00	97.42	98.33	98.17	97.17	94.67
	0.10	98.00	96.75	89.42	87.25	96.42	95.08	90.83	88.58	94.17	93.33	88.50	85.67	95.92	94.17	87.42	78.75
	0.15	94.42	88.33	77.00	68.33	92.33	89.33	79.33	71.67	83.33	85.17	73.33	68.17	87.08	85.75	69.50	56.83
	0.20	83.25	75.50	62.83	48.25	82.50	79.42	67.33	52.00	69.50	71.08	61.92	48.75	76.33	75.83	52.33	34.50
6	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.00	99.00	98.00	98.00	98.92	99.00	98.00	98.00	98.00	99.00	98.00	98.00	99.00	99.00	98.00	96.67
	0.10	98.00	98.00	93.25	89.33	98.00	97.58	94.67	89.92	96.75	95.08	93.00	88.83	97.00	96.17	91.58	84.42
	0.15	95.17	91.17	83.75	77.17	93.08	92.00	85.08	79.42	88.92	90.67	82.25	77.50	92.25	90.83	79.00	68.50
	0.20	90.50	82.67	70.75	58.58	88.17	84.58	73.42	61.25	77.75	80.00	69.42	57.42	86.83	84.08	62.33	45.67
8	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.50	99.00	98.00	98.00	99.25	99.00	98.00	98.00	98.42	99.00	98.00	98.00	99.50	99.00	98.00	96.83
	0.10	98.00	98.00	95.00	89.92	98.00	97.92	96.50	90.58	97.75	96.92	95.83	89.50	97.00	97.00	92.67	86.50
	0.15	95.42	92.58	85.67	82.00	94.08	92.08	87.33	82.92	91.42	91.83	85.00	81.83	95.50	93.08	83.75	73.75
	0.20	91.33	86.17	74.83	65.00	88.92	87.08	77.50	67.50	81.33	84.67	74.08	64.25	88.67	86.17	68.42	52.25
10	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.58	99.00	98.00	98.00	99.42	99.00	98.00	98.00	98.58	99.00	98.00	98.00	100.00	99.00	98.00	97.00
	0.10	98.00	98.00	96.50	90.17	98.00	97.92	97.17	91.00	98.00	97.42	96.50	90.00	97.00	97.50	93.50	88.00
	0.15	95.58	93.08	87.25	83.58	94.50	92.17	88.50	84.17	92.25	91.92	86.33	83.75	95.92	93.17	85.17	76.00
	0.20	91.75	87.67	78.17	70.42	89.67	88.67	79.92	72.33	83.58	86.92	76.42	69.33	91.08	88.00	73.50	58.00

Table 3.10: Classification rates of GRHFM at $s = 0.5, 1, 2, 4$, non-orthogonal (ART, GFD, RM), Jacobi polynomial-based (ZM, PZM, OFMM, CHFM, PJFM), and eigenfunction-based (FBM, BFM, DHC) methods on NoiseAll dataset under different levels of Gaussian noise $\sigma^2 = 0.00, 0.05, 0.10, 0.15, 0.20$ and at different values of $K = 3, 6, 9, 12, 15$.

K	σ^2	GRHFM				ART	GFD	RM	ZM	PZM	OFMM	CHFM	PJFM	FBM	BFM	DHC
		s=0.5	s=1	s=2	s=4											
3	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.00	97.92	97.75	97.00	96.75	97.92	79.25	78.67	93.83	97.42	97.92	96.50	97.33	97.92	96.42
	0.10	86.58	87.08	81.58	81.83	79.92	83.50	39.17	43.92	68.00	77.00	78.50	79.58	87.67	84.42	83.50
	0.15	59.50	65.42	56.50	50.42	51.50	54.50	19.33	20.42	43.00	55.50	53.67	55.92	66.58	65.67	55.25
	0.20	38.42	41.25	33.25	32.08	28.67	31.92	11.17	9.67	26.08	36.83	31.67	32.50	42.33	39.67	34.83
6	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.00	98.00	98.00	97.92	98.00	98.00	93.08	97.58	98.00	98.00	98.00	98.00	98.25	98.00	98.00
	0.10	92.75	92.50	88.25	88.83	89.75	91.83	63.50	84.58	90.58	92.83	89.75	89.08	94.50	94.42	93.33
	0.15	74.00	76.25	67.42	65.83	68.58	67.42	36.00	56.75	67.75	73.83	67.42	67.00	81.67	82.83	75.67
	0.20	45.58	49.50	44.83	43.83	44.75	41.67	20.67	35.00	44.33	54.00	42.00	44.58	57.42	56.83	52.92
9	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.00	98.00	98.00	98.00	98.00	98.00	95.50	97.92	98.00	98.00	98.00	98.00	99.00	98.00	98.00
	0.10	94.00	93.25	91.42	90.17	92.17	93.67	71.75	90.50	93.08	94.83	91.67	91.17	95.67	95.50	94.25
	0.15	79.50	81.33	72.83	74.00	75.83	74.92	45.00	72.08	77.17	81.67	73.33	74.42	85.92	86.33	83.42
	0.20	51.50	56.75	52.58	51.00	50.17	49.17	26.17	44.83	55.58	64.33	48.08	49.08	68.33	67.17	64.42
12	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.00	98.00	98.00	98.00	98.00	98.00	97.25	98.00	98.00	98.08	98.00	98.00	99.00	98.00	98.00
	0.10	95.00	94.58	93.25	91.42	93.33	94.17	77.08	93.33	93.92	96.00	92.67	92.25	96.50	96.08	95.50
	0.15	83.42	84.75	76.25	78.00	77.58	79.00	48.83	77.67	81.75	86.08	77.42	77.83	90.17	91.25	88.67
	0.20	58.42	62.33	57.67	57.08	54.33	53.83	28.92	56.08	63.42	71.75	54.58	54.17	74.42	76.08	71.42
15	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.00	98.00	98.00	98.00	98.00	98.00	97.50	98.00	98.00	98.25	98.00	98.00	99.00	98.25	98.25
	0.10	95.58	95.33	94.25	92.25	94.08	94.42	79.67	94.50	94.42	96.42	93.50	92.92	97.58	96.50	95.92
	0.15	85.75	86.50	79.67	79.58	80.33	81.42	51.75	81.08	85.08	89.67	80.00	80.75	92.75	93.00	91.33
	0.20	63.33	66.17	61.17	62.33	58.75	57.83	30.83	60.83	67.33	75.67	58.17	57.17	79.75	81.50	76.17

Table 3.11: Classification rates of GRHFM at $s = 0.5, 1, 2, 4$, non-orthogonal (ART, GFD, RM), Jacobi polynomial-based (ZM, PZM, OFMM, CHFM, PJFM), and eigenfunction-based (FBM, BFM, DHC) methods on NoiseInner dataset under different levels of Gaussian noise $\sigma^2 = 0.00, 0.05, 0.10, 0.15, 0.20$ and at different values of $K = 1, 2, 3, 4, 5$.

K	σ^2	GRHFM				ART	GFD	RM	ZM	PZM	OFMM	CHFM	PJFM	FBM	BFM	DHC
		s=0.5	s=1	s=2	s=4											
1	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	95.67	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	94.00	97.67	97.67	99.00	91.83	97.08	93.83	43.08	72.42	83.08	85.58	86.92	83.25	85.08	88.17
	0.10	79.08	83.00	91.83	98.92	75.08	84.08	72.17	28.00	50.92	63.08	69.33	61.25	64.58	62.92	66.75
	0.15	67.50	71.08	82.83	97.33	64.58	74.58	57.00	21.75	40.00	50.50	53.33	45.67	48.25	47.92	51.75
	0.20	56.17	61.75	69.42	94.83	56.58	67.17	47.83	18.83	34.00	40.17	45.00	38.00	39.25	36.92	42.17
2	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.25	99.75	100.00	100.00	100.00	100.00	100.00	93.00	98.08	100.00	99.58	99.75	99.25	99.25	100.00
	0.10	96.17	98.67	100.00	100.00	98.67	99.67	97.33	82.75	91.75	97.00	95.92	97.83	93.75	93.58	98.92
	0.15	92.25	94.67	98.42	100.00	94.50	97.67	93.25	73.50	84.75	92.25	88.00	92.00	88.67	88.42	95.33
	0.20	86.17	90.50	94.92	99.83	92.67	95.58	87.08	64.17	76.50	85.50	80.67	84.50	83.33	84.00	89.25
3	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.58	99.92	100.00	100.00	100.00	100.00	100.00	98.25	100.00	100.00	99.58	100.00	99.33	99.83	100.00
	0.10	98.50	98.92	100.00	100.00	99.00	100.00	100.00	95.00	99.33	100.00	98.83	99.00	98.25	99.00	99.92
	0.15	97.08	97.25	99.17	100.00	98.83	99.00	99.33	87.75	95.92	96.50	97.58	97.75	95.25	95.42	98.67
	0.20	93.92	95.33	98.58	99.92	97.75	98.08	97.58	80.25	91.25	91.83	93.33	95.50	93.25	91.33	96.58
4	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.67	99.83	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.67	100.00	100.00	100.00	100.00
	0.10	98.67	99.00	100.00	100.00	100.00	100.00	100.00	96.83	99.92	100.00	99.00	99.00	98.67	99.00	100.00
	0.15	97.92	97.67	100.00	100.00	99.00	99.00	99.75	93.67	98.42	98.67	98.00	98.33	97.25	97.83	100.00
	0.20	96.92	96.92	99.42	99.83	98.58	98.92	99.00	91.25	96.58	96.17	96.58	97.08	96.25	95.50	99.08
5	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.92	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.10	98.92	99.00	100.00	100.00	100.00	100.00	100.00	99.25	100.00	100.00	99.00	99.00	99.00	99.00	100.00
	0.15	98.00	98.00	100.00	100.00	99.00	99.17	99.83	97.25	99.33	99.00	98.00	98.75	97.75	98.33	100.00
	0.20	98.00	97.17	99.58	99.92	98.75	99.00	99.00	94.00	98.58	97.33	97.42	97.00	97.00	96.50	99.92

Table 3.12: Classification rates of GRHFM at $s = 0.5, 1, 2, 4$, non-orthogonal (ART, GFD, RM), Jacobi polynomial-based (ZM, PZM, OFMM, CHFM, PJFM), and eigenfunction-based (FBM, BFM, DHC) methods on NoiseOuter dataset under different levels of Gaussian noise $\sigma^2 = 0.00, 0.05, 0.10, 0.15, 0.20$ and at different values of $K = 2, 4, 6, 8, 10$.

K	σ^2	GRHFM				ART	GFD	RM	ZM	PZM	OFMM	CHFM	PJFM	FBM	BFM	DHC
		s=0.5	s=1	s=2	s=4											
2	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	97.42	98.08	96.08	94.08	91.42	96.08	70.75	59.50	87.17	93.00	93.83	93.67	98.08	95.25	96.33
	0.10	92.92	91.00	82.58	79.08	74.00	81.33	37.00	21.42	62.17	76.42	79.17	81.42	89.33	85.17	84.00
	0.15	82.75	78.33	65.25	52.92	52.75	63.08	21.58	9.83	41.58	64.33	64.17	63.58	76.08	73.00	61.75
	0.20	67.92	65.25	52.25	34.92	32.50	46.00	15.08	7.42	28.67	50.83	51.67	51.33	58.25	57.08	46.92
4	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.42	99.00	98.00	97.92	97.92	98.00	91.83	92.58	98.00	98.00	98.25	98.00	99.67	98.17	98.00
	0.10	96.42	95.08	90.83	88.58	87.58	91.33	66.00	77.08	91.67	92.92	92.83	92.92	96.92	95.83	95.17
	0.15	92.33	89.33	79.33	71.67	73.17	77.58	37.75	57.17	74.08	81.50	84.08	83.17	94.00	88.92	86.25
	0.20	82.50	79.42	67.33	52.00	56.75	58.42	20.50	38.17	60.75	73.75	70.08	68.58	86.58	79.92	76.25
6	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	98.92	99.00	98.00	98.00	98.00	98.00	95.08	98.00	98.08	98.25	99.00	98.00	100.00	99.08	99.67
	0.10	98.00	97.58	94.67	89.92	92.58	94.08	74.08	89.42	94.83	96.92	94.00	94.25	98.67	97.92	97.00
	0.15	93.08	92.00	85.08	79.42	81.67	84.42	47.42	75.83	87.33	88.00	88.00	88.00	95.50	93.75	91.33
	0.20	88.17	84.58	73.42	61.25	63.33	65.00	30.58	57.33	76.00	79.25	76.92	77.17	91.92	88.08	84.50
8	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.25	99.00	98.00	98.00	98.00	98.00	96.17	98.00	99.17	99.00	99.00	98.00	100.00	100.00	100.00
	0.10	98.00	97.92	96.50	90.58	94.25	95.75	81.42	93.25	96.50	97.33	96.17	95.58	98.92	98.00	97.75
	0.15	94.08	92.08	87.33	82.92	85.25	86.00	52.42	84.42	89.42	89.50	90.08	89.75	96.67	94.67	93.50
	0.20	88.92	87.08	77.50	67.50	69.50	68.83	35.25	71.33	82.17	83.92	80.67	81.33	93.17	90.67	87.42
10	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	0.05	99.42	99.00	98.00	98.00	98.00	98.00	97.50	98.33	99.83	99.42	99.00	98.00	100.00	100.00	100.00
	0.10	98.00	97.92	97.17	91.00	95.17	96.67	84.33	96.17	97.42	97.83	97.58	97.42	99.00	98.00	98.08
	0.15	94.50	92.17	88.50	84.17	85.92	87.58	56.08	87.50	91.33	92.17	91.25	90.67	97.17	95.17	94.50
	0.20	89.67	88.67	79.92	72.33	73.67	72.17	38.83	77.08	84.17	86.42	84.75	82.50	94.75	92.75	88.58

Chapter 4

Sparse Representation for Image Analysis and Recognition

Contents

4.1 Sparse modeling of signals/images

This section presents some basics of a sparse signal/image modeling framework. It is started with a mathematical formulation of sparse representation, then by a justification for the use of ℓ_1 regularization in the framework. The Bayesian interpretation of the framework is also given along with different strategies for the design of dictionaries. All these aspects are followed by a sketch of contributions that will be presented in this chapter.

4.1.1 Mathematical formulation

Let f be an input image of size $w \times h$ which is cast as a vector $\mathbf{x} \in \mathbb{R}^m$ ($m = wh$) by stacking its columns; let $\mathbf{D} \in \mathbb{R}^{m \times p}$ be an overcomplete dictionary with $m < p$. The linear system of equations $\mathbf{D}\boldsymbol{\alpha} = \mathbf{x}$ with $\boldsymbol{\alpha} \in \mathbb{R}^p$ is under-determined since it has more unknowns than equations. This system of equations has either no solution, if $\mathbf{x}$ is not in the span of the columns of $\mathbf{D}$, or otherwise infinitely many solutions. Assuming that $\mathbf{D}$ is a full-rank matrix, i.e., its columns span the entire space $\mathbb{R}^m$, additional criteria are needed if a well-defined solution $\boldsymbol{\alpha}$ is desired. The common approach is to introduce a *regularization term* J in order to give preference to a particular solution that has desirable properties, with a smaller value mean a better solution. The preferred solution $\boldsymbol{\alpha}$ could then be obtained by solving the following optimization problem:

$$(P_J): \qquad \min_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) \qquad \text{subject to} \qquad \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}.$$

When J is a strictly convex function, it is well-known that (P_J) has a unique solution. In the literature, the most common choice for J is the squared Euclidean norm defined by $J(\boldsymbol{\alpha}) = J_2(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|_2^2$. (P_J) then becomes (P_2) as

$$(P_2): \qquad \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_2^2 \qquad \text{subject to} \qquad \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}.$$

(P_2) defined as above is in fact a variant of the Tikhonov regularization problem [218], it has a unique closed-form solution $\hat{\boldsymbol{\alpha}}_2$, the so-called minimum-norm solution, which is given explicitly by

$$\hat{\boldsymbol{\alpha}}_2 = \mathbf{D}^T\left(\mathbf{D}\mathbf{D}^T\right)^{-1}\mathbf{x}.$$

Due to the above closed-form and unique solution $\hat{\boldsymbol{\alpha}}_2$, J_2 has been used extensively in various fields of applications. The interpretation here is that $\hat{\boldsymbol{\alpha}}_2$ has the smallest energy among all solutions of (P_2) since the squared ℓ_2-norm is a *measure of energy*. However, in many cases, the smallest energy is a misleading notion that calls for more appropriate choices for J. In image processing, it is now well-established that J should be a *measure of sparsity* in order to promote sparsity in the solution of (P_J). The simplest and most intuitive measure of sparsity of a vector $\boldsymbol{\alpha}$ is the number of non-zero elements in $\boldsymbol{\alpha}$ defined by the ℓ_0-norm as[6]

$$\|\boldsymbol{\alpha}\|_0 = \#\{j : \alpha_j \neq 0\}. \tag{4.1}$$

A vector $\boldsymbol{\alpha}$ is often said to be sparse if there are "few" non-zero elements in $\boldsymbol{\alpha}$, or $\|\boldsymbol{\alpha}\|_0 \ll p$.

Consider now the problem (P_0) obtained from (P_J) by using $J(\boldsymbol{\alpha}) = J_0(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|_0$. Finding the sparse solution $\hat{\boldsymbol{\alpha}}_0$ of (P_0) is equivalent to solving the following ℓ_0 optimization problem:

$$(P_0): \qquad \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \qquad \text{subject to} \qquad \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}.$$

[6] ℓ_0-norm is in fact not a norm as defined in mathematics because it does not satisfy the positive homogeneity and triangle inequality conditions. Nevertheless, for the sake of convenience, the term norm is used for this function.

Superficially, the sparsity-minimizing problem (P_0) looks like the energy-minimizing problem (P_2) but there are some startling differences between them. The solution $\hat{\boldsymbol{\alpha}}_2$ of (P_2) is always unique and can be obtained easily by using now-standard tools from computational linear algebra. Solving (P_0) for $\|\boldsymbol{\alpha}\|_0$, however, poses many challenges that have roots in the discrete and discontinuous nature of the ℓ_0-norm. (P_0) is a classical problem of combinatorial search with exhaustive search complexity is exponential in p. Indeed, it was proven that (P_0) is, in general, NP-hard [160]. Suboptimal solutions of (P_0) can be found by *greedy algorithms* like matching pursuit (MP) [146], orthogonal matching pursuit (OMP) [172], or *convex relaxation techniques* like FOCUSS [88], basis pursuit (BP) [41]. As an example, BP proposes to use $J(\boldsymbol{\alpha}) = J_1(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|_1$ and solve the following ℓ_1 optimization problem:

$$(P_1): \qquad \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \qquad \text{subject to} \qquad \mathbf{x} = \mathbf{D}\boldsymbol{\alpha}.$$

This problem is intermediate between (P_0) and (P_2): it is a convex problem that is, in some sense, closest to (P_0). Interestingly, results in [59] show that if the solution $\hat{\boldsymbol{\alpha}}_0$ of (P_0) is sufficiently sparse, it is equal to the solution $\hat{\boldsymbol{\alpha}}_1$ of (P_1). Moreover, it was also proven that the solution of (P_1) is more stable than that of (P_0) in the sense that small variations in $\mathbf{x}$ result in more similar active sets (i.e., the selected atoms from $\mathbf{D}$).

When the condition $\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}$ is relaxed by $\mathbf{x} = \mathbf{D}\boldsymbol{\alpha} + \mathbf{z}$, where $\mathbf{z} \in \mathbb{R}^p$ is a noise term with $\|\mathbf{z}\|_2^2 \leq \epsilon$ to account for the possible inclusion of small dense noise in the input signal or to allow small error in the representation, (P_1) now transforms into the following basis pursuit denoising problem (BPDN):

$$(P_1^\epsilon): \qquad \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \qquad \text{subject to} \qquad \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \leq \epsilon. \tag{4.2}$$

(P_1^ϵ) is a convex optimization problem, its solution can be found by minimizing the corresponding Lagrangian function as

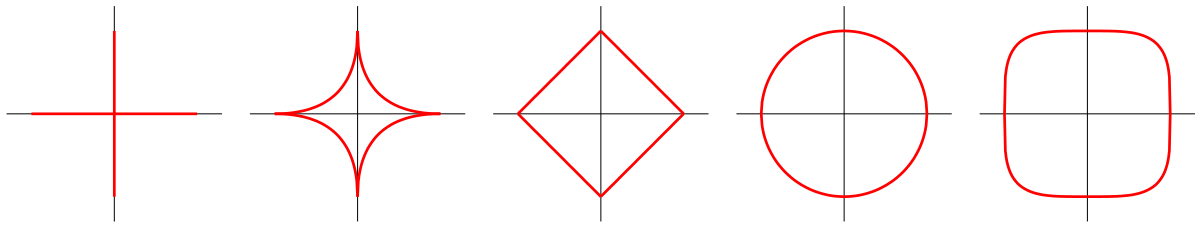
$$(Q_1^\lambda): \qquad \min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{4.3}$$

where the parameter λ is the Lagrange multiplier that depends on ϵ, it balances the sparseness in $\boldsymbol{\alpha}$ and the reconstruction error in $\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}$. In statistics, (Q_1^λ) is known as the LASSO problem [217] and has been well-studied by optimization specialists, leading to many practical methods for solving it. For large-scale applications, the following special purpose optimizers are frequently used: iterative reweighed least square [48], iterative shrinkage-thresholding [16], and least angle regression [69]. Nevertheless, when the dictionary $\mathbf{D}$ is not derived from an analytical transform that has fast implementation, due to high computational complexity, small image patches of size below $w \times h = 32 \times 32$ are usually used in the optimization process, instead of the whole image.

It should be noted that, sometimes, a more generic form of (Q_1^λ) defined as

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} f(\boldsymbol{\alpha}) + \lambda\Omega(\boldsymbol{\alpha})$$

is also used, where $f : \mathbb{R}^p \to \mathbb{R}$ is a convex differentiable function that describes the fidelity of the representation $\mathbf{D}\boldsymbol{\alpha}$ to the empirical data $\mathbf{x}$ and $\Omega : \mathbb{R}^p \to \mathbb{R}$ is a sparsity-promoting function which is typically non-smooth and non-Euclidean. However, the more flexibility in the above formulation has to be paid by the more complexity in the algorithm that solves it. In the remaining of this chapter, unless explicitly specified as in Section 4.4, the sparse representation is obtained as the solution of (Q_1^λ).



(a) $q = 0$ (b) $q = 0.5$ (c) $q = 1$ (d) $q = 2$ (e) $q = 4$

Figure 4.1: Illustration of the level sets (contours of constant value) of $|\alpha_1|^q + |\alpha_2|^q$ for some selected values of q. The open ball of ℓ_q-norm is convex when $q \geq 1$, strictly convex when $q > 1$, and concave when $q < 1$.

4.1.2 The ℓ_1 regularization

In order to justify the choice of ℓ_1-norm for J, consider the following problem:

$$(P_q^\epsilon): \qquad \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_q^q \qquad \text{subject to} \qquad \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \leq \epsilon$$

where

$$\|\boldsymbol{\alpha}\|_q = \left(\sum_{j=1}^{m} |\alpha_j|^q \right)^{1/q}, \qquad (q > 0)$$

is the ℓ_q-norm[7] and $\|\boldsymbol{\alpha}\|_0$ is defined as in Eq. (4.1). It is straightforward that $\|\boldsymbol{\alpha}\|_q^q$ is convex when $q \geq 1$, strictly convex when $q > 1$, and concave when $q < 1$. Illustration of the convexity of $\|\boldsymbol{\alpha}\|_q^q$ for the case $p = 2$ is shown in Fig. 4.1 for some selected values of q. In addition, the constraint $\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \leq \epsilon$ defines a feasible set centered at the least square estimate $\boldsymbol{\alpha}_{\text{LS}}$. When this set does not contain the origin, the solution of (P_q^ϵ) should be sought at the set's boundary.

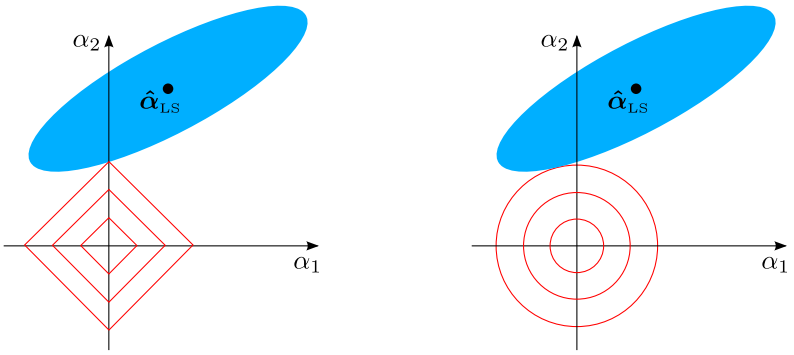
Now consider the set

$$\mathcal{S}_q^k = \{\boldsymbol{\alpha} : \|\boldsymbol{\alpha}\|_q^q \leq k \in \mathbb{R}\}.$$

$\mathcal{S}_q^k$ is a "ball" containing all the vectors that have length (in terms of ℓ_q-norm) less than or equal to $\sqrt[q]{k}$. By geometric intuition and at a specific value of q, $\mathcal{S}_q^0$ contains only the origin and an increase in the value of k is equivalent to an increase in the size of the ball. Since solution of (P_q^ϵ) exists if and only if $\mathcal{S}_q^k$ intersects the feasible set, solving (P_q^ϵ) could be done by "blowing" the ball $\mathcal{S}_q^k$ until it first touches the feasible set. The characterization of such a tangency point is determined by the "shape" of $\mathcal{S}_q^k$, which is in turn determined by the value of q. When $q \leq 1$, the concavity ($q < 1$) or non-strict convexity ($q = 1$) of $\mathcal{S}_q^k$ makes the tangency point lie at one of the ball's corners. Since corners usually take place on the axes, some coordinates of the tangency point are zeros or, in other words, the set of coordinates is likely to be sparse. On the contrary, when $q > 1$, the strict convexity of $\mathcal{S}_q^k$ does not require the tangency point to be at the ball's corners, meaning that the set of coordinates is not sparse.

Fig. 4.2 depicts the solution of (P_q^ϵ) at $q = 1, 2$ for the case $p = 2$. The feasible sets are plotted as shaded cyan regions and have elliptical contours centered at $\boldsymbol{\alpha}_{\text{LS}}$. The target functions

[7] Note that, when $q < 1$, ℓ_q-norm is not a norm because it does not satisfy the triangle inequality condition. Nevertheless, for the sake of convenience, the term norm is used for this function.



(a) $q = 1$ (b) $q = 2$

Figure 4.2: Illustration of the solution of (P_q^ϵ) for $q = 1$ (left) and $q = 2$ (right) for the case $p = 2$. Shown in the figure are the feasible sets and level sets of the target functions: the shaded cyan regions are the constraint regions $\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \leq \epsilon$, while the red contours are the level sets of $|\alpha_1| + |\alpha_2|$ (left) and $\alpha_1^2 + \alpha_2^2$ (right).

are $|\alpha_1| + |\alpha_2|$ and $\alpha_1^2 + \alpha_2^2$ for $q = 1$ and $q = 2$ respectively. The solutions are defined as the points where the level sets of these target functions, diamonds and disks respectively, hit the elliptical contours during the blowing process. Unlike the disk, the diamond has corners and if the solution occurs at a corner, it has one coordinate equal to zero. When $p > 2$, the diamond becomes a rhomboid that has many corners, flat edges and faces. Accordingly, there are many more opportunities for the solution's coordinates to be zero.

Putting things together, it is clear that, among $\|\boldsymbol{\alpha}\|_q^q$, $\|\boldsymbol{\alpha}\|_1$ is the only norm that has both properties: sparsity-promoting and convex. While sparsity is the target of the representation, convexity is the preferred property for practical implementation. Moreover, from the above analysis, it can be seen that sparsity-promotion and convexity are the two contradicting properties: a convex set $\mathcal{S}_q^k$ is likely to result in a non-sparse solution, and vice versa. Note that this observation is valid not only for ℓ_q regularization but also for other regularization functions. The only compromise between sparsity-promotion and convexity occurs in the case $J_1(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|_1$ and this justifies for the popular use of ℓ_1-norm as the sparse regularization.

4.1.3 Bayesian interpretation

Besides the intuitive interpretation of obtaining a sparse representation that, at the same time, minimizes the reconstruction error, the problem (Q_1^λ) in Eq. (4.3) has an equivalence in Bayesian decision framework [232]. Assuming that $\mathbf{x}$ is generated by the following model:

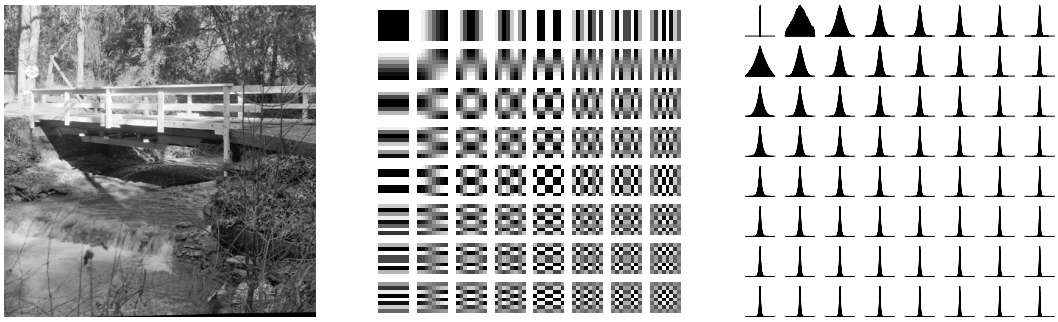
$$\mathbf{x} = \mathbf{D}\boldsymbol{\alpha} + \mathbf{z}, \tag{4.4}$$

where $\mathbf{z}$ is the additive white Gaussian noise (AWGN) distributed as

$$p(\mathbf{x}|\boldsymbol{\alpha}; \sigma) = \left(2\pi\sigma^2\right)^{-\frac{m}{2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2\right).$$

The prior distribution of $\boldsymbol{\alpha}$ is also assumed to be the generalized Gaussian distribution:

$$p(\boldsymbol{\alpha}; \beta) = \frac{q}{2\beta\Gamma\left(\frac{1}{q}\right)} \exp\left(-\sum_{j=1}^{p}\left|\frac{\alpha_j}{\beta}\right|^q\right),$$



(a) Image "stream" (b) 8×8 DCT dictionary (c) Coefficient distributions

Figure 4.3: Distributions of the coefficients of the 512×512 image "stream" (a) using a standard 8×8 DCT dictionary (b). Empirical distributions of the coefficients (c) associated to 64 DCT atoms have heavy tails that are similar to those of the Laplacian distribution.

where $\Gamma(\cdot)$ denotes the Gamma function. When $q \in [0, 1]$, this prior is known to encourage sparsity in many situations because of its heavy tails and a sharp peak at zero. Given this prior, the maximum *a posteriori* (MAP) solution to Eq. (4.4) is formulated as

$$\begin{aligned} \boldsymbol{\alpha}_{\text{MAP}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}}\, p(\boldsymbol{\alpha}|\mathbf{x}; \sigma, \beta) &= \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \big(-\log p(\mathbf{x}|\boldsymbol{\alpha}; \sigma) - \log p(\boldsymbol{\alpha}; \beta) \big) \\ &= \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left(\frac{1}{2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_q^q \right), \end{aligned} \tag{4.5}$$

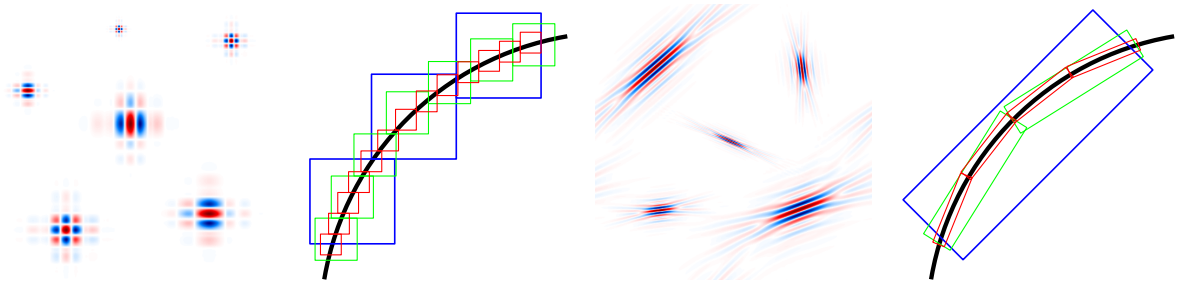
with $\lambda = \frac{\sigma^2}{\beta^q}$. It is clear that when $q = 1$, the generalized Gaussian distribution becomes the Laplacian distribution and the above optimization problem is equivalent to (Q_1^λ). In the literature, Laplacian distribution has been the dominant choice that balances simplicity in the model and fidelity to the empirical data [125]. As an experimental evidence, Fig. 4.3c shows the histograms of the DCT coefficients using a standard 8×8 DCT dictionary given in Fig. 4.3b. The image used in the experiment is the popular 512×512 "stream" picture shown in Fig. 4.3a. Experimental results like in Fig. 4.3c indicate that the histograms resemble Laplacian distributions when the Kolmogorov–Smirnov goodness-of-fit test is used [186].

4.1.4 Dictionary design

The ability of (Q_1^λ) in Eq. (4.3) to guarantee a good representation and a sparse solution depends not only on the signal or the algorithm that solves it, but also on the overcomplete dictionary. For a certain algorithm used on a specific class of signals, it is observed that different dictionaries lead to different representation performance and/or sparsity in the solution. There exist dictionaries that more likely lead to sparse solutions than the others because they contain atoms that explain better the underlying phenomena in the signals. Dictionary design thus deals with the problem of finding "optimal" dictionaries for different classes of signals in different applications. In the literature, there are two main classes of dictionaries: one is based on analytical transforms and the other is based on learning methods in order to adapt dictionaries to signals.

Analytical dictionaries

Dictionaries defined based on analytical transforms have pre-defined analytical atoms which are, in general, inflexible in representing signals. However, one of the main advantages of analytical



(a) Wavelets (b) Wavelet alignment (c) Curvelets (d) Curvelet alignment

Figure 4.4: Some undecimated wavelets/curvelets and their alignment with a contour. Wavelets have square supports and require many coefficients to capture a smooth contour. Curvelets have elongated supports and can effectively represent a smooth contour with fewer coefficients.

dictionaries over learned ones that partially explains for their existence is that they are usually accompanied by fast implicit implementations. In addition, developments in computational harmonic analysis within the last decade have led to the proposal of a number of transforms, each of which gives sparse representation to a certain class of signals. Notable examples are the curvelet transform and the undecimated wavelet transform: the first allows an almost optimal sparse representation of components with singularities along smooth curves whereas the second gives sparse representation to textual components. Since these two transforms are going to be used in this chapter, a brief review on their characteristics is given below. Some samples of undecimated wavelets and curvelets and their alignment with a contour are depicted in Fig. 4.4.

Undecimated wavelet transform (UWT) is the undecimated version of the orthogonal wavelet transform (OWT) obtained by skipping the decimation step. It is designed to overcome the lack of shift-invariance property in OWT. UWT can be represented as a transformation matrix which has more columns than rows. The redundancy factor (i.e., the ratio between the number of columns to the number of rows) is $3J + 1$, where J is the number of scales in the decomposition. UWT is expected to give sparse representation to isotropic features and non-sparse representation to highly anisotropic features. The "à trous" algorithm [201] provides an efficient way to implement forward and inverse UWT.

Curvelet transform is an extension of the wavelet transform to represent images which are smooth apart from singularities along smooth curves, similar to the interpretation that wavelet transform is an extension of the Fourier transform to represent 1D piece-wise smooth signals with a finite number of discontinuities. It has been shown that curvelets constructed as in [33] are multi-scale, multi-directional, and elongated. They define a tight frame in $L^2\left(\mathbb{R}^2\right)$, obey the parabolic scaling relation ($width = length^2$), and exhibit an oscillating behavior in the direction perpendicular to their orientation. A curvelet frame can be used as an overcomplete dictionary with a redundancy factor of $16J + 1$. The curvelet transform is expected to give sparse representation to anisotropic structures, and smooth curves and edges of different lengths.

Learned dictionaries

Dictionary learning for natural images under the sparsity assumption [164] aims at maximizing the likelihood (ML) that natural images have efficient, sparse representations in an overcomplete

dictionary. Mathematically speaking, for a given signal $\mathbf{x}$, the goal of learning is to find the overcomplete dictionary $\mathbf{D}^*$ such that

$$\begin{aligned}\mathbf{D}^* &= \underset{\mathbf{D}}{\operatorname{argmax}} \log p(\mathbf{x}|\mathbf{D}) \\ &= \underset{\mathbf{D}}{\operatorname{argmax}} \log \int_{\boldsymbol{\alpha}} p(\mathbf{x}|\mathbf{D},\boldsymbol{\alpha}) p(\boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}.\end{aligned} \tag{4.6}$$

In order to solve the above difficult optimization problem, two main assumptions were introduced. First, the coefficients α_i are independent and each has a Laplacian distribution, which nicely fits the probability distribution of α_i when the decomposition is sparse. The second assumption is that the reconstruction error $\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}$ can be modeled as the additive white Gaussian noise. Under these two assumptions, the problem in Eq. (4.6) transforms into the following energy minimization problem:

$$\begin{aligned}\mathbf{D}^* &= \underset{\mathbf{D},\boldsymbol{\alpha}}{\operatorname{argmin}} E(\mathbf{x},\boldsymbol{\alpha}|\mathbf{D}) \\ &= \underset{\mathbf{D},\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1,\end{aligned}$$

where the energy function is defined as $E(\mathbf{x},\boldsymbol{\alpha}|\mathbf{D}) = -\log p(\mathbf{x}|\mathbf{D},\boldsymbol{\alpha})p(\boldsymbol{\alpha})$. When a set of n signals $\mathbf{X} = [\mathbf{x}^1, \cdots, \mathbf{x}^n]$ is used to learn the dictionary $\mathbf{D}$, the above problem becomes

$$(D_1^\lambda): \qquad \min_{\mathbf{D},\mathbf{A}} \frac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \sum_{i=1}^{n} \lambda\|\boldsymbol{\alpha}^i\|_1, \tag{4.7}$$

where $\mathbf{A} = [\boldsymbol{\alpha}^1, \cdots, \boldsymbol{\alpha}^n]$ and $\|\cdot\|_F$ denotes the Frobenius norm which has the following definition:

$$\|\mathbf{A}\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |\alpha_{ij}|^2\right)^{1/2}.$$

Note that it is necessary to bound the columns $\mathbf{d}_i$ $(i = 1, \ldots, p)$ of the dictionary $\mathbf{D}$ such as $\|\mathbf{d}_i\|_2^2 = 1$ to prevent $\mathbf{D}$ from being arbitrarily large. This is because the term $\mathbf{DA}$ is invariant by multiplying $\mathbf{D}$ by a diagonal matrix on the right and then multiplying $\mathbf{A}$ by its inverse on the left. Moreover, the dictionary learning problem (D_1^λ) in Eq. (4.7) could also be viewed as a matrix factorization problem using ℓ_1 regularization where the matrix $\mathbf{X}$ is factored into the two matrices $\mathbf{D}$ and $\mathbf{A}$. In the literature, (D_1^λ) is usually solved by alternating between two steps:

- *Sparse coding*: $\mathbf{D}$ is kept constant, the energy function is minimized with respect to $\mathbf{A}$.
- *Dictionary update*: $\mathbf{A}$ is kept constant, the energy function is minimized with respect to $\mathbf{D}$.

The algorithm alternates between sparse coding and dictionary update until convergence. Different dictionary learning methods use different strategies to perform these two steps, of which the sparse coding step is essentially the problem (Q_1^λ) in Eq. (4.3). For example, the original method [164] uses convex optimization for sparse coding and gradient descent for dictionary update. The method of optimal direction (MOD) [74] uses OMP for space coding and introduces a closed-form solution for dictionary update. The majorization method is used [237] to minimize the energy function in both sparse coding and dictionary update steps.

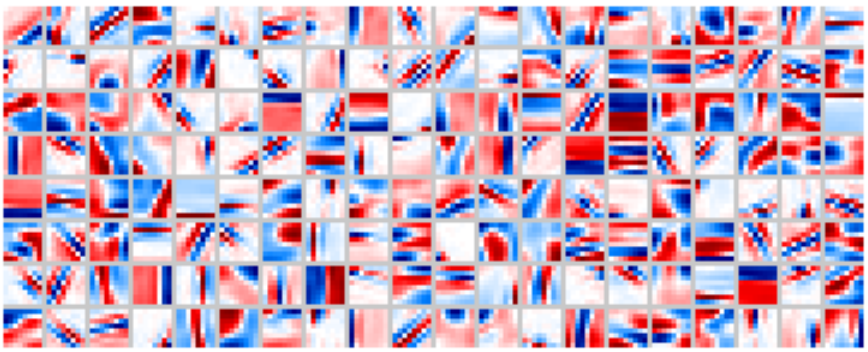


Figure 4.5: The overcomplete dictionary of 8×8 atoms learned from the image "stream" in Fig. 4.3a. The 512×512 image is used to generate a large dataset of 255025 image patches of size 8×8. The parameter λ has a value of 0.1 in the learning process.

A fast online learning algorithm was recently proposed [142]. The method is different from the others in the sense that it does not use the whole training data at each iteration. Instead, it uses a subset of the training data for the optimization problem (D_1^λ) in Eq. (4.7) and then augments the subset with new training samples to compute a new solution using the previous solution as initialization. The online algorithm iterates until all training data are used. This strategy thus alleviates the expensive computation when the training data is large and is appropriate for dynamic systems where data evolve over time.

Another family of dictionary learning techniques is based on vector quantization (VQ) by means of K-means clustering. The VQ approach for dictionary learning was first proposed [197] for MP-based video coding using the implicit assumption that each image patch can be represented by a single atom with the corresponding coefficient equal to one. This assumption reduces the learning procedure to K-means clustering. Dictionary update is done by first grouping patches to minimize their distance to a given atom, and then by updating the atom to minimize the overall distance in the group of patches. A generalization of the K-means for dictionary learning, called the K-SVD algorithm, was proposed [2] by not using the single-atom assumption. The method instead uses OMP for sparse coding and singular value decomposition (SVD) for updating each atom sequentially. The update step is thus a generalized K-means algorithm since each patch can be represented by a set of atoms with different weights.

The overcomplete dictionary of 8×8 atoms learned from the image "stream" in Fig. 4.3a is given in Fig. 4.5. It can be observed that the learned dictionary contains many atoms that are localized, oriented, and bandpass. This type of atoms represents well the oriented edges in images. In addition, the dictionary also consists of some atoms that are center-surround and grating, which better approximate textures in images. Dictionary learning thus results in atoms which identify the most important building blocks in natural images. As a result, learned dictionaries usually lead to state-of-the-art results in many practical signal processing applications.

4.1.5 Contributions

In employing sparse modeling in some image processing and classification tasks, this chapter makes the following main contributions:

- *Denoising*: It uses the synthesis operator of curvelet transform as the dictionary of a sparse representation framework for directional denoising. It demonstrates both theoretically and experimentally that the information about the level of edge noise has a linear relationship

with the only framework's parameter. It shows that the proposed sparsity-based denoising method leads to superior performance over comparison methods on edge noise removal in bilevel graphical document images.

- *Separation*: It applies an existing sparsity-based separation technique using two appropriately chosen discriminative overcomplete dictionaries, each gives sparse representation over one type of images and non-sparse representation over the other, for the classical problem of extracting text components from graphical document images. It proposes some heuristic rules to group text components into text strings in the post-processing step. It shows experimentally that the proposed sparsity-based text extraction method leads to better performance than the current benchmark.

- *Classification*: It proposes a new discriminative sparse coding method by adding a discriminative term to the conventional sparse representation framework, resulting in a model that is a controlled trade-off between sparsity, fidelity to the data, and discrimination power. It uses an information theoretic-based criterion, called minimum message length, to select the optimal statistical model. It shows that the proposed method leads to superior classification performance over comparison methods on the two common handwritten and face datasets.

The remainder of this chapter is organized as follows. The directional denoising framework for edge noise removal is presented in Section 4.2. Section 4.3 discusses the sparsity-based extraction of text components from graphical document images. The MML-based discriminative sparse modeling is introduced in Section 4.4. Finally conclusions are drawn in Section 4.5.

4.2 Graphical document image denoising

The scanning and binarization processes that produce binary document images introduce noise that concentrates on the edges of the image objects [14]. This edge noise has influence on later steps in the chain of automatic document processing. It could affect feature measurement in recognition, reduce image redundancy in compression, and distort skeletons in vectorization. For accurate analysis and recognition of document images, edge noise needs to be removed. This section proposes to use a sparsity-based method, which relies on an image degradation model, to remove noise in graphical document images. The relationship between the approach's parameter and the degradation model's parameter is investigated and the performance of the proposed approach is compared with those of existing methods on carefully designed datasets.

4.2.1 Image degradation model

A scanner model which is usually based on the physics in the document acquisition process provides a theoretical platform for the analysis of that process. The scanner model that is described below is a portion of the model presented in [9] and is schematically described in Fig. 4.6. It is assumed in this model that when a spatially continuous image o is converted to digital form using either a digital camera or a document scanner, the value of each pixel in the scanned image before quantization, $s[i, j]$, depends on the light collected at the corresponding discrete sensor. This collected light in turn depends on the reflectance in the original image in the neighborhood around that sensor, that is a function of the optics and the sensors. The contribution of the source reflectance to the sensor value is usually described by a function of the distance from the sensor center, called the point spread function PSF. Thus, the signal value

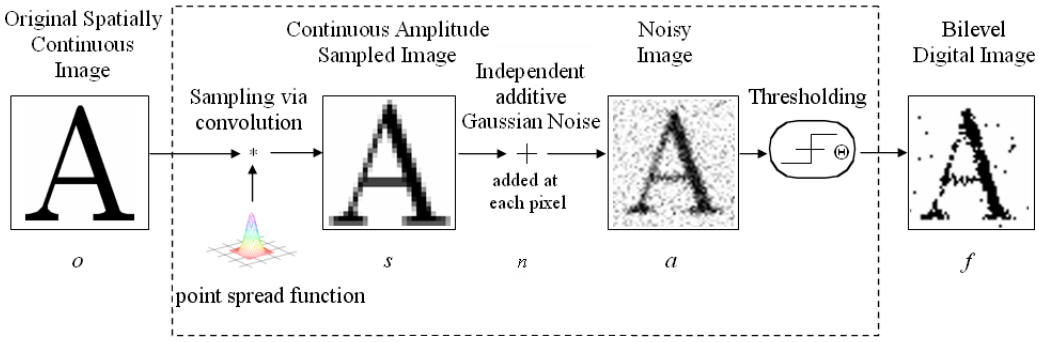


Figure 4.6: The scanner model used to determine the value of the pixel $[i, j]$ centered on each sensor element. It is modeled as a multi-stage process whose steps include convolving with a point spread function (PSF), sampling, adding noise, and thresholding.

that is received at each sensing element (i, j) is modeled as

$$s[i, j] = \iint \mathrm{PSF}(x_i - u, y_j - v, w)\, o(u, v)\, \mathrm{d}u\mathrm{d}v.$$

In order to model the noise that would occur during the acquisition process, it is generally assumed that additive white Gaussian noise (AWGN) n of standard deviation σ_{noise} is added to these values as

$$a[i, j] = s[i, j] + n[i, j].$$

Moreover, since document and graphical images are usually processed in bilevel, the noisy image is then thresholded, usually with a global threshold Θ, as

$$f[i, j] = \begin{cases} 1, & a[i, j] \geq \Theta \\ 0, & a[i, j] < \Theta. \end{cases}$$

While the AWGN is evenly distributed over the whole grayscale image, the effect of AWGN after thresholding is concentrated along the edges. The process of turning a smooth edge into a rough one and the analysis of this rough edge are discussed as follows.

Edge without noise

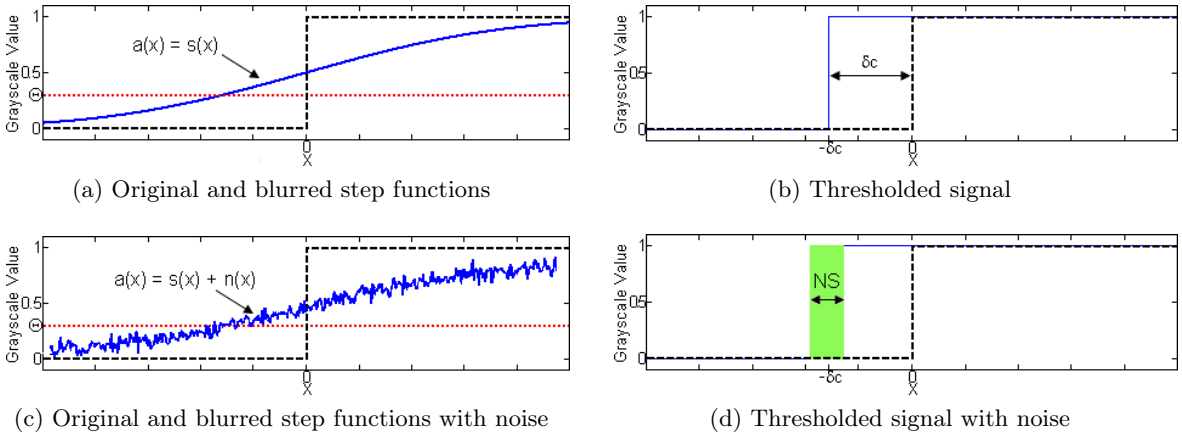
In graphical document images, the image content contains large regions of white (0) background, with foreground image features displayed in black (1). When documents are scanned in grayscale, the edges change from step functions to functions sloped in the shape of the edge spread function (ESF), which is the cumulative marginal of the PSF. The changes in edge functions in turn cause changes in the edge locations after thresholding [15]. For a PSF parameterized by a width w, the amplitude of the ESF is

$$s(x) = \mathrm{ESF}\left(\frac{x}{w}\right).$$

An example of an ESF is shown in Fig. 4.7a. When there is no noise, the edge location after thresholding occurs at the point where the amplitude is equal to the threshold Θ and would be at the location $x = -\delta_c$, where

$$\delta_c = -w\, \mathrm{ESF}^{-1}(\Theta).$$

The shift δ_c in edge location that depends on s and Θ in Fig. 4.7a is depicted in Fig. 4.7b.



(a) Original and blurred step functions

(b) Thresholded signal

(c) Original and blurred step functions with noise

(d) Thresholded signal with noise

Figure 4.7: (a) Edge after blurring with a generic PSF of width w. (b) When no noise is added, the thresholding produces the edge shift δ_c. (c) Blurred edge with added noise. (d) The uncertain boundary region, shown shaded, is the NS region.

Edge with noise and noise spread

When additive noise n is considered, the edge intensity after blurring a will fluctuate around s as illustrated in Fig. 4.7c. This fluctuation results in a small region near the edge of the step function in which the value of the pixel could be above or below the threshold. So, after thresholding, the thresholded signal has a noisy edge that may be at any position in that region. The breadth of that region is defined as noise spread (NS) (Fig. 4.7d). It has been shown recently [153] that NS is not just dependent on the standard deviation σ_{noise} of AWGN but also on the width w of PSF, which determines the slope of the grayscale edges, and the level of the binarization threshold Θ through the relation

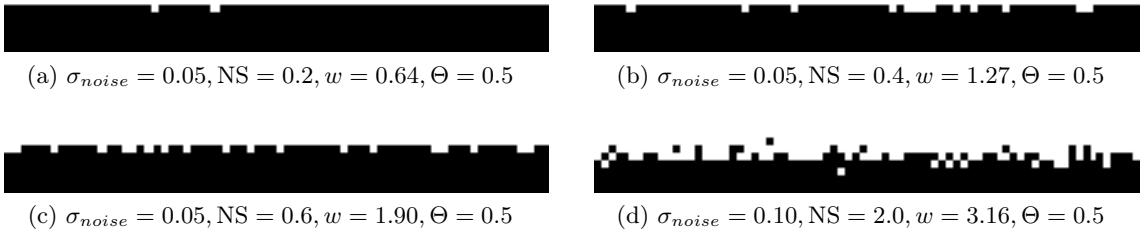
$$\mathrm{NS} = \frac{\sqrt{2\pi} \cdot \sigma_{noise} \cdot w}{\mathrm{LSF}(\mathrm{ESF}^{-1}(\Theta))},$$

where LSF is the line spread function, or one-dimensional PSF.

Some examples of edges with noise are shown in Fig. 4.8. The standard deviation σ_{noise} of the additive noise is kept fixed in the three images in Figs. 4.8a–4.8c. Images with a common σ_{noise} are conventionally thought of as having the same noise level. However, it can be seen that these three images have edges with distinctly different amounts of perceptual noise. At a constant value of σ_{noise}, an increase in NS make the image more noisier in an amount proportional to the increase in the value of NS. In Fig. 4.8d, a significantly larger NS is shown and its effect on the edge can be easily observed: there are generally two rows of pixels affected by the additive noise when NS = 2.0. In this manner, NS provides a measure that can numerically quantify the amount of edge noise and also relates to the noise visually observed on bilevel images.

Relationship between NS and Hamming distance

The real benefit of determining NS of a scanned object is that NS provides an effective measure of how noisy a bilevel object is. However, for two bilevel images of the same size, the Hamming distance [93], which is defined as the number of substitutions required to change one image into the other, is often used as a measure of difference between them. It is very useful and hence is usually used for the analysis of noise in bilevel images when the noise-free template is known.



(a) $\sigma_{noise} = 0.05, \text{NS} = 0.2, w = 0.64, \Theta = 0.5$ (b) $\sigma_{noise} = 0.05, \text{NS} = 0.4, w = 1.27, \Theta = 0.5$

(c) $\sigma_{noise} = 0.05, \text{NS} = 0.6, w = 1.90, \Theta = 0.5$ (d) $\sigma_{noise} = 0.10, \text{NS} = 2.0, w = 3.16, \Theta = 0.5$

Figure 4.8: Illustrations of edges with varying amounts of NS. (a)–(c) While the noise standard deviation is the same for each image, NSs are different. (d) NS qualitatively describes how many edge rows are affected.

The formula that relates the expected Hamming distance $E\{H\}$ to NS for straight edges was developed [153] to be

$$E\{H\} = \frac{\text{NS} \cdot \rho}{\pi}, \tag{4.8}$$

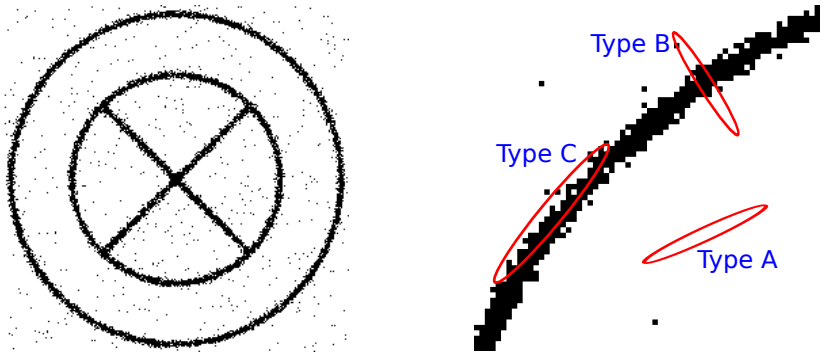
where ρ is a constant equal to the length of the edge segment. The above equation shows that Hamming distance is directly proportional to NS, leading to the possibility of using NS as a predictor of the Hamming distance between a degraded image and the predicted noise-free image, and vice versa. In addition to the theoretical result in Eq. (4.8), experiments have also been carried out to validate this linear relationship between NS and the expected Hamming distance.

The ability of NS in providing a quantitative measure that also qualitatively describes the amount of noise and its linear relationship with the Hamming distance lead to a potential that NS could be used as an input to a denoising algorithm that works on bilevel images, in a similar fashion to how the standard deviation of the AWGN is often used as an input to denoising algorithms that work on grayscale images.

4.2.2 Related works

Let the noisy image f be the result of scanning and then global thresholding a noise-free input image f_0 of size $w \times h$, it contains edge noise of a certain NS. Denoising f is viewed as an estimation problem, i.e., one needs to find an estimated image $\hat{f}$ from f which is close to f_0 and, at the same time, has some preferred properties like contour smoothness for graphical document images. Many methods exist for removing noise from digital images [30], each has its own properties that make it suitable for some particular situations. The aim of this subsection is thus not to give a long list of existing methods, but to review some relevant ones to the problem of noise removal from bilevel graphical document images based on the preferred criteria of image recovery and contour smoothness.

Bilevel image denoising: For bilevel document images, the most famous and frequently used denoising methods are contour smoothing using chain codes, median filtering, morphological operators, and kFill filtering. Contour smoothing based on chain codes [241] replaces a chain code sequence by a simpler sequence, usually a shorter one that corresponds to the shortest path. The simplicity in the code is enforced by minimizing the total change in the code sequence, which is in turn done by recursively replacing two consecutive changes by a single one. Even though this method can produce a smooth contour from a jagged one, it cannot be performed on an



(a) A noisy image (b) Contour alignment

Figure 4.9: Geometric illustration of directional denoising using curvelets: (a) an example noisy graphical document image; (b) three types of alignment of curvelets with a noisy graphics contour, only curvelets of type C capture the graphics contour and have significant magnitude.

image such as the one shown in Fig. 4.9a where the noisy pixels are not only distributed along the contours but also over the whole image region.

The main idea of median filtering [5] is to run through the image pixel by pixel, replacing each pixel's value with the median of neighboring pixels' values. Due to its nature, median filtering is particularly effective at removing outliers, such as "salt & pepper" noise or noise whose probability density has heavier tails than the Gaussian. Morphological operators [148] like erosion, dilation, and their combinations, i.e. opening and closing, have their root in set and lattice theories. The popularity and efficiency of the simple morphological openings and closings to suppress positive and negative impulse noise have theoretical supports. The kFill algorithm [163] is designed to remove "salt & pepper" noise iteratively while maintaining the readability of text by using a filter that retains text corners. The value of the parameter k of the kFill algorithm can be chosen adaptively based on text size and image resolution.

Among the aforementioned denoising methods, median filtering, morphological operators, and kFill filtering perform isotropically and geometrically local smoothing and thus may not be sufficient for denoising tasks that need directional smoothing or contour preservation. These methods are known to be unable to preserve fine image details and may unintentionally remove thin lines and corners. This is because they are general-purpose denoising methods which are not designed specifically for edge noise and do not exploit the directional information in their operations. Contours denoised by these methods are jagged and sometimes shifted from their original positions. Another shortcoming of existing binary image denoising methods is that they do not take into account the information about the level of noise that exists in the binarized document images. Denoising is performed in a "blind", non-adaptive way.

Total variation denoising: Noise removal by minimizing the total variation (TV)

$$\mathrm{TV}(f) = \int |\nabla f(\vec{x})| \, \mathrm{d}\vec{x} = \sum_{i,j} \sqrt{|f_{i+1,j} - f_{i,j}|^2 + |f_{i,j+1} - f_{i,j}|^2}$$

of an image f while preserving some "fit" to the original measured data was first proposed in [192]. The method is based on the principle that an image with excessive and possibly spurious details has a high TV, that is the integral of the absolute gradient of the image is high. According

to this principle, the problem of removing noise from a noisy image f based on TV could be posed as the following optimization problem:

$$\hat{f} = \underset{y}{\text{argmin}}\ \text{TV}(y) \qquad \text{subject to} \qquad \|y - f\|_2 \leq \epsilon_{\text{TV}}, \tag{4.9}$$

where the parameter ϵ_{TV}, which is related to the estimated noise level, determines the sharpness or smoothness of the restored image $\hat{f}$. It has been proven [208] that, for general texture images, this noise removal method has an edge-preserving property which is better than simple methods such as linear smoothing or median filtering, which reduce noise, but at the same time smooth away edges. However, the edge-preserving effects of TV regularization is somewhat local; that is, the effect on one edge in the image has little or no correlation with the effect on the others. This local property results in the inability of TV-based denoising methods to exploit the global long contours that exist in graphical document images in order to produce smooth ones.

Anisotropic diffusion: In image processing, anisotropic diffusion aims at suppressing image noise without removing significant parts of the image content. It is motivated by minimizing the energy functional of an image f defined by

$$E_f = \frac{1}{2}\int_{\Omega} g\left(||\nabla f(x)||^2\right)\,\mathrm{d}x,$$

where g is a real-valued function and the gradient descent is defined as

$$\frac{\partial f}{\partial t} = -\nabla E_f = \text{div}(g'\left(||\nabla f(x)||^2\right)\nabla f).$$

Thus by letting $c = g'$, the anisotropic diffusion equation becomes

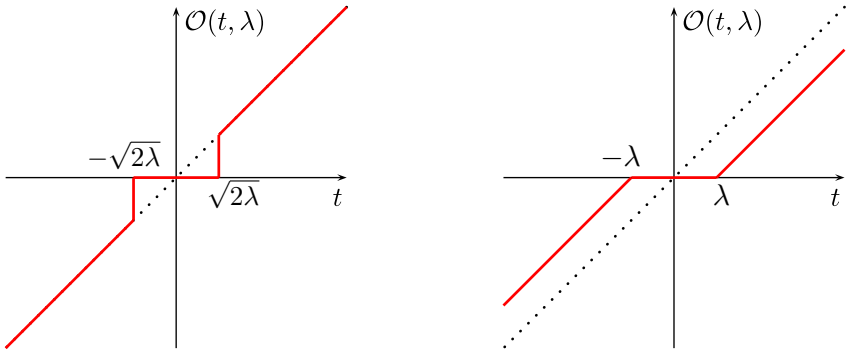
$$\frac{\partial f}{\partial t} = \text{div}\left(c(x,y,t)\nabla f\right) = \nabla c \cdot \nabla f + c(x,y,t)\Delta f,$$

where Δ denotes the Laplacian, ∇ denotes the gradient, and $\text{div}(\cdot)$ is the divergence operator.

In the original formulation [173], the diffusion coefficient $c(x,y,t)$ that controls the diffusion rate was proposed to be

$$c\left(||\nabla f||\right) = \mathrm{e}^{-(||\nabla f||/K)^2},\ \text{or}$$
$$c\left(||\nabla f||\right) = \frac{1}{1+\left(\frac{||\nabla f||}{K}\right)^2},$$

where K is a constant that controls the sensitivity to edges. The filter is in fact isotropic but depends on the image content in the way that it approximates an impulse function close to edges and other image's structures that need to be preserved over different levels of the resulting scale-space. A more general formulation allows the filter to adapt locally to be truly anisotropic near linear structures such as edges or lines. The filter has an orientation similar to that of the structure: it is elongated along the structure and narrow across. Such a method is referred to as *coherence enhancing diffusion* [231]. As a consequence, the resulting images preserve linear structures while at the same time smoothing is made along these structures.



(a) Hard-thresholding (b) Soft-thresholding

Figure 4.10: Illustrations of the hard-thresholding and soft-thresholding operators. $\mathcal{O}$ is a function of the input t for a fixed λ. The black dotted curves are the function $\mathcal{O}(\cdot, 0)$ (no regularization), whereas the red plain curves correspond to $\mathcal{O}(\cdot, \lambda)$.

Orthogonal wavelet denoising: Wavelet-based image denoising has been used widely and its success is due to the tendency of images to have a compact representation in the wavelet transform domain [182]. The efficiency of image approximation using a small subset of wavelets also led to the adoption of wavelet transform in JPEG-2000 image compression and coding systems. In denoising problems, it is legitimate to assume that only a few large wavelet coefficients contain information about the images while small coefficients are attributed to the noise. Thus, the common procedure in wavelet-based denoising methods is to first apply the discrete wavelet transform (DWT) (analysis operator $\mathbf{T}$) to the noisy image f, then use a nonlinear estimation rule $\mathcal{O}$ to the transform coefficients, and finally compute the inverse DWT (synthesis operator $\mathbf{T}^T = \mathbf{T}^{-1}$) to get an estimate $\hat{f}$ of the noisy image f. This procedure can be described symbolically as

$$\hat{f} = \mathbf{T}^T \mathcal{O}(\mathbf{T} f). \tag{4.10}$$

This approach was already proven to be very successful on both practical and theoretical sides [108]. Many thresholding or shrinkage rules were proposed for the operator $\mathcal{O}$ with hard-thresholding and soft-thresholding are certainly the most well-knowns. For the 1D variable t, these thresholding operators are defined as follows.

- *Hard-thresholding* [229] consists of setting to zero all coefficients whose magnitudes are less than a value (Fig. 4.10a):

$$\mathcal{O}(t,\lambda) = \begin{cases} t, & |t| \geq \sqrt{2\lambda} \\ 0, & \text{otherwise.} \end{cases} \tag{4.11}$$

- *Soft-thresholding* [61] is defined as the kill–or–shrink rule with the coefficients above a value are shrunk toward the origin (Fig. 4.10b):

$$\mathcal{O}(t,\lambda) = \text{sign}(t)(|t| - \lambda)^+, \tag{4.12}$$

where $(\cdot)^+ = \max(\cdot, 0)$.

Noise modeling: It can be easily seen that most of the denoising methods mentioned above are parametric and they require knowledge of noise to set their parameters in order to have good

performance. For example, TV denoising needs to set the value for ϵ_{TV}; multiscale denoising has to do the same for λ. The knowledge of noise is thus a crucial factor in applying denoising methods. In practical situation, the noise level is usually obtained from image noise models [23], which are built from the knowledge of noise generation processes or from some measured values. For the aforementioned edge noise which can be measured quantitatively, the availability of NS could pave the way for the design of a new method to remove edge noise.

The following subsection describes a new parametric method for edge noise removal in bilevel graphical document images that exploits the directional information of graphics contours. Information about the level of edge noise represented by NS is used as an input to the denoising process in order to have good performance. Directional denoising is facilitated by using a sparse representation framework. This is done by promoting sparse representation of graphical document images in an overcomplete dictionary using a basis pursuit denoising algorithm with curvelets as the dictionary. The images reconstructed from their sparse representations are grayscale ones, which can be simply thresholded to obtain the final bilevel denoised images.

4.2.3 Sparsity-based edge noise removal

Let $\mathbf{x}_0$, $\mathbf{x}$, and $\hat{\mathbf{x}} \in \mathbb{R}^p$ $(p = wh)$ be the vectors generated by stacking the columns of f_0, f, and $\hat{f}$ respectively then $\mathbf{x} = \mathbf{x}_0 + \mathbf{z}$, where $\mathbf{z} \in \mathbb{R}^p$ stands for the unknown additive edge noise. This subsection is devoted to the finding of $\hat{\mathbf{x}}$ from $\mathbf{x}$ by combining the recent ideas of directional representation and sparse representation in order to achieve the two preferred criteria of image recovery and contour smoothness. Contour smoothness is guaranteed by the use of a curvelet dictionary whereas image recovery is by a fidelity constraint in the sparsity framework.

Multiscale directional denoising

Although applications of wavelets in image processing have become increasingly popular, it is well-established that traditional wavelets are only good at representing point singularities since they ignore the geometric properties of structures and do not exploit the regularity of edges. The images denoised by using traditional wavelets usually have unfavorable blocky artifacts. For these reasons, wavelet-based denoising becomes inefficient for geometric line-like features and surface singularities.

To overcome the missing directional selectivity of conventional 2D DWTs, there have been several developments of wavelet frames in recent years. Steerable wavelets [84], Gabor wavelets [131], brushlets [155], beamlets [60], ridgelets [56], curvelets [33], contourlets [57], shearlets [92], wave atoms [54], surflets [39] were proposed independently with a common goal: better representing directional features in images. Among these X-lets, curvelets have the highest publicity and have found applications in several domains [140]. In the 2D case, the curvelet transform allows an almost optimal sparse representation of objects with singularities along smooth curves. For a smooth object f with piecewise C^2 singularities, the best N-term approximation $\hat{f}_N$ of f, which is a linear combination of only N elements of the curvelet frame, obeys $\|f - \hat{f}_N\|_2 \leq CN^{-2}(\log N)^3$, while for wavelets the decay rate is only N^{-1}.

The curvelet transform has a property that the coefficients of those curvelets whose essential supports do not overlap with, or overlap with but are not tangent to edges are small and negligible. For example, in Fig. 4.9b, coefficients of curvelets of types A and B are negligible, most of the energy of the graphics is localized in just a few coefficients of curvelets of type C. In other words, curvelet transform produces a sparse representation of objects, most of the energy of the objects is localized in just a few coefficients of curvelets which overlap and are nearly tangent to the

object contours. Based on this property, the application of curvelet transform for image denoising image is straightforward; it could be simply done by hard-thresholding curvelet coefficients as in Eq. (4.11) [205]. In addition, the images reconstructed by curvelets exhibit higher perceptual quality than those by wavelets. They have visually sharper and, in particular, higher-quality recovery of edges and of linear/curvilinear features.

Thresholding as solving an optimization problem

Assuming that denoising is performed by using Eq. (4.10) where $\mathbf{T}$ is unitary like in the case of DWT. Consider the following optimization problem:

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_q^q \tag{4.13}$$

which needs to be solved to obtain an estimated image $\hat{\mathbf{x}} = \mathbf{T}^T\hat{\boldsymbol{\alpha}}$ where $\mathbf{D} = \mathbf{T}^T = \mathbf{T}^{-1}$. Due to the unitarity of $\mathbf{T}$, it is straightforward that

$$\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 = \|\mathbf{x} - \mathbf{T}^T\boldsymbol{\alpha}\|_2^2 = \|\mathbf{T}\mathbf{x} - \boldsymbol{\alpha}\|_2^2,$$

and the aforementioned optimization problem translates into

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_q^q, \tag{4.14}$$

where $\boldsymbol{\beta} = \mathbf{T}\mathbf{x}$ is the transform of the noisy image $\mathbf{x}$. Since both $\|\boldsymbol{\beta}-\boldsymbol{\alpha}\|_2^2$ and $\|\boldsymbol{\alpha}\|_q^q$ are separable, the above problem decouples into a set of p independent problems of the form

$$\hat{\alpha}_i = \underset{\alpha_i}{\operatorname{argmin}} \frac{1}{2}|\beta_i - \alpha_i|_2^2 + \lambda|\alpha_i|_q^q, \quad i = 1, 2, \ldots, p. \tag{4.15}$$

It is not difficult to demonstrate that the unique closed-form solutions to these problems in the two notable cases $q = 0$ and $q = 1$ are actually the two thresholding operators defined in Eqs. (4.11) and (4.12) respectively with $\alpha_i = \mathcal{O}(\beta_i, \lambda)$. Thus, for the cases of orthonormal transforms, thresholding-based denoising could be viewed as solving an optimization problem of the form in Eq. (4.13) with $q = 0, 1$ for hard-thresholding and soft-thresholding operators respectively.

Basis pursuit denoising

The thresholding operators defined in Eqs. (4.11) and (4.12) are the exact solutions to the optimization problem in Eq. (4.13) for the two cases $q = 0$ and $q = 1$ only if $\mathbf{D}$ is unitary. When a redundant transform like the curvelet transform is used, the corresponding overcomplete dictionary $\mathbf{D}$ has more columns than rows and thus is non-unitary ($\mathbf{D}\mathbf{D}^T = \mathbf{I}$ but $\mathbf{D}^T\mathbf{D} \neq \mathbf{I}$ where $\mathbf{I}$ is the identity matrix). The problem in Eq. (4.13) does not have a simple and closed-form solution, even in the two notable cases $q = 0$ and $q = 1$. This is because the presence of the non-unitary matrix $\mathbf{D}$ destroys the separability that allows solving the relatively easy problem in Eq. (4.15) instead of the more demanding problem in Eq. (4.14).

However, for the graphical document image denoising problem, the formulation in Eq. (4.13) at $q = 1$ is still adopted with the overcomplete dictionary $\mathbf{D}$ being defined as the synthesis operator of the curvelet transform in order to obtain smooth graphical contours. Moreover, for the purpose of facilitating the investigation of the dependence of the framework parameter on the noise level, the optimization problem is rewritten as

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\boldsymbol{\alpha}\|_1 \qquad \text{subject to} \qquad \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2 \leq \epsilon, \tag{4.16}$$

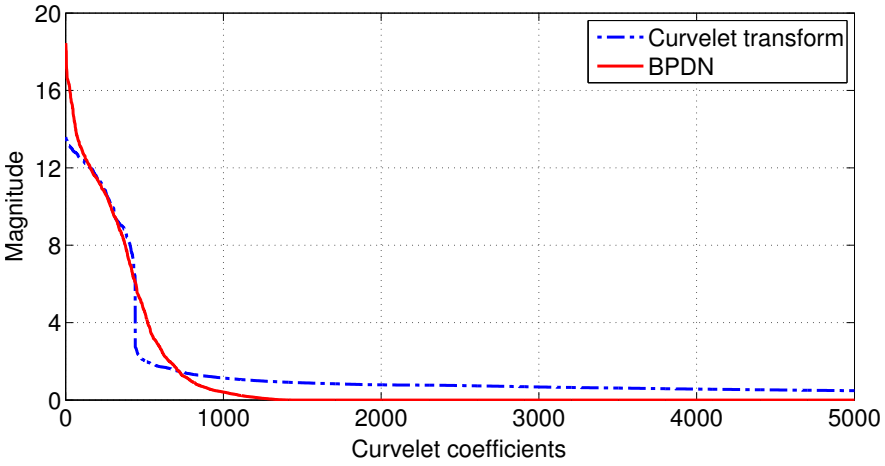


Figure 4.11: The distribution of the magnitudes of the 5000 largest coefficients of the noisy image in Fig. 4.9a obtained from curvelet transform and BPDN with $\epsilon = 48$. It can be seen that small-valued coefficients resulting from curvelet transform are zeroed out by BPDN.

where ϵ is the precision parameter that depends on $\mathbf{z}$. Note that the above problem is a slightly modified version of the BPDN problem defined in Eq. (4.2) where the squared Euclidean norm is replaced by the Euclidean norm. This modification, however, does not change the nature of the problem because the value of ϵ could also be changed accordingly:

$$\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2 \leq \epsilon \qquad \Leftrightarrow \qquad \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \leq \epsilon^2.$$

From this observation, solutions of the problem in Eq. (4.16) could be found by using methods designed for BPDN that have been discussed in Subsection 4.1.1. Interestingly, it was shown that the simple shrinkage could be interpreted as the first iteration of an algorithm that solves BPDN [72]. By solving the problem in Eq. (4.16), the estimated image $\hat{\mathbf{x}}$ could be obtained from the sparse reconstruction as $\hat{\mathbf{x}} = \mathbf{D}\hat{\boldsymbol{\alpha}}$. Since $\hat{\mathbf{x}}$ is reconstructed from curvelets contained in $\mathbf{D}$, it is of course in grayscale; it could finally be converted to bilevel by a simple thresholding operation as $\tilde{\mathbf{x}} = \mathcal{T}(\hat{\mathbf{x}})$ where $\mathcal{T}$ is the thresholding operator.

In the above BPDN problem, the ℓ_1-norm is used instead of a more general ℓ_q-norm to avoid the NP-hard problem [49] when $q = 0$, non-convexity when $q < 1$, non-sparse and over-fitting solutions when $q > 1$. In addition, a sparse solution, which guarantees directional denoising by curvelets, is still obtained if the solution of the following ℓ_0-norm optimization problem is sufficiently sparse [59]:

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\boldsymbol{\alpha}\|_0 \qquad \text{subject to} \qquad \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2 \leq \epsilon.$$

As the overcomplete dictionary $\mathbf{D}$ is constructed from curvelets and the images to be processed contain mainly graphical contours, this requirement is easily satisfied. Illustration of the distribution of the magnitudes of the 5000 largest curvelet coefficients of the image in Fig. 4.9a obtained from curvelet transform and BPDN is given in Fig. 4.11. It is observed that BPDN results in a sparse representation, many elements of $\hat{\boldsymbol{\alpha}}$ have almost zero value. The sparsity in $\hat{\boldsymbol{\alpha}}$ is, in some respects, better than that in the coefficients resulting from the curvelet transform $\mathbf{D}^T\mathbf{x}$. In addition, the shape of the distribution of $\hat{\boldsymbol{\alpha}}$ resembles that of a Laplacian distribution and this agrees with the Bayesian formulation of sparse coding presented in Subsection 4.1.3.

It should also be noted that the problem in Eq. (4.16) and the one in Eq. (4.9) are similar, they are both minimization problems with a fidelity constraint. The main difference between them is that TV denoising pursues an estimation that is sparse in the spatial domain (sparse gradient) whereas BPDN denoising has a desire for sparsity in the transform domain. Fusing TV with BPDN amounts to solving [34]

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\ \mathrm{TV}(\boldsymbol{\alpha}) \qquad \text{subject to} \qquad \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2 \leq \epsilon.$$

However, as the gradient of $\boldsymbol{\alpha}$ is not clearly defined, imposing piece-wise smoothness in $\boldsymbol{\alpha}$ by TV regularization in this way usually produces images with non-deterministic artifacts.

The precision parameter ϵ

The BPDN problem in Eq. (4.16) has a non-negative precision parameter ϵ that describes the desired fidelity of the reconstructed image $\hat{\mathbf{x}}$ to the noisy image $\mathbf{x}$. It is the only parameter, besides the selected dictionary $\mathbf{D}$, that controls the quality of denoised images. It is straightforward that when $\epsilon = 0$, the BPDN problem reduces to a simple curvelet transform:

$$\mathbf{x} = \mathbf{D}\hat{\boldsymbol{\alpha}} \qquad \longrightarrow \qquad \hat{\boldsymbol{\alpha}} = \mathbf{D}^T\mathbf{x}$$

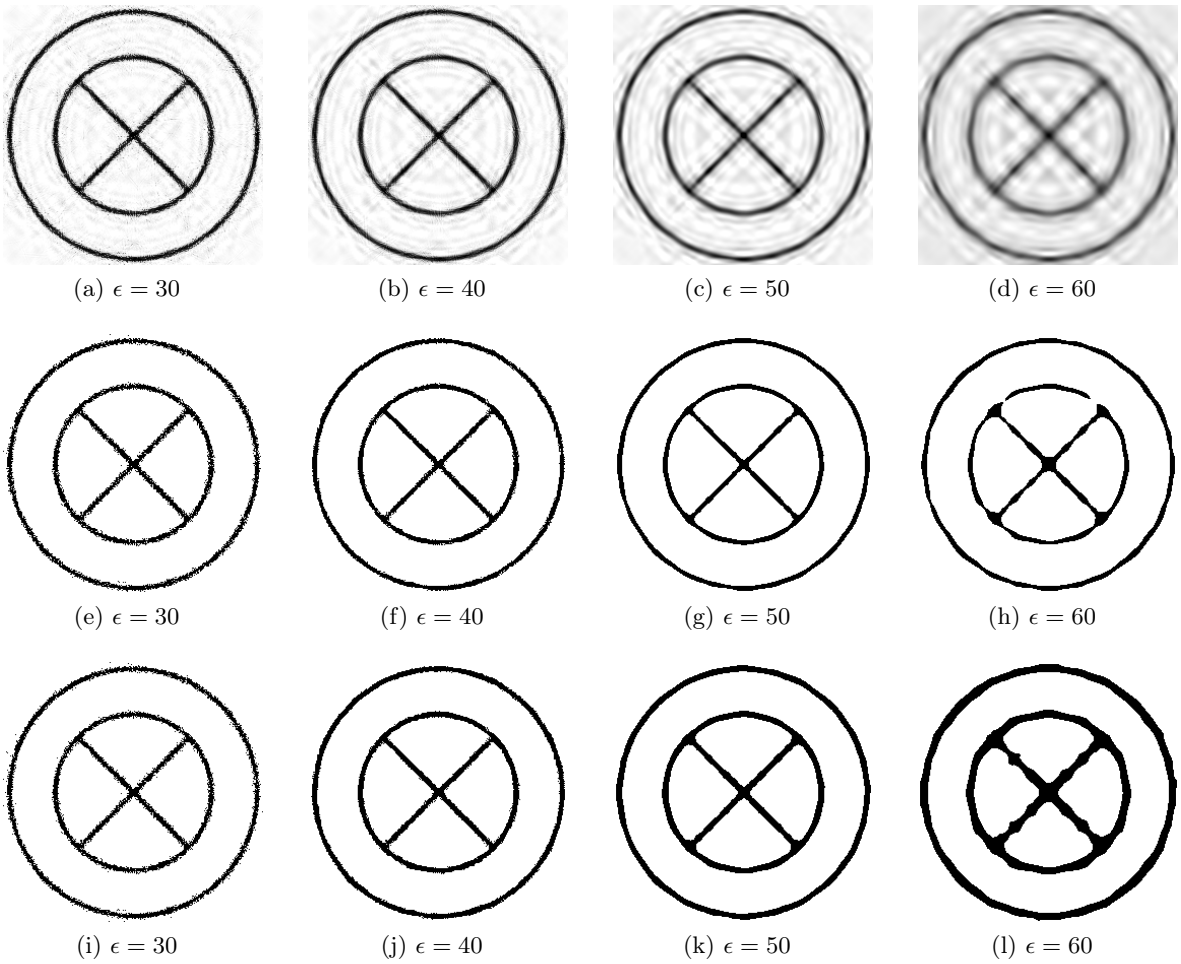
and one easily has $\mathbf{x} = \hat{\mathbf{x}} = \tilde{\mathbf{x}}$. As the value of ϵ increases, the measure of sparsity $\|\hat{\boldsymbol{\alpha}}\|_1$ of the solution $\hat{\boldsymbol{\alpha}}$ must monotonically decrease since the feasible set of solutions $\mathcal{S} = \{\boldsymbol{\alpha} : \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2 \leq \epsilon\}$ gets wider, taking the feasible set of solutions at a smaller ϵ as a subset:

$$\epsilon_1 < \epsilon_2 \qquad \longrightarrow \qquad \mathcal{S}_1 \subset \mathcal{S}_2.$$

A sparser solution means a better alignment of curvelets with graphics contours, which consequently increases the denoising performance. However, when ϵ has a reasonably large value, the solution $\hat{\boldsymbol{\alpha}}$ of the BPDN problem may be overly sparse in terms of ℓ_1-norm and the estimated image $\hat{\mathbf{x}}$ gets overly blurred. In addition, due to the thresholding operation to get the binary image $\tilde{\mathbf{x}}$ from $\hat{\mathbf{x}}$, deformation in $\tilde{\mathbf{x}}$ will appear. Illustration of the influence of the value of ϵ on the estimated images using the noisy image in Fig. 4.9a at $\epsilon = 30, 40, 50, 60$ is given in Fig. 4.12. It can be seen that for both thresholding operations using a fixed threshold of 0.5 or using Otsu's threshold [167]:

- A small value of $\epsilon = 30$ results in insufficient blurring in the estimated images. The binarized images still have noise along the contours.
- A large value of $\epsilon = 60$ causes over blurring in the estimated images. Deformation can be observed in the binarized images.

The selected value of ϵ should depend on the level of noise that exists in the images. In the literature, there exists no work that discusses in detail the dependance of ϵ on an image's noise level. For zero-mean white and homogeneous Gaussian noise with a known standard-deviation σ, the value of ϵ is usually chosen as $cn\sigma^2$, with $0.5 \leq c \leq 1.5$ [70, Chapter 14]. For graphical document images, the theory of edge noise presented in Subsection 4.2.1 sheds light on this problem by the established linear relationship between NS and the expected Hamming distance as given in Eq. (4.8). It is thus fair to conjecture that the relation $\epsilon(\mathrm{NS})$ should also be linear and is of the form $\epsilon = k\mathrm{NS}$. This is because $\|\mathbf{x} - \mathcal{T}(\mathbf{D}\boldsymbol{\alpha})\|_2$ is essentially the Hamming distance between the binary denoised image $\tilde{\mathbf{x}}$ and its corresponding noisy one $\mathbf{x}$. The linear relationship between ϵ and NS will have experimental evidence in the following section.



(a) $\epsilon = 30$ (b) $\epsilon = 40$ (c) $\epsilon = 50$ (d) $\epsilon = 60$

(e) $\epsilon = 30$ (f) $\epsilon = 40$ (g) $\epsilon = 50$ (h) $\epsilon = 60$

(i) $\epsilon = 30$ (j) $\epsilon = 40$ (k) $\epsilon = 50$ (l) $\epsilon = 60$

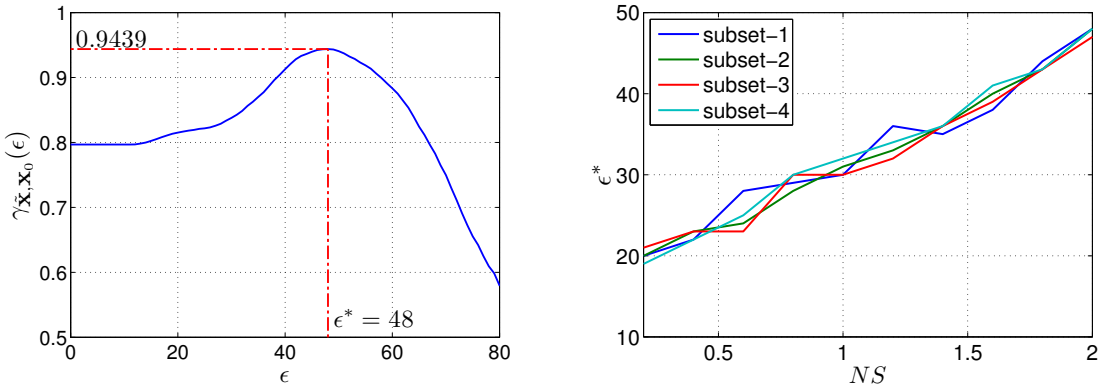
Figure 4.12: Influence of the value of ϵ on the estimated images using the noisy image in Fig. 4.9a at $\epsilon = 30, 40, 50, 60$. *Top row*: estimated images in grayscale, *middle row*: bilevel denoised images using a fixed threshold of 0.5, *bottom row*: bilevel denoised images using Otsu's threshold.

4.2.4 Experimental results

In order to demonstrate the effectiveness of bilevel graphical document image denoising using BPDN with curvelets as the overcomplete dictionary $\mathbf{D}$, two types of experiments have been carried out: one for the validation of the linear relationship between the parameter ϵ and NS; the other for the demonstration of the superiority of the proposed method over comparison ones in terms of image recovery and contour smoothness. While the use of image recovery as an evaluation criteria is straightforward, contour smoothness is adopted to quantitatively evaluate the capability of comparison methods in producing denoised images of good visual quality.

The relation ϵ(NS)

A dataset SetA containing 40 noisy images has been generated from a ground-truth and noise-free image "symbol017" to be used as the testing dataset. "symbol017" is a graphical symbol image of size 256×256 taken from the GREC2005 database [65], its noisy version is given in Fig. 4.15b. SetA is divided into four subsets, each contains 10 noisy images that correspond to 10 values



(a) A plot of $\gamma_{\tilde{\mathbf{x}},\mathbf{x}_0}(\epsilon)$ at NS = 2.0 (b) ϵ^*(NS) relation

Figure 4.13: Determination of the value of the precision parameter ϵ in Eq. (4.16): (a) its optimal value ϵ^* is determined by means of image recovery; (b) the linear relationship between ϵ^* and NS for the four experimental subsets.

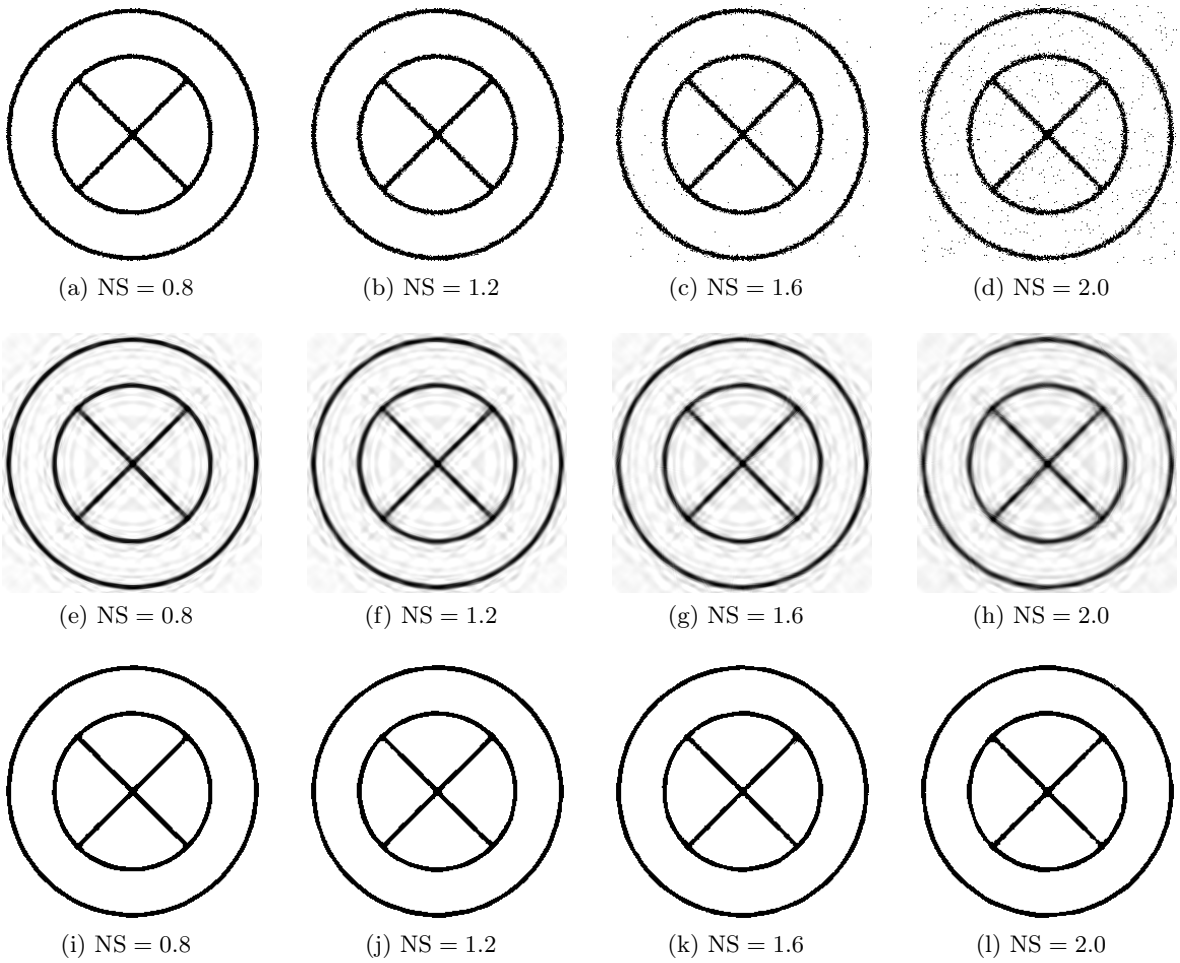
of NS ranging from 0.2 to 2.0 with increments of 0.2. Assuming that the parameter ϵ of BPDN in Eq. (4.16) takes the value ϵ^* that corresponds to the peak in denoising performance in terms of a measure of fidelity between $\tilde{\mathbf{x}}$ and $\mathbf{x}_0$, the relation ϵ^*(NS) thus needs to be established experimentally. The measure of fidelity, denoted by $\gamma_{\tilde{\mathbf{x}},\mathbf{x}_0}$, employed in this work is the normalized cross-correlation between 2D data as defined in Eq. (2.29).

Illustration of the determination of ϵ^* by means of $\gamma_{\tilde{\mathbf{x}},\mathbf{x}_0}(\epsilon)$ is given in Fig. 4.13a where the noisy image in Fig. 4.15b with NS = 2.0 is taken as the input image. To compute the value of $\gamma_{\tilde{\mathbf{x}},\mathbf{x}_0}(\epsilon)$ at each possible value of ϵ, the ℓ_1-optimization problem in Eq. (4.16) is solved for $\hat{\boldsymbol{\alpha}}$ and then the value of the bilevel denoised image $\tilde{\mathbf{x}} = \mathcal{T}(\hat{\mathbf{x}}) = \mathcal{T}(\mathbf{D}\hat{\boldsymbol{\alpha}})$ can be easily obtained. The plot of $\gamma_{\tilde{\mathbf{x}},\mathbf{x}_0}(\epsilon)$ in the case of a fixed thresholding of 0.5 has its maximum value of 0.9439 at $\epsilon^* = 48$. This means that if the input noisy image has NS = 2.0, ϵ in Eq. (4.16) should take the value 48 in order to have the "best" denoising performance. It should be noted here that, due to the blunt maxima in $\gamma_{\tilde{\mathbf{x}},\mathbf{x}_0}$, a small deviation of the selected value of ϵ form ϵ^* has almost no effect on the performance of the proposed method.

Having determined the value of ϵ^* for each image of a certain NS in SetA, the relation ϵ^*(NS) is established for each of the four subsets of SetA. These four relations are then plotted separately in Fig. 4.13b. It can be seen that an image of higher NS requires a larger value of ϵ^* for optimal performance. In addition, ϵ^* has a nearly linear relationship with NS for all four subsets. A narrow band formed by ϵ^*(NS) also means that the standard deviation of ϵ^* is reasonably small. Combining this fact with the blunt maxima in $\gamma_{\tilde{\mathbf{x}},\mathbf{x}_0}(\epsilon)$, it thus can be concluded that the performance of the proposed method with curvelets as the dictionary is guaranteed to be almost optimal if its only parameter ϵ is estimated from the relation ϵ^*(NS).

Comparison with existing methods

The proposed method for denoising bilevel graphical document images using BPDN was evaluated on two datasets. The first is SetA as described in the previous experiment. The second, SetB, is generated from four ground-truth and noise-free graphical symbol images "symbol016", "symbol017", "symbol024", and "symbol081" also from the GREC2005 database. Some noisy versions of the four ground-truth images used to generate SetB are given in Figs. 4.15a–4.15d.



(a) NS = 0.8 (b) NS = 1.2 (c) NS = 1.6 (d) NS = 2.0

(e) NS = 0.8 (f) NS = 1.2 (g) NS = 1.6 (h) NS = 2.0

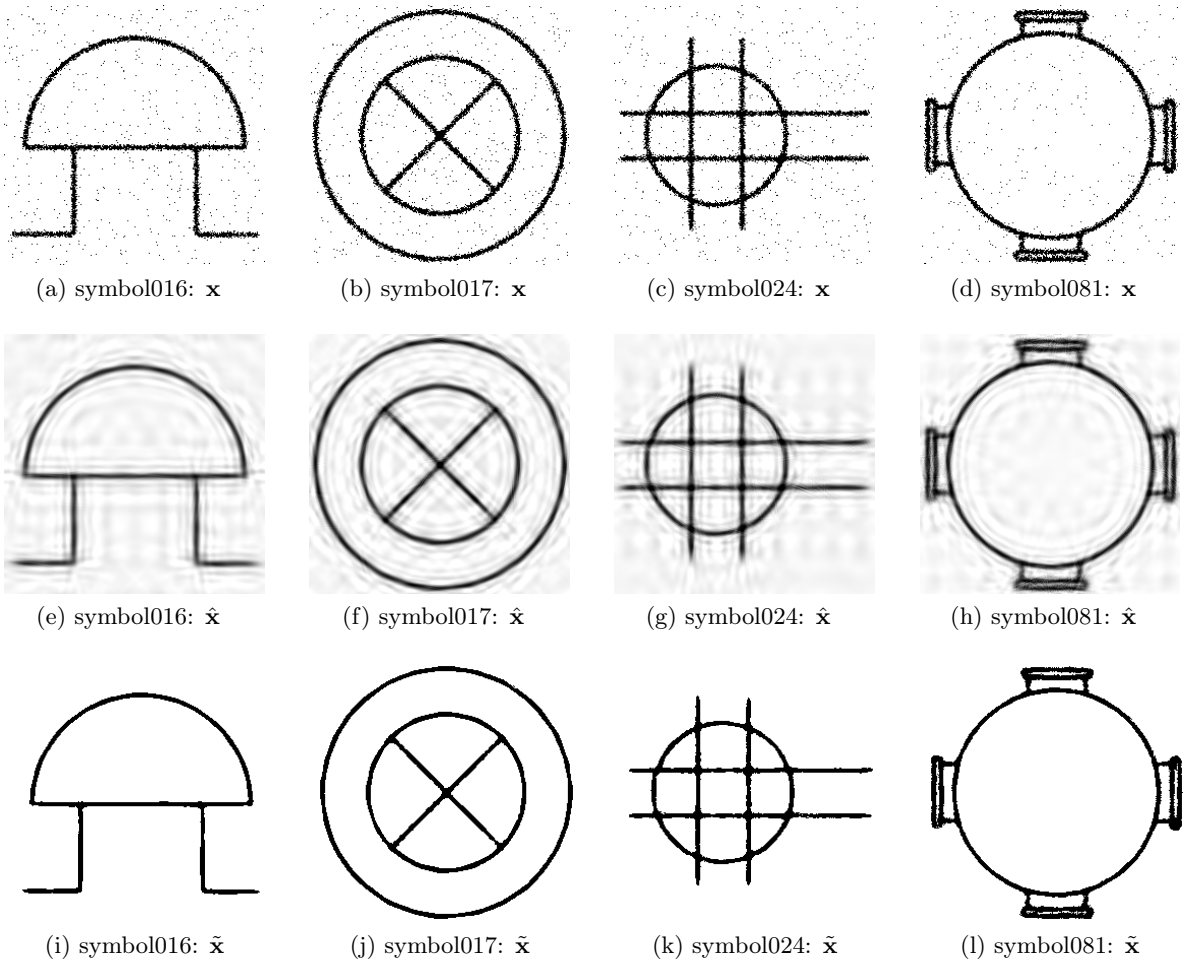
(i) NS = 0.8 (j) NS = 1.2 (k) NS = 1.6 (l) NS = 2.0

Figure 4.14: Some samples of noisy images from the dataset SetA at different values of NS and the corresponding denoised images obtained by using the proposed method. *Top row*: original noisy images, *middle row*: estimated images in grayscale, *bottom row*: denoised images in binary.

These images are selected due to the existence of all possible graphics contour directions and various configurations of contours that may cause difficulties in denoising. For each ground-truth image, five noisy images that correspond to five values of NS $= 0.2, 0.6, 1.0, 1.5, 2.0$ have been generated. SetB thus has a total of 20 noisy images.

Figs. 4.14 and 4.15 provide examples of denoised graphical symbols by the proposed method for SetA and SetB respectively with $\epsilon = \epsilon^*$ for each case in order to have optimal performance. In these figures, the original noisy images of NS $= 0.8, 1.2, 1.6, 2.0$ for SetA and NS $= 2.0$ for SetB are given in the first row. The corresponding estimated images in grayscale are given in the middle row. Evidence of directional denoising along noisy contours exists in the corresponding estimated images in grayscale: edge noise is smoothed out in the direction that is perpendicular to the noisy contours. This is like the images have been filtered locally along the noisy contours by anisotropic filters, each has its direction coincident with the local direction of the nearest contour. Due to this directional filtering phenomena, the denoised images in binary using a fixed threshold of 0.5 in the bottom row are clean and have smooth contours.

To demonstrate the effectiveness of the proposed method, comparison with the following



(a) symbol016: $\mathbf{x}$ (b) symbol017: $\mathbf{x}$ (c) symbol024: $\mathbf{x}$ (d) symbol081: $\mathbf{x}$

(e) symbol016: $\hat{\mathbf{x}}$ (f) symbol017: $\hat{\mathbf{x}}$ (g) symbol024: $\hat{\mathbf{x}}$ (h) symbol081: $\hat{\mathbf{x}}$

(i) symbol016: $\tilde{\mathbf{x}}$ (j) symbol017: $\tilde{\mathbf{x}}$ (k) symbol024: $\tilde{\mathbf{x}}$ (l) symbol081: $\tilde{\mathbf{x}}$

Figure 4.15: Some samples of noisy images from the dataset SetB at NS = 2.0 and the corresponding denoised images obtained by using the proposed method. *Top row*: original noisy images, *middle row*: estimated images in grayscale, *bottom row*: denoised images in binary.

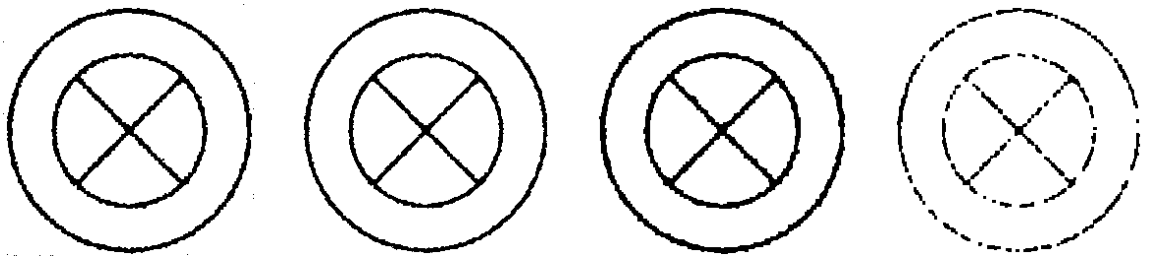
frequently used methods has been carried out:

- Median filtering using a 3×3 neighborhood, kFill filtering with the parameter $k = 3$, closing then opening using a 3×3 structuring element, and opening then closing using a 3×3 structuring element.

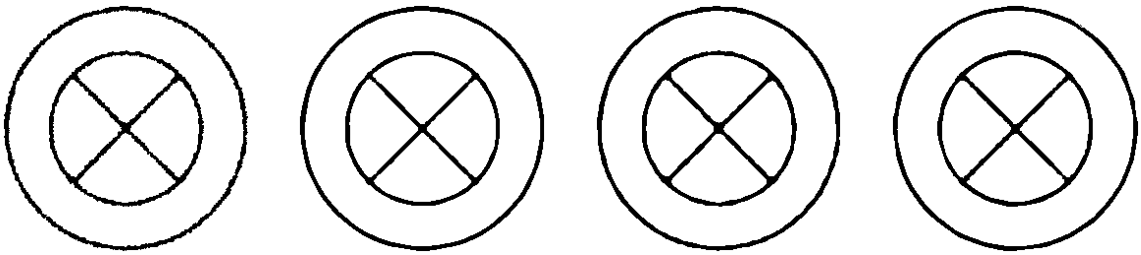
- Total variation: ϵ_{TV} takes the value ϵ^*_{TV} that corresponds to optimal performance in terms of normalized cross-correlation. The selection of ϵ_{TV} is similar to the selection of ϵ in the proposed method.

- Shrinkage: hard-thresholding of curvelet coefficients with one threshold value λ_{jl} is used for all curvelets of scale j and angle l. λ_{jl} is computed by applying a forward curvelet transform on an image containing a delta function at its center.

- Diffusion: iterative applications of anisotropic diffusion and coherence enhancing diffusion in sequence with parameters determined by experience.



(a) kFill (b) Median (c) Close-Opening (d) Open-Closing

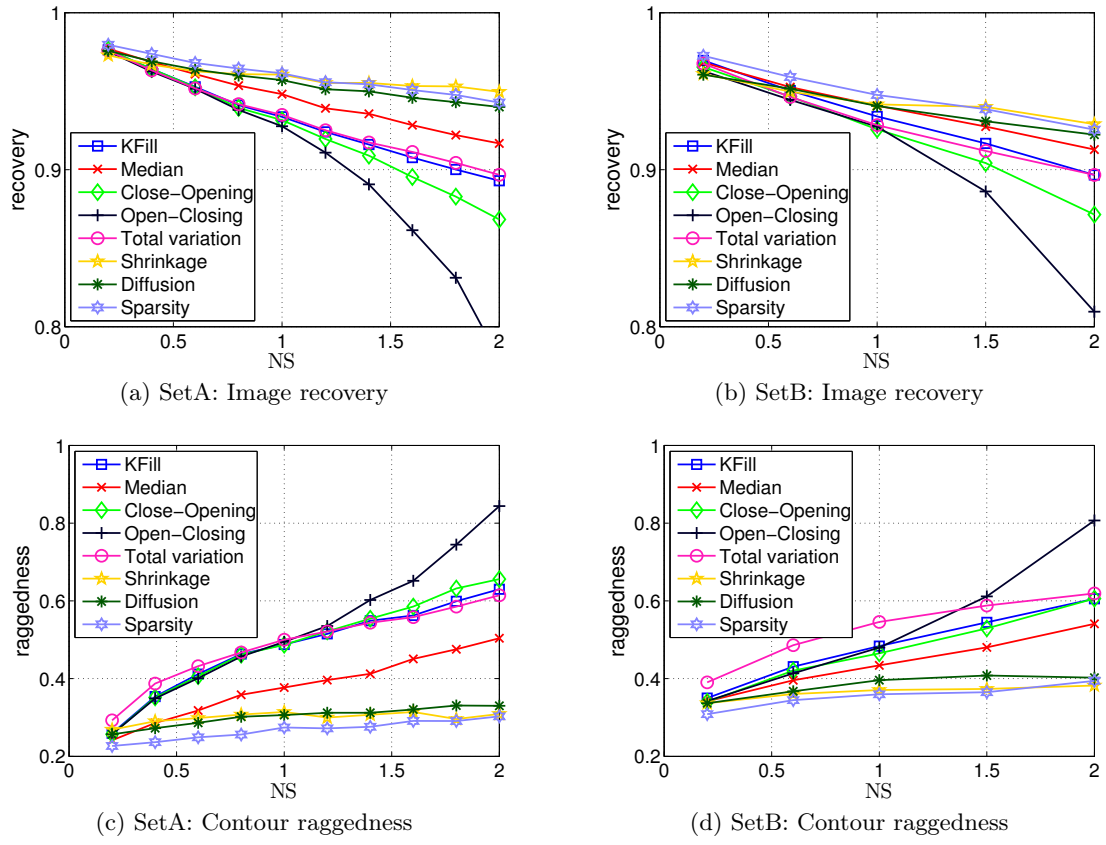


(e) Total variation (f) Shrinkage (g) Diffusion (h) Sparsity

Figure 4.16: Samples of denoised images from comparison methods using an image of NS = 2.0 from the dataset SetA. Shrinkage, diffusion, and sparsity-based methods produce images of good visual quality, whereas the other methods result in images that have ragged edges.

The criteria used for comparison are the ability to recover the original images (measured by the normalized cross-correlation between the denoised and the ground-truth images as defined in Eq. (2.29)) and the raggedness of the graphics contours (a moving average of the raggedness measure defined in [104]). The proposed method and all other comparison methods are each applied to each image in datasets SetA and SetB and then normalized cross-correlation and raggedness are measured for each resulting denoised image. Samples of denoised images from comparison methods using an image of NS = 2.0 from the dataset SetA are given in Fig. 4.16. It can be seen that shrinkage, diffusion, and sparsity-based methods produce denoised images of good visual quality. However, the diffusion method has more difficulties in restoring the sharp corners of the contours. The images resulting from kFill filtering, median filtering, morphology-based methods, and total variation have ragged edges and bad visual quality with the worst images resulting from morphology-based methods.

The performance of each method per noise level on one dataset is defined as the average performance over all the noisy images of the same noise level in that dataset. The comparison results are shown in Fig. 4.17 for these two criteria over a range of noise levels (left column for SetA and right column for SetB). It is observed that as the noise level (NS) increases, the ability to recover the original images decreases and the contour raggedness of the denoised images increases for all methods. The performance of kFill filtering, morphology-based methods, and total variation are similarly bad with open-closing breaks down when the noise level is reasonably high (NS > 1). Median filtering has its performance in the middle and top performance belongs to shrinkage, diffusion, and sparsity-based methods. The decrease in the image recovery and increase in the contour raggedness of these three best methods are nearly constant and are the smallest among all methods for both datasets. Nevertheless, the diffusion method has slightly worse performance than those of shrinkage and sparsity. Moreover, sparsity outperforms shrinkage when the noise level is small and moderate (NS < 1.5) and their performance are comparable



(a) SetA: Image recovery

(b) SetB: Image recovery

(c) SetA: Contour raggedness

(d) SetB: Contour raggedness

Figure 4.17: Performance evaluation of the proposed and comparison denoising methods in terms of image recovery (top row) and contour raggedness (bottom) on the two experimental datasets SetA (left column) and SetB (right column).

when the noise level is high (NS $\geq$ 1.5). The proposed method thus usually results in denoised images of best recovery with smoothest contours at all values of NS. It should also be noted from the comparison results that for noisier images, in terms of image recovery and contour raggedness, the proposed method produces the most significant improvements.

4.3 Text/graphics separation

Besides the denoising problem that has been addressed in the previous section, another main problem in document image processing that could be tackled by sparse coding is the extraction of text components from graphical document image. This section revisits the problem and views it as a blind source separation problem in signal processing with text and graphics components have different morphological characteristics. Based on this new perspective, a new sparse-based solution for the text extraction problem is proposed; its performance is then compared with the previous benchmark.

4.3.1 The text extraction problem

Text extraction from graphical document images is a major problem in document image analysis in which one document input image that contains both text and graphics is processed to produce

two output images, one containing text and the other containing graphics. The importance of text extraction is due to the possible existence of text's semantic meaning, which could be obtained from the extracted text by using an optical character recognition (OCR) engine and a linguistic tool, and, more importantly, could facilitate the interpretation of scanned graphical documents. For this reason, a reliable text extraction method is required to make it usable in automatic document processing systems. At present, applications of such text extraction algorithm are automatic processing of texture documents and architectural/engineering drawings, automatic reading of postal addresses and flexible forms, etc. Basically, graphics components contained in document images are of various types according to each specific application domain but generally they are lines, curves, polygons, circles. Meanwhile, text components consist of characters and digits which form words and phrases to annotate the graphics. Extraction of text components is a challenging problem because of the following reasons:

- Graphical components like lines can be of any length, thickness, and orientation. Circles, polygons can be filled or unfilled. Text components can vary in font styles and sizes.

- There may exist touching between text components and touching, crossing between text and graphics components. Text strings are usually intermingled with graphics and can have any orientation.

- Excluding the pre-processing steps to enhance image quality, text extraction is mostly the first step in the chain of document analysis with limited knowledge about the presence of high-level objects in the images.

4.3.2 Related works

Several methods were proposed to tackle the problem of text extraction from graphics-containing documents and they can be roughly classified into three main families according to their nature:

- *Morphological analysis*: Constrained run-length algorithm (CRLA) [223] is one of the first and best known methods based on morphological filtering to detect long vertical and horizontal text strings. It essentially consists of morphological closing operations using horizontal and vertical structuring elements of specified length. Although CRLA and its improvement [126] are very efficient for textual documents, its use in graphics-rich documents [124, 138] is limited because text could be wrongly labeled as graphics.

- *Connected component analysis*: A well-known approach [80] based on connected component analysis uses some heuristic rules regarding area, dimension ratio and collinearity of connected components to separate text from graphics. Simplicity and scalability are the strength of this approach, making it widely used. However, the weakness of this approach is the inability to directly separate text which touches graphics. An effort [219] to overcome this problem achieves some improvements for graphics-rich document images by incorporating some more heuristic rules.

- *Multi-resolution analysis*: Multi-resolution approach was first proposed in [53] for mail pieces and then adapted to map in [212]. It relies on the assumption that at a certain coarse level of the image pyramid, a text line looks like a long component; and at the next finer level it looks like a regular sequence of transitions. However, when text and graphics components lie closely or touch, this approach induces wrong detection results.

Generally, the aforementioned approaches cannot work with the difficult case of touching between text and graphics components. Several works were proposed to tackle the touching problem, each of them belongs to one of the following two families:

- *Touching lines detection*: The common strategy to separate text from graphics is by detecting and then removing touching lines, assuming that lines can be detected easily. For example, Hough transform is used in [86] to detect vertical and horizontal lines in form structure recognition; slant lines in engineering drawings are detected by first stretching the document to certain angles and then tracing black pixels horizontally and vertically [138]; linear shapes in simple maps are located by employing directional morphological filtering [139]; vectorization is used in [64, 209] to detect and remove graphics components.

- *Local statistics*: Another direction is based on the local discriminative statistics of text and graphics primitives. Dimensioning text components are detected in [63] based on the presence of neighboring already-detected graphics primitives, such as bars, arcs, and arrowheads [62]. The method in [36] uses a skeletonized version of a map and consider short and long skeleton segments as skeletons of text and graphics components respectively. A recently proposed method [246] generates local consecutive segments (LCSs) and distinguishes LCSs of text from those of graphics by means of some statistical measures.

Each of the aforementioned methods that deals with touching is initially designed for a specific application; it is not robust and almost inapplicable to graphical images of the others. For example, with a graphics-rich and complex engineering document image showed in Fig. 4.18a, none of the above methods provides reliable results, giving rise to a demand for a new proposal. The method proposed in this section extracts text components in a totally different way from existing ones. A document image $\mathbf{x}$ that contains text and graphics components is henceforth considered as a 2D signal, which is the mixture of two separate 2D signals of the same size as

$$\mathbf{x} = \mathbf{x}_t + \mathbf{x}_g,$$

where $\mathbf{x}_t$ and $\mathbf{x}_g$ contain text and graphics components respectively. The problem of text extraction is now seen as the inverse problem of recovering $\mathbf{x}_t$ and $\mathbf{x}_g$ from $\mathbf{x}$, which essentially has the same nature as the blind source separation problem in multi-dimensional signal processing [111].

In order to solve this problem, the morphological component analysis (MCA) algorithm proposed in [206], which allows the separation of morphologically different features in an image, has been employed. MCA-based separation is facilitated by promoting sparse representation of these features in two appropriately chosen dictionaries, each leads to sparse representation over one feature and non-sparse representation over the other. Having done in this way, some post-processing steps could be needed to extract text strings from $\mathbf{x}_t$; this is done with the help of some heuristic rules proposed in this section based on the discriminative characteristics of text components. The proposed method is robust to touching between text and graphics. It can extract text components that are in any form, have any font style/size, and are placed anywhere with any orientation in the documents.

4.3.3 Morphological component analysis

The MCA method is a further development of the framework represented in Section 4.1 to deal with the problem of separating an image content into semantic parts. MCA has been shown to be very useful for decomposing images into texture and piece-wise smooth (cartoon) parts or for inpainting applications [71, 76]. It was also adopted in [168] for the segmentation of text

from complex background. The application of MCA to a new application domain, text/graphics separation, is presented in this subsection. The representation is followed by some post-processing steps proposed uniquely for this type of applications in the next subsection.

Let a signal $\mathbf{x} \in \mathbb{R}^p$ be a linear combination of two parts as $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$ where $\mathbf{x}_1$ and $\mathbf{x}_2$ represent two "different" types of signals. Assuming that there exist two overcomplete dictionaries $\mathbf{D}_1, \mathbf{D}_2 \in \mathbb{R}^{p \times K}$ that satisfy the following two conditions:

- Solving (for $i = 1, 2$)

$$\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{\alpha}_i\|_1 \qquad \text{subject to} \qquad \mathbf{x}_i = \mathbf{D}_i \boldsymbol{\alpha}_i, \tag{4.17}$$

  leads to a sparse representation $\hat{\boldsymbol{\alpha}}_i$ of $\mathbf{x}_i$ in $\mathbf{D}_i$.

- Solving (for $i \neq j$)

$$\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{\alpha}_i\|_1 \qquad \text{subject to} \qquad \mathbf{x}_i = \mathbf{D}_j \boldsymbol{\alpha}_i, \tag{4.18}$$

  leads to a non-sparse representation $\hat{\boldsymbol{\alpha}}_i$ of $\mathbf{x}_i$ in $\mathbf{D}_j$.

In this manner, the two dictionaries $\mathbf{D}_1$ and $\mathbf{D}_2$ are said to be discriminative in the sense of sparse representation to different content types, $\mathbf{x}_1$ and $\mathbf{x}_2$. MCA method thus proposes to solve the following optimization problem:

$$\min_{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2} \left(\|\boldsymbol{\alpha}_1\|_0 + \|\boldsymbol{\alpha}_2\|_0\right) \qquad \text{subject to} \qquad \mathbf{x} = \mathbf{D}_1 \boldsymbol{\alpha}_1 + \mathbf{D}_2 \boldsymbol{\alpha}_2,$$

which can be converted to:

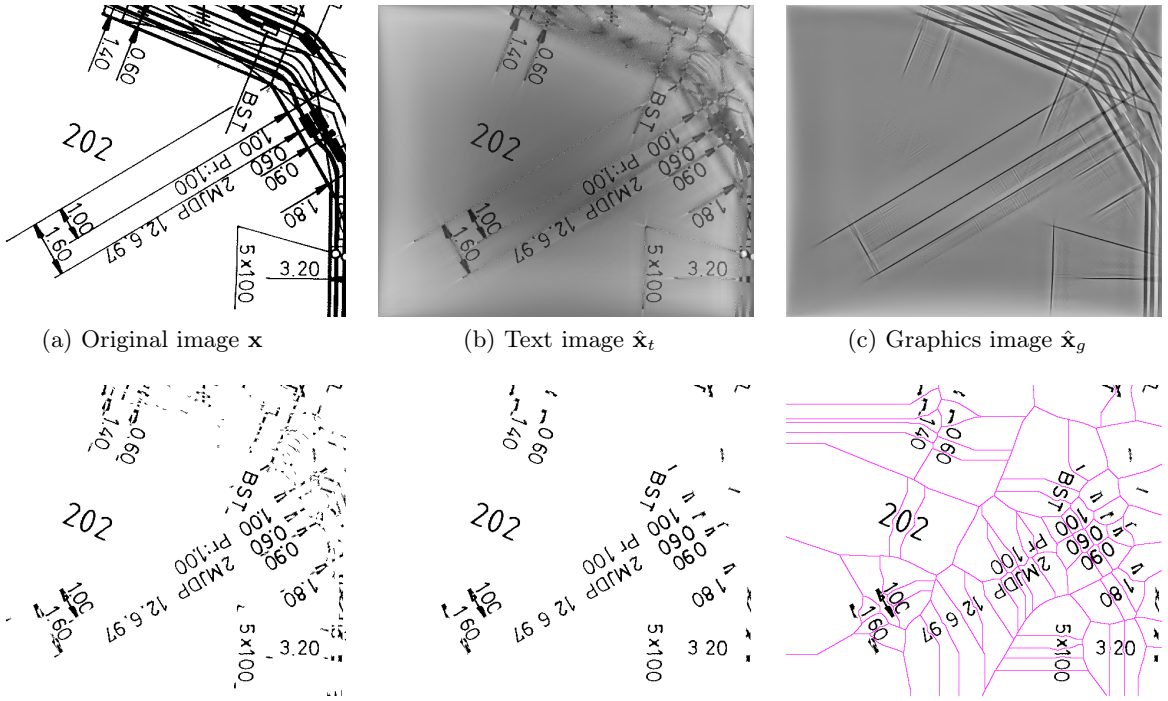
$$\min_{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2} \left(\|\boldsymbol{\alpha}_1\|_1 + \|\boldsymbol{\alpha}_2\|_1 + \lambda \|\mathbf{x} - \mathbf{D}_1 \boldsymbol{\alpha}_1 - \mathbf{D}_2 \boldsymbol{\alpha}_2\|_2\right). \tag{4.19}$$

Solving the optimization problem in Eq. (4.19) gives $\hat{\boldsymbol{\alpha}}_1$ and $\hat{\boldsymbol{\alpha}}_2$, the sparse representation of $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\mathbf{D}_1$ and $\mathbf{D}_2$ respectively, meaning that the original signal $\mathbf{x}$ has been separated into two parts $\hat{\mathbf{x}}_1 = \mathbf{D}_1 \hat{\boldsymbol{\alpha}}_1$ and $\hat{\mathbf{x}}_2 = \mathbf{D}_2 \hat{\boldsymbol{\alpha}}_2$, which are in turn the approximations of $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively. For this problem structure, the block-coordinate relaxation (BCR) method [196], which was developed based on the shrinkage method [61], provides fast numerical computation. BCR only uses matrix-vector multiplications with the unitary transforms and their inverses.

The success of MCA is guaranteed if the two conditions stated in Eqs. (4.17) and (4.18) are satisfied. Thus, selecting two appropriate dictionaries $\mathbf{D}_1$ and $\mathbf{D}_2$ is an essential step in applying MCA for signal separation. For numerical reasons, $\mathbf{D}_1$ and $\mathbf{D}_2$ should also have fast forward and inverse implementations. The approach here is to choose these dictionaries from existing transforms based on experience: curvelets are used as the dictionary for graphics components and undecimated wavelets as the dictionary for text components.

Text image extraction

Supposed that an input document image $\mathbf{x}$ can be decomposed into two images of the same size as $\mathbf{x} = \mathbf{x}_t + \mathbf{x}_g$, where $\mathbf{x}_t$ and $\mathbf{x}_g$ contain text and graphics components respectively. Applying MCA on $\mathbf{x}$ with undecimated wavelets and curvelets as the two overcomplete dictionaries will result in $\hat{\mathbf{x}}_t$ and $\hat{\mathbf{x}}_g$, which are approximations of $\mathbf{x}_t$ and $\mathbf{x}_g$ respectively. As an explicit example, let $\mathbf{x}$ be the graphical image in Fig. 4.18a, then the separated text and graphics images $\hat{\mathbf{x}}_t$ and $\hat{\mathbf{x}}_g$ are given in Figs. 4.18b and 4.18c respectively. It is observed from these figures that, by using MCA, text and graphics components are not totally separated. This phenomenon has the following two possible explanations:



(a) Original image $\mathbf{x}$ (b) Text image $\hat{\mathbf{x}}_t$ (c) Graphics image $\hat{\mathbf{x}}_g$

(d) Binarized text image (e) Cleaned binarized text image (f) Area Voronoi diagram

Figure 4.18: Text extraction using morphological component analysis and some post-processing steps applied on the obtained text image: (a)→(b) and (c) by MCA, (b)→(d) by adaptive thresholding, (d)→(e) by removal of small components, (e)→(f) by distance transform.

- There exists an overlap between the two chosen dictionaries, both can represent the low-frequency contents efficiently and hence both consider these contents as theirs.
- Some graphics (like arrowheads, short curve segments) have morphological characteristics that are similar to those of text components. They are thus more likely to be represented by the dictionary defined from undecimated wavelets and thus may appear in the text image.

Since these ambiguities come from both the chosen dictionaries $\mathbf{D}_1$ and $\mathbf{D}_2$ (the overlap between them) and the input graphical document images themselves (the similarity between features), they cannot be totally avoided by using dictionaries that are optimized for these text/graphics separation task, regardless of the nature of dictionaries (e.g., pre-defined or learned). Nevertheless, in order to minimize the effect of these ambiguities in the final results, some post-processing steps presented in the next subsection are proposed to combine the extracted text components into text strings.

4.3.4 Grouping text components into text strings

In order to be able to group text components, the text image outputted from MCA (Fig. 4.18b) is first converted to binary by adaptive thresholding [87] (Fig. 4.18d) and then cleaned by removing small connected components (Fig. 4.18e). During this process, it is acknowledged that small text components like '·', ':' are also removed due to their small size. These small components, however, can be easily retrieved later, as shown later in the experimental subsection, because they are located the text zones. The remaining connected components in Fig. 4.18e are not all text

components, they may be parts of graphics, and hence an algorithm to group text components into text strings is required for a successful extraction. For this type of problems, algorithms based on Hough transform performed on the centroid of connected components [80, 219] can be used. However, to be robust, the grouping algorithm should be dependent on the style of text that exists in graphical document images. A new efficient method is proposed here to group text components in straight fonts with a belief that a large part of text in graphical document images is typeset in straight font style. The grouping criteria come from heuristics based on text components' properties: *neighborhood*, *inter-distance*, *orientation*, and *overlap*.

Neighborhood

Text components that belong to one text string need to be neighbors continuously. The neighborhood between connected components is determined by means of an area Voronoi diagram [96] (Fig. 4.18f shows the area Voronoi diagram of the binarized image in Fig. 4.18d). In this diagram, each connected component is represented by one Voronoi region that contains points that are closer to that connected component than to any other. Based on this definition and the natural perception of neighborhood, two connected components are said to be neighbors if their representing Voronoi regions are adjacent.

Inter-distance

Neighboring text components in one text string should have "close" position and their actual distance should depend on the font size. From this observation, the inter-distance $\text{dist}(g_i, g_j)$ between two neighboring text components g_i and g_j, defined as the shortest distance between the two points in their regions as

$$\text{dist}(g_i, g_j) = \min_{p \in g_i, q \in g_j} \text{dist}(p, q),$$

should satisfy the condition

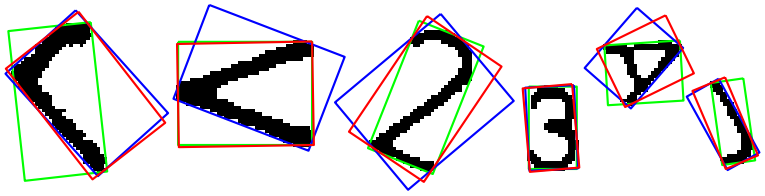
$$\text{dist}(g_i, g_j) < T_d \max\{h(g_i), h(g_j)\},$$

where $h(g)$ denotes the height of the component g. The value of T_d is determined by experience and is equal to 1.2.

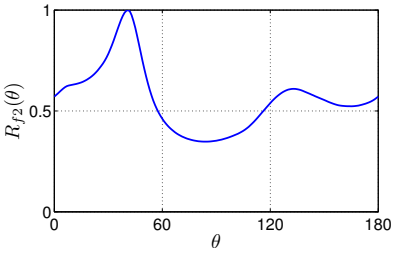
Orientation

Text components that belong to one text string need to have similar orientation. Due to the lack of a universal method for orientation estimation, the determination of the orientation of connected components resorts to both the definition of minimum-area enclosing rectangle (MAER) [83] and R-transform in Eq. (2.9). Generally, MAER (green rectangles in Fig. 4.19a) can be used to determine the orientation of most characters, however, it fails with some characters like 'A', 'r', 'J', etc. For these characters, R-transform comes as a solution by exploiting their geometrical properties.

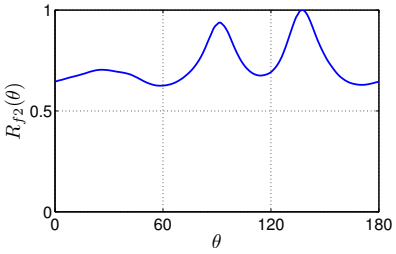
- For characters that have a dominant stroke like 'r', 'J', 'l', their R-transforms have a dominant peak that corresponds to the orientation of the dominant stroke. The problem of orientation estimation now becomes a much more simpler problem of finding the peak position in the R-transform. As an example, Fig. 4.19b shows the R-transform of the character 'r' in Fig. 4.19a. The enclosing rectangles that have orientations determined by means of the maxima of R-transforms are plotted in blue in Fig. 4.19a.



(a) Enclosing rectangles of some characters



(b) R-transform of 'r'



(c) R-transform of 'A'

Figure 4.19: Determination of text components' orientations by using the minimum-area enclosing rectangle and the R-transform. (a) MAERs are in green while rectangles determined from the R-transform by means of its maxima and correlation are in blue and red respectively. (b)–(c) the R-transforms of the two characters 'r' and 'A'.

- For symmetric characters like 'A', 'x', 'V', their R-transforms are symmetric in the angular variable. Due to this property, the orientations of these characters are determined by finding the angular points that cut their R-transforms into two vectors of the same length having the highest correlation. As an example, Fig. 4.19c shows the R-transform of the character 'A' in Fig. 4.19a. The enclosing rectangles that have orientations determined by means of correlation between two halves of the R-transforms are plotted in red in Fig. 4.19a.

Let $[o_{i1}, o_{i2}, o_{i3}]$ be the three orientations of a connected component g_i determined by the three aforementioned methods (MAER, the maxima and correlation of the R-transform). The difference in orientation between two connected components g_i and g_j is defined as

$$O_{ij} = \min_{1 \leq m,n \leq 3} |o_{im} - o_{jn}|.$$

From this definition, two neighboring connected components g_i and g_j need to have $O_{ij} \leq T_o$ to be considered as belonging to one text string. The value of T_o is determined by experience and is equal to 0.15 (radian).

Overlap

The two neighboring text components g_i and g_j of a text string need to overlap to a certain degree along their common orientation, which is defined as the orientation of the bisector t_{ij} of the angle formed by the two lines that have the same orientations as those of g_i and g_j (see Fig. 4.20). Let $[a_i, b_i]$ and $[a_j, b_j]$ be the orthogonal projections of g_i and g_j onto t_{ij} respectively, the degree of overlap of two connected components g_i and g_j is defined as

$$L_{ij} = \frac{\max\{\min(b_i - a_j, b_j - a_i), 0\}}{\min(b_i - a_i, b_j - a_j)}. \tag{4.20}$$

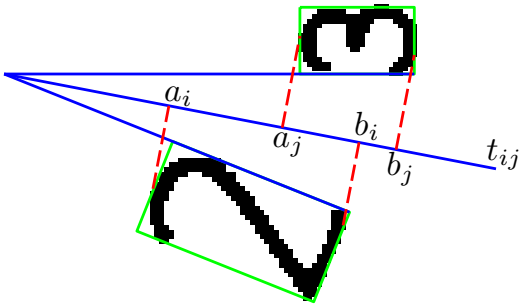


Figure 4.20: Determination of the overlap between two neighboring text components. The overlap is measured by using the their projections onto the bisector of the angle formed by the two lines that have the same orientations as their.

The numerator of Eq. (4.20) is interpreted as the length of the overlapping segment whereas the denominator is the shorter projection of the two connected components g_i and g_j onto t_{ij}. Thus two neighboring components g_i and g_j need to satisfy $L_{ij} \geq T_l$ to be considered as belonging to one text string. The value of T_l is determined by experience and is equal to 0.75.

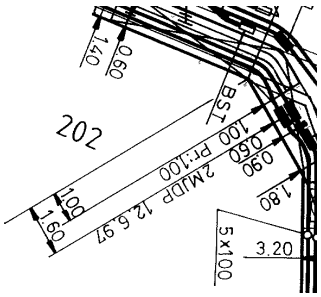
4.3.5 Experimental results

In order to demonstrate the effectiveness of the proposed method for text extraction from graphical document images, experiments have been carried out on the dataset used in [219], which contains five graphical document images of different types as shown in the first column of Fig. 4.21. The corresponding text images in grayscale obtained by applying MCA using undecimated wavelets and curvelets as the two overcomplete dictionaries for text and graphics components respectively are given in the second column of Fig. 4.21. From these text images, it is observed that:

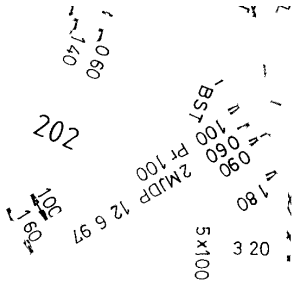
- Text components appear in good shape and they can be perceived readily. This means that text components are properly represented by undecimated wavelets as intended.
- Graphics with more global morphological characteristics (e.g., long lines, curves) do not appear in text images, meaning that they are properly represented by curvelets as intended.
- Some parts of graphics which have local morphological characteristics like those of text still remain in text images. These graphics parts are undesirably represented by undecimated wavelets.

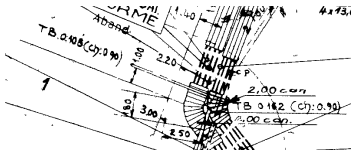
It thus can be concluded from these observations that the adopted MCA cannot totally separate text and graphics components; post-processing steps are required to minimize the effect of the remaining ambiguities on the final results.

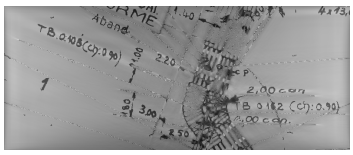
The third column of Fig. 4.21 gives the binarized images of those in the second column after removing small connected components (composed of less than 50 pixels). The obtained results demonstrate clearly that text/graphics separation using MCA overcomes the touching problem between text and graphics and is invariant to different font styles, sizes, and orientations. A quantitative evaluation of the proposed method on the five experimental images is given in Table 4.1 with the recall rate of text components is used as the evaluation measure. The column #**Texts** indicates the number of text components that exist in each graphical document image. For comparison purpose, results of the previous benchmark [219] are given in the column **Tombre**

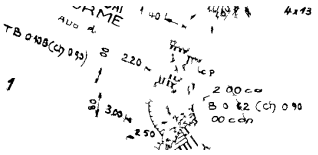

(a) Image 1: Original image

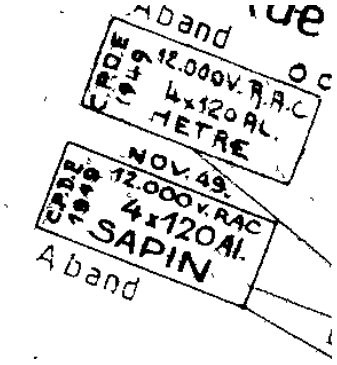

(b) Image 1: Text image

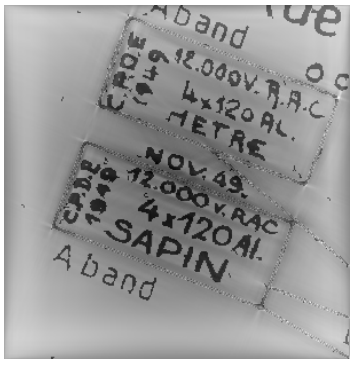

(c) Image 1: Binarized text image

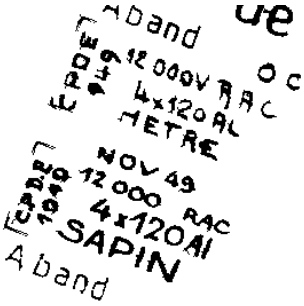

(d) Image 2: Original image

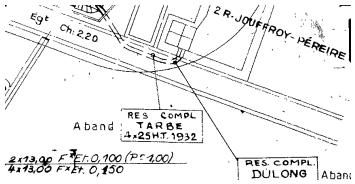

(e) Image 2: Text image

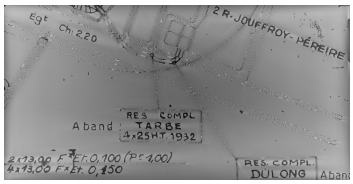

(f) Image 2: Binarized text image

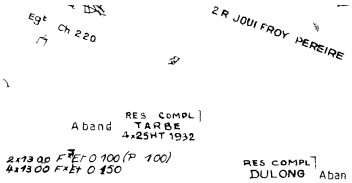

(g) Image 3: Original image

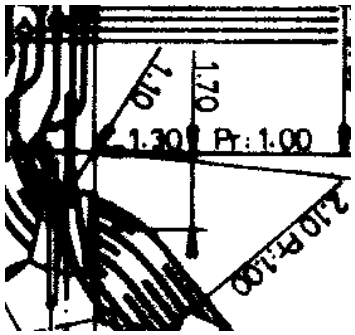

(h) Image 3: Text image

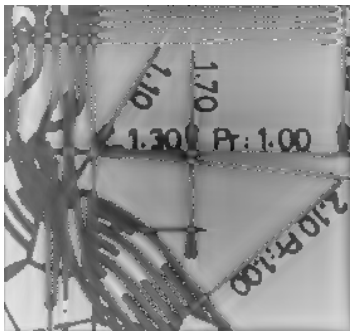

(i) Image 3: Binarized text image


(j) Image 4: Original image


(k) Image 4: Text image


(l) Image 4: Binarized text image


(m) Image 5: Original image


(n) Image 5: Text image

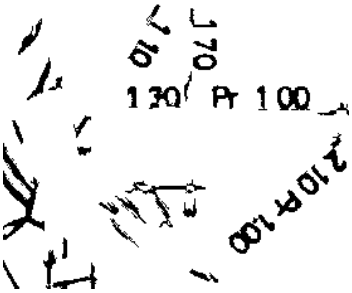

(o) Image 5: Binarized text image

Figure 4.21: Experimental results on text/graphics separation using sparse representation carried out on the dataset used in [219]. Original graphical document images are shown in the left column with the grayscale and binarized text images in the middle and right columns respectively.

Table 4.1: Performance evaluation of the proposed text extraction method in terms of recall rate of text components using the previous benchmark's dataset. Recall rate is computed separately for each graphical document image.

Images	#Texts	Tombre et al. [219]	Proposed method
1	53	49 (92.4%)	53 (100%)
2	78	59 (75.6%)	62 (79.5%)
3	78	68 (87.2%)	75 (96.2%)
4	106	92 (86.8%)	104 (98.1%)
5	21	1 (4.8%)	21 (100%)

et al. whereas results of the proposed method are in the column **Proposed method**. It can be seen that the proposed MCA-based text/graphics separation leads to a sharp increase in the recall rates for all the images, even in the "difficult" case Image 5 in Fig. 4.21m.

The technique to group text components in straight fonts into text strings has also been evaluated on the three input images, one of which is from the dataset used in [219], showed in the first row of Fig. 4.22. The grouped text strings are enclosed by red rectangles in the three corresponding images in the second row. By using the heuristic criteria proposed in Subsection 4.3.4, most of the text strings containing different characters/numbers and of different orientations have been successfully grouped. The only exception is the string 'PTT(0.60)' in Image 6 where only a portion of it, 'PTT(0', is successfully grouped. The reasons for this are the touching between '6' and '0' that changes the computed orientation and the embedment of ')' in a line. Additionally, it should be noted that, in Fig. 4.22d, the enclosing rectangles of grouped text strings are not drawn on Fig. 4.18e, but on Fig. 4.18d. The purpose of doing this is to retrieve all the small text components like '·', ':' that lie inside these enclosing rectangles and have been removed previously. This simple strategy turns out to be useful in guaranteeing a successful extraction of all text components.

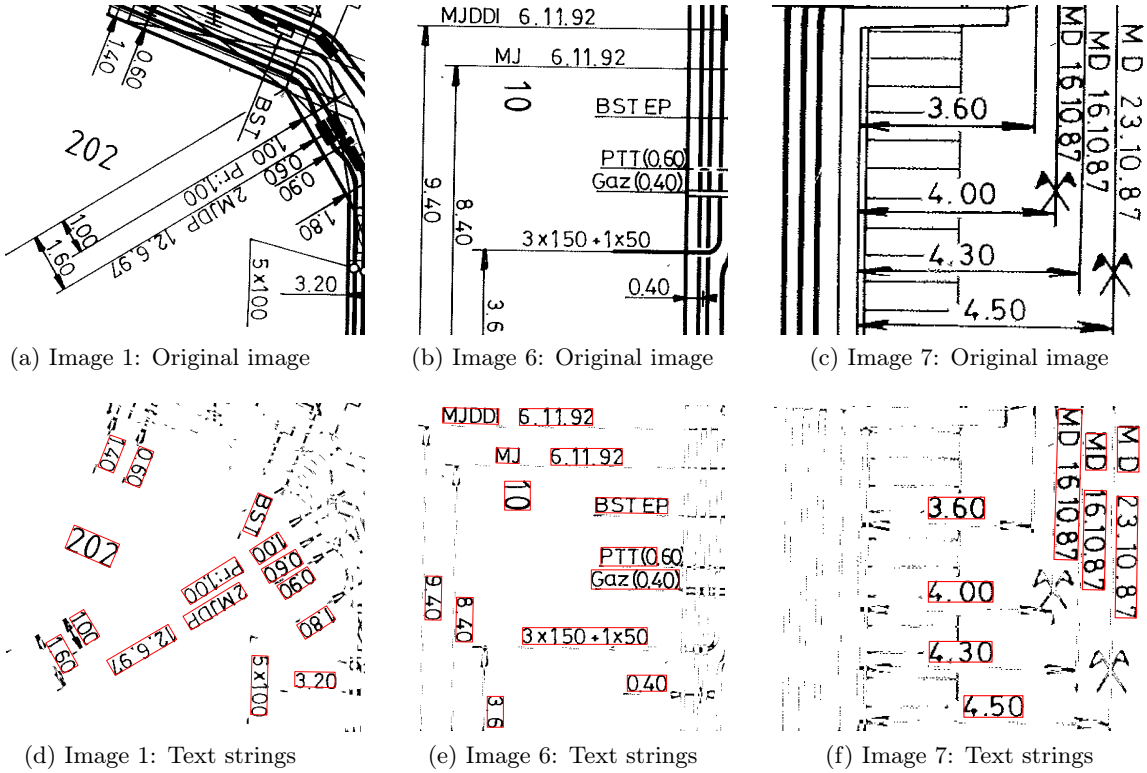
4.4 Sparse representation for classification

It is recently well-established that sparse signal models with dictionaries learned from data as in Eq. (4.7) are well suited for restoration tasks; the reported results are comparable to or even surpass the state-of-the-art in many practical signal/image processing applications. This section aims at using sparse representation for classification tasks. As the learned dictionary $\mathbf{D}$ is specifically tuned to the training data $\mathbf{X}$, a direct use of sparse representation usually does not lead to satisfactory classification results. This issue is resolved in this section by adding a discrimination term into the sparse model discussed in Section 4.1.

4.4.1 Reconstructive vs. discriminative models

In the literature, regardless of sparsity, there are two main lines of methods for classification problems: reconstruction-based methods and discrimination-based methods. These two types of methods have broad applications in classification and the difference between them has been widely investigated. It is known that:

- *Reconstructive methods*, such as principle component analysis (PCA) [110] and independent



(a) Image 1: Original image (b) Image 6: Original image (c) Image 7: Original image

(d) Image 1: Text strings (e) Image 6: Text strings (f) Image 7: Text strings

Figure 4.22: Experimental results on grouping text components in straight fonts into text strings using the criteria proposed in Subsection 4.3.4. The grouped text components in the binarized text images in the bottom row are enclosed by red rectangles.

component analysis (ICA) [46], generate representations that enable sufficient reconstruction of signals.

- *Discriminative methods*, such as linear discriminant analysis (LDA) [67, Chapter 5], produce representations that aim at maximizing the separation of signals from different classes.

- Discriminative methods often outperform the reconstructive ones in classification [17, 151].

The comparison between these two types of methods, however, assumes that the signals being classified are ideal (i.e., noiseless), complete (without missing data), and free of outliers. When this assumption does not hold, discriminative methods suffer more from signal corruptions because they contain insufficient information in order to successfully deal with corrupted data. In other words, for optimal classification performance, the representations provided by discriminative methods do not need to contain sufficient information for signal reconstruction while sufficient reconstruction is necessary for removing noise, recovering missing data, and detecting outliers. Evidence of performance degradation of discriminative methods on corrupted signals could be found in the examples in [78]. On the contrary, reconstructive methods could address these problems with some degree of success. For example, sparse representation has been shown to achieve the state-of-the-art performance in image denoising [72, 145] and in recovering missing pixels (i.e., image inpainting) [76]; PCA method with effective sub-sampling is able to detect and exclude outliers for the subsequent LDA analysis [133].

The above discussion leads to a conclusion that reconstructive and discriminative information are the two contradicting desires in signal representation: an increase in discrimination power usually has to be paid by a decrease in robustness to corrupted data, and vice versa. This observation motivates the design of a new signal representation that combines the advantages of both reconstructive and discriminative methods to address the problem of robust classification when the obtained signals are corrupted. The combination should result in a representation that contains discriminative information for classification, crucial information for signal reconstruction. In addition, sparsity is also a preferable criterion in order to have a compact representation that agrees with the Occam's razor. It should be noted here that the idea of combining discriminative and reconstructive criteria in signal representation was proposed in the machine learning community, where feature extraction is modified to include generative information to improve the robustness to noise in the training step [78].

4.4.2 Related works

From the first marriage of discrimination and reconstruction under a sparse modeling framework in [101], a number of works on using sparse representation for classification tasks were recently reported and they can be roughly classified into three main classes: sparse representation-based classification (SRC), dictionary learning-based classification (DLC), and discriminative sparse coding (DSC).

Sparse representation-based classification: This type of approaches directly uses training data as the dictionaries for sparse coding and classification. It is based on the assumption that the samples from each class form a subspace embedded in a high-dimensional signal space. As a result, by promoting sparse representation in a dictionary formed by all samples, each sample in the union of subspaces will be represented mainly by data points of the same class [73].

In order to formulate the above idea mathematically, let $\mathbf{D} = [\mathbf{D}^1, \mathbf{D}^2, \ldots, \mathbf{D}^C]$ be the set of all training samples, where $\mathbf{D}^c$ with $c = 1, \ldots, C$ is the set of training samples from class c. Given a new testing sample $\mathbf{x}$, its sparse representation $\hat{\boldsymbol{\alpha}}$ in $\mathbf{D}$ could be obtained by solving (Q_1^λ) in Eq. (4.3). The classification is then performed as

$$\text{identity}(\mathbf{x}) = \underset{c}{\text{argmin}}\ \mathcal{R}_c, \tag{4.21}$$

where $\mathcal{R}_c = \|\mathbf{x} - \mathbf{D}^c\hat{\boldsymbol{\alpha}}^c\|_2$ is the reconstruction error associated with class c and $\hat{\boldsymbol{\alpha}}^c$ is the part of $\hat{\boldsymbol{\alpha}}$ that corresponds to $\mathbf{D}^c$ (i.e., $\hat{\boldsymbol{\alpha}} = \left[\hat{\boldsymbol{\alpha}}^1; \hat{\boldsymbol{\alpha}}^2; \ldots; \hat{\boldsymbol{\alpha}}^C\right]$). This type of approaches was successfully used for face recognition [234] and motion segmentation [185] where the above assumption holds.

Dictionary learning-based classification: In cases where the above assumption on subspace does not hold, the dictionary $\mathbf{D}^c$ for each class c could be learned from the training data by solving (D_1^λ) in Eq. (4.7) and the reconstruction errors are again used for classification. This is essentially the strategy employed in [174, 203] for texture classification. However, the learned dictionaries are not suitable for classification tasks since they are only learned to faithfully represent the training samples. This issue was addressed in [143] for texture segmentation and scene analysis by using the idea that a dictionary $\mathbf{D}^c$ associated to a class c should be "good" at reconstructing this class and, at the same time, "bad" for the same purpose with the other classes. A discriminative reconstruction constraint by using the classical *softmax* discriminative cost function is added to the dictionary learning model in order to have discrimination when the classification scheme in Eq. (4.21) is used. However, the resulting dictionary learning model is not convex and does not

Algorithm 1 Simultaneous orthogonal matching pursuit (SOMP)

Input: a dictionary $\mathbf{D} = \begin{bmatrix}\mathbf{d}_1 & \mathbf{d}_2 & \dots & \mathbf{d}_p\end{bmatrix}$, a data matrix $\mathbf{X}$, the number of coefficients K
Output: a set Λ_K containing K indices, a residual matrix $\mathbf{R}_K$

1: **Initialize**: the residual $\mathbf{R}_0 = \mathbf{X}$, the index set $\Lambda_0 = \varnothing$, the iteration counter $k = 1$
2: **while** $k < K$ **do**
3: Find the index of the best-approximating atom: $\lambda_k = \text{argmax}_i \left\|\mathbf{R}_{k-1}^T \mathbf{d}_i\right\|_1$
4: Update the index set: $\Lambda_k = \Lambda_{k-1} \cup \{\lambda_k\}$
5: Determine the orthogonal projection: $\mathbf{P}_k = \mathbf{D}_{\Lambda_k}\left(\mathbf{D}_{\Lambda_k}^T \mathbf{D}_{\Lambda_k}\right)^{-1} \mathbf{D}_{\Lambda_k}^T$
6: Calculate the new residual: $\mathbf{R}_k = \mathbf{X} - \mathbf{P}_k\mathbf{X}$
7: Increment the counter: $k = k + 1$
8: **end while**

explore the discrimination capability of sparse coding coefficients. A more recent method in [183] uses an incoherence promoting term to make dictionaries associated with different classes to be as independent as possible.

Another direction is to learn a common dictionary shared by all classes, as well as a classifier of the coding coefficients for classification. Samples are classified by using the coding coefficients as the feature vector since the shared dictionary results in only a single reconstruction error. For example, the method in [144] jointly learns a single dictionary and a function adapted to the classification task for digit recognition and texture classification. Similarly, a joint learning and dictionary reconstruction method with consideration of the linear classifier performance was proposed in [175] for object and face recognition. Based on this method and the K-SVD algorithm, a method called discriminative K-SVD was proposed in [245] for face recognition.

Discriminative sparse coding: The last type of approaches pursues sparsity-based classification in the framework of dimensionality reduction where signals are projected onto a common subspace of fewer dimensions before getting classified. The purpose here is not only discrimination but also compact representations for compression and/or coding applications. This is in strong contrast with the above two types of approaches where the primary purpose is to have a high classification rate, regardless of the dimension of the coefficient vectors.

In order to have a common dictionary of a fixed number of atoms for all data samples, an algorithm for simultaneous sparse approximation called *simultaneous orthogonal matching pursuit* (SOMP) [220] (listed in Algorithm 1) is usually used. SOMP is a generalization of OMP to the case of joint signal compression and it can be extended to dimensionality reduction. It is a greedy algorithm that extracts a subset of atoms from the dictionary $\mathbf{D}$ such that all the data samples in $\mathbf{X}$ are simultaneously approximated. The basic idea of SOMP is to select in each iteration k an atom from $\mathbf{D}$ that best matches all the columns in the residual matrix $\mathbf{R}_{k-1}$, which is the difference between $\mathbf{X}$ and its projection onto the subspace formed by the already selected atoms. The atom selection process in SOMP is, however, unsupervised for reconstruction purpose.

Supervised atom selection (SAS) was proposed separately in [119] and [191] by modifying Step 3 in Algorithm 1 to include an additional term for discrimination purpose as

$$\lambda_k = \underset{i}{\text{argmax}} \left\|\mathbf{R}_{k-1}^T \mathbf{d}_i\right\|_1 + \lambda J(\mathbf{d}_i),$$

where $J(\cdot)$ is the cost function that captures the separability of different classes. The definition of $J(\cdot)$ is inspired by LDA to be the quotient between the ℓ_2-norm of within-class and between-class

scatter matrices, similar to the definition of the discriminative term in [101]. Due to the use of $J(\cdot)$ in the atom selection process, it is expected that the resulting SAS coefficients are discriminative, leading to better classification results.

Different from SAS, a novel discriminative sparse coding method is proposed in the remaining of this section by including a discrimination term in a statistical modeling framework. More formally, let the training data $\mathbf{X}$ be denoted by Ω, a statistical model M is then estimated to have the best posterior probability $\max_M p(M/\Omega) \propto p(\Omega/M)p(M)$. The model is formed from the dictionary atoms and priors: sparsity and discrimination can be expressed though priors $p(M)$, whereas reconstruction error through the likelihood $p(\Omega/M)$. This representation depends on the dictionary and its cardinality; a high cardinality improves at least the reconstruction error, but the number of coefficients increases. The optimal model thus ensures a trade-off between the model complexity and the reconstruction error. The representation is then seen as the estimation of the k-order statistical model M_k formed by the first k basis members and priors such that $\max_{M_k} p(M_k/\Omega) \propto p(\Omega/M_k)p(M_k)$. This model selection problem can be tackled by Bayesian approaches, information theoretic approaches, or variational approaches. In the next subsection, the minimum message length (MML) principle [224] is used for the selection of the optimal statistical model.

4.4.3 MML-based sparse modeling

The optimal statistical model is proposed to be derived according to MML. Compared to other information theoretic-based criteria, MML has good performance in the cases of Gamma, Dirichlet, and Gaussian distributions. In order to make explicit the quantization effects of priors, the posterior probability density function can be approximated by [26]

$$p\left(M_k/\Omega\right) \simeq \frac{f\left(\Omega/M_k\right) h\left(M_k\right)}{\sqrt{\left|F\left(M_k\right)\right|}\, \mathrm{e}^{\frac{N_k}{2}\left(1+\log\frac{1}{12}\right)}},$$

where $f\left(\Omega/M_k\right)$ is the likelihood, $h\left(M_k\right)$ the prior of the model, $\left|F\left(M_k\right)\right|$ the determinant of the Fisher information matrix [132], and N_k the number of model parameters. The MML-based model can be seen as minus the logarithm of posterior and is written as

$$\mathrm{MML}_{\Omega}\left(M_k\right) = -\log f\left(\Omega/M_k\right) - \log h\left(M_k\right) + \frac{1}{2}\log\left|F\left(M_k\right)\right| + \frac{N_k}{2}\left(1+\log\frac{1}{12}\right). \tag{4.22}$$

Let the number of classes is C, the whole statistical model M_k of order k (i.e., the dictionary $\mathbf{D}$ is composed of k atoms $\mathbf{d}_i$ with $i = 1, \ldots, k$) is formed by the C class models M_k^c, where $c = 1, \ldots, C$. Assuming that all class models have the same order to make easier the implementation of management tasks, the development of each term in the above equation will be given in sequence as follows.

Likelihood

For the likelihood, assuming a zero-mean Gaussian distribution in the reconstruction error, the likelihood for all classes Ω_c $(c = 1, \ldots, C)$ is defined as

$$f\left(\Omega/M_k\right) = \prod_{c=1}^{C} f_c\left(\Omega/M_k^c\right) = \prod_{c=1}^{C}\prod_{\mathbf{x}\in\Omega_c}\left(2\pi\sigma^2\right)^{\frac{-m}{2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}^c\|_2^2\right)$$

or

$$-\log f\left(\Omega/M_k\right) = \frac{mCN_{\mathbf{x}}}{2}\log\left(2\pi\right) + mCN_{\mathbf{x}}\log\sigma + \frac{1}{2\sigma^2}\sum_{c=1}^{C}\sum_{\mathbf{x}\in\Omega_c}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}^c\|_2^2, \tag{4.23}$$

where $M_k = \left(M_k^1, \ldots, M_k^C, \sigma\right)$ with $M_k^c = \boldsymbol{\alpha}^c = (\alpha_1^c, \ldots, \alpha_k^c)$ is the statical model of order k, $N_{\mathbf{x}}$ the number of images in each class, and m the dimension of $\mathbf{x}$. The model M_k is thus completely defined by $N_k = kC + 1$ parameters.

Priors

In the above likelihood, σ is the scaling parameter while $\boldsymbol{\alpha}^c$ for $c = 1, \ldots, C$ are location parameters. Since a prior knowledge about the scale parameter has no influence on location parameters and a prior knowledge about the location parameters of one class have no influence on the location parameters of another class, it is legitimate to assume that

$$h\left(M_k\right) = h\left(\sigma\right)\prod_{c=1}^{C}h\left(\boldsymbol{\alpha}^c\right) = h\left(\sigma\right)\prod_{c=1}^{C}\prod_{i=1}^{k}h\left(\alpha_i^c\right), \tag{4.24}$$

where the factorization $h\left(\boldsymbol{\alpha}^c\right) = \prod_{i=1}^{k}h\left(\alpha_i^c\right)$ means that atoms in the dictionary $\mathbf{D}$ are independent. This requirement is behind the ICA algorithm to ensure both the independence of features and the reduction of high order redundancy. Since the scaling parameter σ is positive, its prior is proposed to be the inverse Gamma distribution:

$$h\left(\sigma\right) = \frac{\kappa^{\gamma}}{\Gamma\left(\gamma\right)}\sigma^{-\gamma-1}\mathrm{e}^{-\frac{\kappa}{\sigma}}, \tag{4.25}$$

where κ and γ are hyper-parameters that control the shape of the distribution. For discrimination purpose, ideally the dot product of two coefficient vectors $\boldsymbol{\alpha}^c$ and $\boldsymbol{\alpha}^l$ of any two different classes c and l is zero. The prior of the location parameter $\boldsymbol{\alpha}^c$ is then proposed to be

$$h\left(\boldsymbol{\alpha}^c\right) = h_1\left(\boldsymbol{\alpha}^c\right)h_2\left(\boldsymbol{\alpha}^c\right), \tag{4.26}$$

where $h_1\left(\boldsymbol{\alpha}^c\right)$ is for sparsity and $h_2\left(\boldsymbol{\alpha}^c\right)$ for discrimination.

- $h_1\left(\boldsymbol{\alpha}^c\right)$ must be a highly peaked distribution with long and heavy tails such as the generalized Gaussian distribution

$$h_1\left(\alpha_i^c\right) = \frac{q}{2\beta\Gamma\left(\frac{1}{q}\right)}\exp\left(-\left|\frac{\alpha_i^c}{\beta}\right|^q\right), \tag{4.27}$$

  where q and β are hyper-parameters with $q > 0$ is inversely proportional to the decreasing rate of the peak.

- $h_2\left(\boldsymbol{\alpha}^c\right)$ must be high when the dot product of vectors is low, or

$$h_2\left(\boldsymbol{\alpha}^c\right) = \prod_{l\neq c}\frac{1}{\sqrt{2\pi}\iota}\exp\left(-\frac{\left(\boldsymbol{\alpha}^c\cdot\boldsymbol{\alpha}^l\right)^2}{2\iota^2}\right), \tag{4.28}$$

  where ι is a hyper-parameter.

Thanks to the hyper-parameters, the resulting representation will be a controlled trade-off between sparsity, reconstruction error, and discrimination power. From Eqs. (4.24)–(4.28), the resulting prior of the model is

$$h\left(M_k\right) = \frac{\kappa^\gamma}{\Gamma\left(\gamma\right)}\sigma^{-\gamma-1}\mathrm{e}^{-\frac{\kappa}{\sigma}}\prod_{c=1}^{C}\prod_{i=1}^{k}\frac{q}{2\beta\Gamma\left(\frac{1}{q}\right)}\exp\left(-\left|\frac{\alpha_i^c}{\beta}\right|^q\right)\prod_{c=1}^{C}\prod_{l\neq c}\frac{1}{\sqrt{2\pi}\iota}\exp\left(-\frac{\left(\boldsymbol{\alpha}^c\cdot\boldsymbol{\alpha}^l\right)^2}{2\iota^2}\right)$$

or

$$\begin{aligned}-\log h\left(M_k\right) &= -\log h\left(\sigma\right) - \sum_{c=1}^{C}\sum_{i=1}^{k}\log h\left(\alpha_i^c\right) \qquad (4.29)\\ &= -\log\frac{\kappa^\gamma}{\Gamma\left(\gamma\right)} + \left(\gamma+1\right)\log\sigma + \frac{\kappa}{\sigma} - kC\log\frac{q}{2\beta\Gamma\left(\frac{1}{q}\right)} + \sum_{c=1}^{C}\sum_{i=1}^{k}\left|\frac{\alpha_i^c}{\beta}\right|^q\\ &\quad + \frac{C\left(C-1\right)}{2}\log\left(2\pi\right) + C\left(C-1\right)\log\iota + \frac{1}{2\iota^2}\sum_{c=1}^{C}\sum_{l\neq c}\left(\boldsymbol{\alpha}^c\cdot\boldsymbol{\alpha}^l\right)^2.\end{aligned}$$

It should be noted that several existing approaches can be deduced by setting $q = 2$.

Fisher information

The independency assumption employed for priors can also be used for Fisher information, which can then be approximated by

$$F\left(M_k\right) = F\left(\sigma\right)\prod_{c=1}^{C}F\left(M_k^c\right) = F\left(\sigma\right)\prod_{c=1}^{C}\prod_{i=1}^{k}F\left(\alpha_i^c\right). \qquad (4.30)$$

The development of each multiplicative term in the above equation is done by using the definition of Fisher information as follows.

- For the scale parameter σ:

$$-\frac{\partial^2\log f\left(\Omega/M_k\right)}{\partial\sigma^2} = -\frac{mCN_{\mathbf{x}}}{\sigma^2} + \frac{3}{\sigma^4}\sum_{c=1}^{C}\sum_{\mathbf{x}\in\Omega_c}\|\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}^c\|_2^2.$$

  Since each element of $\mathbf{x}-\mathbf{D}\boldsymbol{\alpha}^c$ is approximated as a zero-mean Gaussian noise with unknown variance σ^2 then

$$F\left(\sigma\right) = \mathrm{E}\left[-\frac{\partial^2\log f\left(\Omega/M_k\right)}{\partial\sigma^2}\middle|\,\sigma\right] = -\frac{mCN_{\mathbf{x}}}{\sigma^2} + \frac{3}{\sigma^4}mCN_{\mathbf{x}}\sigma^2 = \frac{2mCN_{\mathbf{x}}}{\sigma^2}. \qquad (4.31)$$

- For the location parameters $\boldsymbol{\alpha}^c$:

$$-\frac{\partial^2\log f\left(\Omega/M_k\right)}{\partial\left(\alpha_i^c\right)^2} = \frac{1}{\sigma^2}\sum_{\mathbf{x}\in\Omega_c}\|\mathbf{d}_i\|_2^2,$$

  then

$$F\left(\alpha_i^c\right) = \mathrm{E}\left[-\frac{\partial^2\log f\left(\Omega/M_k\right)}{\partial\left(\alpha_i^c\right)^2}\middle|\,\alpha_i^c\right] = \frac{1}{\sigma^2}\sum_{\mathbf{x}\in\Omega_c}\|\mathbf{d}_i\|_2^2 = \frac{N_{\mathbf{x}}}{\sigma^2} \qquad (4.32)$$

  since $\|\mathbf{d}_i\|_2^2 = 1$.

From Eqs. (4.30)–(4.32), the resulting Fisher information of the model is

$$F\left(M_k\right) = \frac{2mCN_{\mathbf{x}}}{\sigma^2} \prod_{c=1}^{C} \prod_{i=1}^{k} \frac{N_{\mathbf{x}}}{\sigma^2}$$

or

$$\frac{1}{2} \log \left|F\left(M_k\right)\right| = \frac{1}{2} \log \left(2mC\right) - \left(kC+1\right) \log \sigma + \frac{kC+1}{2} \log N_{\mathbf{x}}. \tag{4.33}$$

Message length of the model

By substituting relevant terms in Eqs. (4.23), (4.29), and (4.33) into Eq. (4.22), the message length of the model of order k is

$$\begin{aligned}
\mathrm{MML}_{\Omega}\left(M_k\right) = {} & \frac{mCN_{\mathbf{x}}}{2} \log \left(2\pi\right) + mCN_{\mathbf{x}} \log \sigma + \frac{1}{2\sigma^2} \sum_{c=1}^{C} \sum_{\mathbf{x} \in \Omega_c} \left\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^c\right\|_2^2 \\
& - \log \frac{\kappa^{\gamma}}{\Gamma\left(\gamma\right)} + \left(\gamma+1\right) \log \sigma + \frac{\kappa}{\sigma} - kC \log \frac{q}{2\beta\Gamma\left(\frac{1}{q}\right)} + \sum_{c=1}^{C} \sum_{i=1}^{k} \left|\frac{\alpha_i^c}{\beta}\right|^q \\
& + \frac{C\left(C-1\right)}{2} \log \left(2\pi\right) + C\left(C-1\right) \log \iota + \frac{1}{2\iota^2} \sum_{c=1}^{C} \sum_{l \neq c} \left(\boldsymbol{\alpha}^c \cdot \boldsymbol{\alpha}^l\right)^2 \\
& + \frac{1}{2} \log \left(2mC\right) - \left(kC+1\right) \log \sigma + \frac{kC+1}{2} \log N_{\mathbf{x}} \\
& + \frac{kC+1}{2} \left(1 + \log \frac{1}{12}\right).
\end{aligned}$$

In this model, all the hyper-parameters are assumed to be known and the estimation of the coefficient vectors $\boldsymbol{\alpha}^c$ $(c = 1, \ldots, C)$ assumes the knowledge of $\mathbf{D}$. At a fixed dictionary $\mathbf{D}$ of k atoms, it can easily seen that the message length depends only on the following function:

$$g_{\Omega}\left(M_k, \mathbf{D}\right) = \frac{1}{2\sigma^2} \sum_{c=1}^{C} \sum_{\mathbf{x} \in \Omega_c} \left\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^c\right\|_2^2 + \sum_{c=1}^{C} \sum_{i=1}^{k} \left|\frac{\alpha_i^c}{\beta}\right|^q + \frac{1}{2\iota^2} \sum_{c=1}^{C} \sum_{l \neq c} \left(\boldsymbol{\alpha}^c \cdot \boldsymbol{\alpha}^l\right)^2, \tag{4.34}$$

which contains three terms that correspond to reconstruction error, sparsity, and discrimination power. This dependance means that the minimum message length of the model of order k can be obtained by minimizing $g_{\Omega}\left(M_k, \mathbf{D}\right)$ or, equivalently:

$$\min_{\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^C} \left[\frac{1}{2\sigma^2} \sum_{c=1}^{C} \sum_{\mathbf{x} \in \Omega_c} \left\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^c\right\|_2^2 + \sum_{c=1}^{C} \sum_{i=1}^{k} \left|\frac{\alpha_i^c}{\beta}\right|^q + \frac{1}{2\iota^2} \sum_{c=1}^{C} \sum_{l \neq c} \left(\boldsymbol{\alpha}^c \cdot \boldsymbol{\alpha}^l\right)^2 \right].$$

Note that the three hyper-parameters σ, β, and ι control the contribution of each of the three terms in the model. In addition, as the above problem reduces to the problem in Eq. (4.5) when $C = 1$, the proposed model can be interpreted as an extension of the conventional sparse model to images of different classes with a preference on the separability of their representation.

Estimation algorithm

In order to minimize $g_\Omega(M_k, \mathbf{D})$ in Eq. (4.34), let $q = 1$ to make the optimization problem convex while, at the same time, having a highly peaked distribution with long and heavy tails for sparsity (Subsection 4.1.2). The coefficient vectors for all classes, $\boldsymbol{\alpha}^c$ $(c = 1, \ldots, C)$, are estimated iteratively using a coordinate descent algorithm. The coordinate descent algorithm for solving (Q_1^λ) in Eq. (4.3) was first proposed in [85] where the objective function is optimized (exactly or approximately) with respect to one coefficient at a time while all others are kept fixed. For the case of $g_\Omega(M_k, \mathbf{D})$, its derivative with respect to coefficient i of class c, α_i^c, is

$$\begin{aligned}\frac{\partial g_\Omega(M_k, \mathbf{D})}{\partial \alpha_i^c} &= -\frac{1}{\sigma^2}\sum_{\mathbf{x}\in\Omega_c}\mathbf{d}_i^T(\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^c) + \frac{1}{\beta}\operatorname{sign}(\alpha_i^c) + \frac{2}{\iota^2}\alpha_i^c\sum_{l\neq c}\left(\alpha_i^l\right)^2 \\ &= -\frac{1}{\sigma^2}\sum_{\mathbf{x}\in\Omega_c}\left[\mathbf{d}_i^T\left(\mathbf{x} - \sum_{j\neq i}\alpha_j^c\mathbf{d}_j\right) - \alpha_i^c\right] + \frac{1}{\beta}\operatorname{sign}(\alpha_i^c) + \frac{2}{\iota^2}\alpha_i^c\sum_{l\neq c}\left(\alpha_i^l\right)^2 \\ &= -\frac{1}{\sigma^2}\sum_{\mathbf{x}\in\Omega_c}\mathbf{d}_i^T\left(\mathbf{x} - \sum_{j\neq i}\alpha_j^c\mathbf{d}_j\right) + \frac{1}{\beta}\operatorname{sign}(\alpha_i^c) + \alpha_i^c\left(\frac{N_\mathbf{x}}{\sigma^2} + \frac{2}{\iota^2}\sum_{l\neq c}\left(\alpha_i^l\right)^2\right).\end{aligned} \tag{4.35}$$

It is not difficult to see that the solution of $\frac{\partial g_\Omega(M_k,\mathbf{D})}{\partial \alpha_i^c} = 0$ could be obtained by using the soft-thresholding operator in Eq. (4.12):

$$\alpha_i^c = \mathcal{O}(t, \lambda) = \operatorname{sign}(t)(|t| - \lambda)^+, \tag{4.36}$$

where

$$\begin{aligned}t &= \left(\frac{N_\mathbf{x}}{\sigma^2} + \frac{2}{\iota^2}\sum_{l\neq c}\left(\alpha_i^l\right)^2\right)^{-1} \times \frac{1}{\sigma^2}\sum_{\mathbf{x}\in\Omega_c}\mathbf{d}_i^T\left(\mathbf{x} - \sum_{j\neq i}\alpha_j^c\mathbf{d}_j\right), \\ \lambda &= \left(\frac{N_\mathbf{x}}{\sigma^2} + \frac{2}{\iota^2}\sum_{l\neq c}\left(\alpha_i^l\right)^2\right)^{-1} \times \frac{1}{\beta}.\end{aligned}$$

This solution is used as the re-estimated value for α_i^c until convergence in the estimated values of all the coefficients are reached. A summary of the MML-based sparse modeling is given in Algorithm 2.

4.4.4 Dictionary design

For general classification problems, the dictionary $\mathbf{D}$ should be generic. In the following experiments, $\mathbf{D}$ is constructed by applying geometric transformations to a generating mother function ϕ [79]. A geometric transformation U is defined as a combination of the following three transformations:

- *Translation* by $\vec{b} = (b_1, b_2)$: $U(\vec{b})$ moves the generating function across the image

$$U(\vec{b})\phi(x, y) = \phi(x - b_1, y - b_2).$$

- *Rotation* by θ: $U(\theta)$ rotates the generating function by an angle θ

$$U(\theta)\phi(x, y) = \phi\left(\cos(\theta)x + \sin(\theta)y, \cos(\theta)y - \sin(\theta)x\right).$$

Algorithm 2 Minimum message length (MML)

Input: a dictionary $\mathbf{D} = \begin{bmatrix}\mathbf{d}_1 & \mathbf{d}_2 & \dots & \mathbf{d}_p\end{bmatrix}$, a data matrix $\mathbf{X}$, k_{min}, k_{max}, hyper-parameters
Output: a statistical model M_{k^*}, the dictionary $\mathbf{D}_{\Lambda_{k^*}}$

Initialize: $k = k_{min}$
while $k < k_{max}$ **do**
    Determine the index set Λ_k: use SOMP in Algorithm 1
    Compute the representation of $\mathbf{X}$ in $\mathbf{D}_{\Lambda_k}$: $\mathbf{A} = \left(\mathbf{D}_{\Lambda_k}^T \mathbf{D}_{\Lambda_k}\right)^{-1} \mathbf{D}_{\Lambda_k}^T \mathbf{X}$
    Compute initial value for $\boldsymbol{\alpha}^c$: average the representations in $\mathbf{D}_{\Lambda_k}$ of all $\mathbf{x} \in \Omega_c$
    while *convergence has not been reached* **do**
        Re-estimate $\boldsymbol{\alpha}^c$ by using Eq. (4.36)
    end while
    Compute the message length $\mathrm{MML}_\Omega(M_k)$: use Eq. (4.35)
    Increment the counter: $k = k + 1$
end while
Determine the minimum message length: $M_{k^*} = \mathrm{argmin}_{M_k} \mathrm{MML}_\Omega(M_k)$

- *Anisotropic scaling* by $\vec{a} = (a_1, a_2)$: $U(\vec{a})$ scales the generating function in the two directions using two separate scaling factors

$$U(\vec{a})\phi(x, y) = \phi\left(\frac{x}{a_1}, \frac{y}{a_2}\right).$$

By letting $\gamma = \{\vec{b}, \theta, \vec{a}\}$, the atom obtained from ϕ that corresponds to γ is

$$U(\gamma)\phi(x, y) = \phi(x', y'),$$

with

$$x' = \frac{\cos(\theta)(x - b_1) + \sin(\theta)(y - b_2)}{a_1},$$
$$y' = \frac{\cos(\theta)(y - b_2) - \sin(\theta)(x - b_1)}{a_2}.$$

Let Γ be the set of all γ used in the construction, the dictionary is then defined as

$$\mathbf{D} = \{U(\gamma)\phi : \gamma \in \Gamma\}.$$

$\mathbf{D}$ constructed as above is a structured dictionary that allows efficient coding since each atom in $\mathbf{D}$ can be fully described by the corresponding transformation parameters γ when ϕ is known. Moreover, the generating mother function ϕ should be selected in such a way that some preferred geometric properties exist in dictionary atoms. The following experiments uses three structured dictionaries that are generated from the following ϕ:

- *Gaussian* function:

$$\phi(x, y) = \frac{1}{\sqrt{\pi}} \exp\left(-\left(x^2 + y^2\right)\right).$$

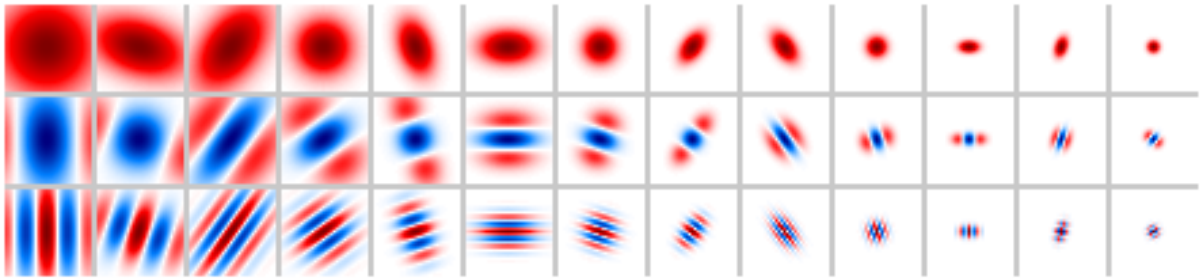


Figure 4.23: Sample atoms from the Gaussian (top row), AnR (middle row), and Gabor (bottom) dictionaries at various scales and orientations. Atoms in each column have the same set of geometric transformation parameters $\gamma = \{\vec{b} = 0, \theta, \vec{a}\}$.

- *Anisotropic refinement* (AnR) function:

$$\phi(x, y) = \frac{2}{\sqrt{3\pi}}(4x^2 - 2)\exp\left(-\left(x^2 + y^2\right)\right).$$

- *Gabor* function:

$$\phi(x, y) = \cos(2\pi x)\exp\left(-\left(x^2 + y^2\right)\right).$$

In order to have an overcomplete dictionary that spans the Hilbert space of the data of interest, the transformation parameters should be carefully sampled. The sampling of these parameters typically determines the dictionary size and therefore its redundancy. It is here proposed to use

- all possible pixel locations for the position shift $\vec{b}$,
- 10 uniformly sampled values in $[0, \pi]$ for the rotation angle θ,
- and five logarithmically equi-distributed scales in $[1, N/4]$ for the scaling factors a_1 and a_2, where N is the image size.

Sample atoms from the constructed Gaussian, AnR, and Gabor dictionaries are shown in Fig. 4.23. The depicted atoms at various scales and orientations are centered on the images ($\vec{b} = 0$).

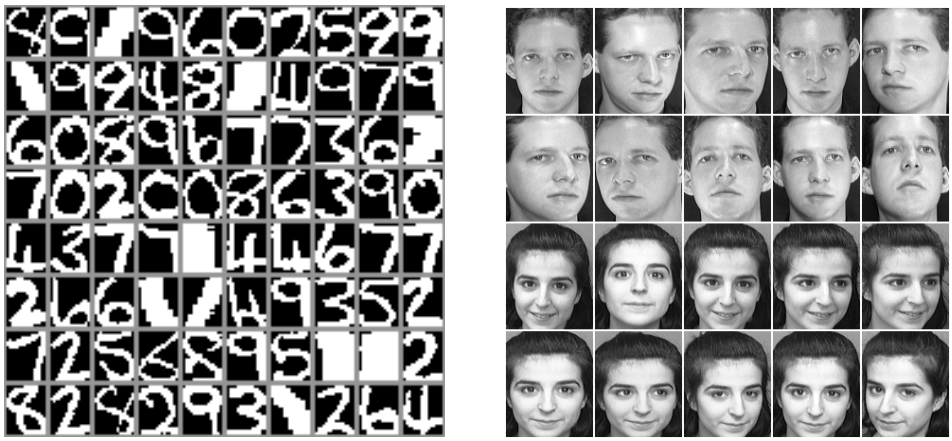
4.4.5 Experimental results

This subsection presents some validating experiments for the proposed discriminative sparse coding method based on the MML principle. It is started with the experimental setup, then the dependance of performance on the selected dictionary. The trade-off between approximation and classification is also mentioned, followed by an evaluation of the classification performance of the proposed method on common datasets. This is done in comparison with some other methods of similar nature.

Setup

Assuming that a common dictionary $\mathbf{D}$ and coefficient vectors for all classes, $\boldsymbol{\alpha}^c$ $(c = 1, \ldots, C)$, have been learned from the training data by using Algorithm 2. Each testing sample $\mathbf{x}$ is then classified by maximization of posterior probability as follows:

$$\text{identity}(\mathbf{x}) = \underset{c}{\text{argmax}}\ f_c\left(\mathbf{x}/M_k^c\right) h\left(\boldsymbol{\alpha}^c\right)$$



(a) Handwritten digit (b) ORL face

Figure 4.24: Sample images from the two datasets used in the experiments. Handwritten digit dataset has 390 images of 10 classes; in each class, 10 images is for training and the 29 remaining images for testing. ORL face dataset has 400 images of 40 subjects; for each subject, half of the images is for training and the remaining half for testing.

$$= \underset{c}{\operatorname{argmin}} \left[\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^c\|_2^2 + \frac{1}{\beta} \|\boldsymbol{\alpha}^c\|_1 + \sum_{l \neq c} \frac{\left(\boldsymbol{\alpha}^c \cdot \boldsymbol{\alpha}^l\right)^2}{2\iota^2} \right].$$

The proposed method is compared with several variants of part-based dimensionality reduction methods such as principle component analysis (PCA) [221], non-negative matrix factorization (NMF) [130], and simultaneous orthogonal matching pursuit (SOMP). For these comparison methods, both the training and testing samples are projected onto the subspace formed by k basis vectors (represented ensemble by $\mathbf{W}$) that have been learned from the training samples accordingly (i.e., $\mathbf{W} = \mathbf{D}_{\Lambda_k}$ for SOMP). In particular, the projection of a sample $\mathbf{x}$ onto $\mathbf{W}$ is given by

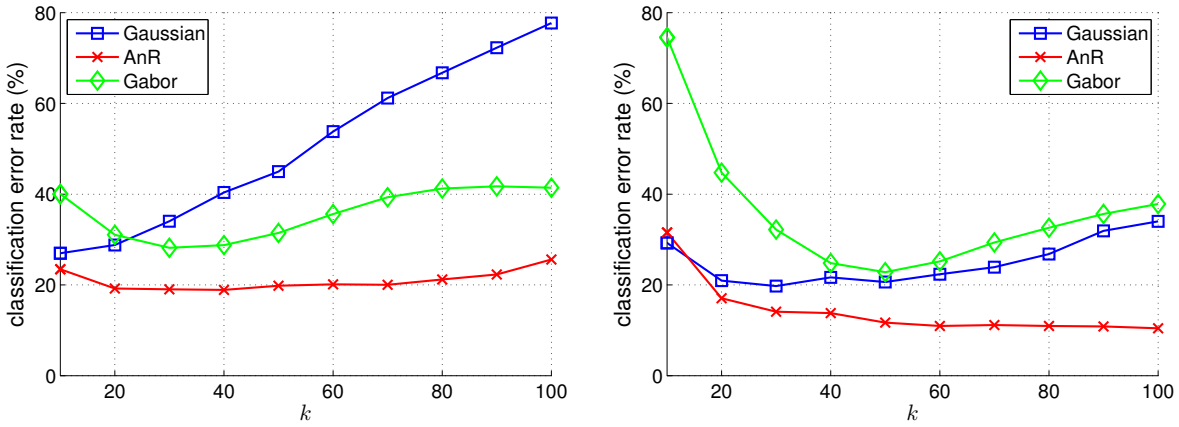
$$\boldsymbol{\alpha} = \left(\mathbf{W}^T\mathbf{W}\right)^{-1}\mathbf{W}^T\mathbf{x}.$$

The representing coefficient vector $\boldsymbol{\alpha}^c$ for each class is calculated by averaging the coefficient vectors over all the training samples of that class. The classification is performed in the reduced space by nearest neighbor classifier (kNN with $k = 1$). Each testing sample is then classified and assigned the label of its nearest neighbor among all the representing coefficient vectors. The performance is measured in terms of classification error rate, which is the percentage of the testing samples that have been misclassified. The following datasets are used in the experiments:

- Handwritten digit image dataset that is publicly available[8] contains binary images of handwritten digits 0–9, each class has 39 different samples of size 20×16. Some sample digit images are shown in Fig. 4.24a. The training set is composed of 10 randomly-selected different images per class and the testing set uses the remaining 29 images.

- ORL face dataset [195] contains facial images of 40 individuals, each has 10 different images of slightly different lighting conditions, facial expressions (smiling/non smiling), and poses.

[8] http://www.cs.nyu.edu/~roweis/data/binaryalphadigs.mat



(a) Handwritten digit (b) ORL face

Figure 4.25: Classification performance of SOMP using one of the three dictionaries (Gaussian, AnR, and Gabor) on the handwritten digit (a) and ORL face (b) datasets. AnR dictionary is superior to Gaussian and Gabor dictionaries for both datasets.

Facial images of two sample subjects are illustrated in Fig. 4.24b. The size of each image is down-sampled to 28×23 for computational efficiency. The training set is composed of five randomly-selected different images per subject and the testing set uses the remaining 5 images.

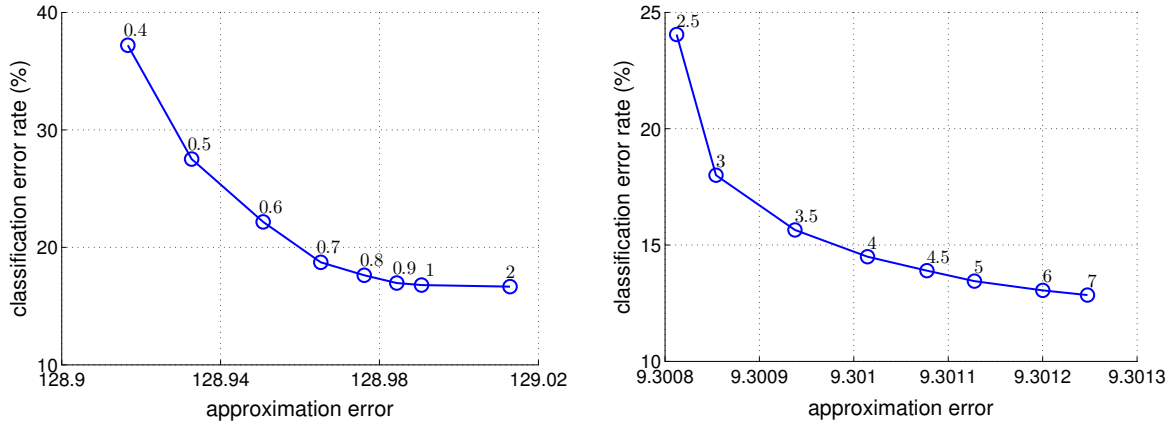
Due to the small number of training and testing samples per class for both handwritten digit and ORL face datasets, for more reliable results, the following experimental results are the average across 10 random realizations of the training/testing sets.

Dictionary choice

The impact of the dictionary on the classification performance is investigated by comparing the effectiveness of the three generating functions described above (i.e., Gaussian, AnR, and Gabor). SOMP features on both digit and face datasets are extracted and then classified in the reduced space of dimensions $k = 10, 20, \ldots, 100$. Fig. 4.25a and 4.25b depict the classification error rates obtained from the three dictionaries for the two datasets. It can be seen from the figures that the dictionary built from AnR functions results in superior performance over dictionaries built from Gaussian and Gabor functions for both datasets. This is likely due to the fact that AnR atoms, which have oscillation in the direction that is perpendicular to their orientation, are good at representing piecewise constant images like binary digit data. In addition, the slightly oscillating atoms of AnR dictionary can capture the edge-like details like eyes, mouth in the face images. The bad performance of Gabor dictionary may be because its atoms are overly oscillating. On the contrary, the lack of oscillation in Gaussian atoms explains for their inefficiency in capturing edge-like features and, consequently, for the bad performance of Gaussian dictionary. In the following experiments, AnR dictionary is therefore chosen as the dictionary for both digit and face datasets.

Approximation vs. classification

This experiment demonstrates the approximation–classification trade-off, which is driven by the relative values of the three hyper-parameters σ, β, and ι in Eq. (4.34). Fig. 4.26 illustrates the



(a) Handwritten digit (b) ORL face

Figure 4.26: Approximation–classification trade-off with MML algorithm on the handwritten digit (a) and ORL face (b) datasets. For each dataset, the values of σ and β are fixed while that of ι is varied as depicted in the figures: (a) $\sigma = 0.1$, $\beta = 5$; (b) $\sigma = 0.1$, $\beta = 10$.

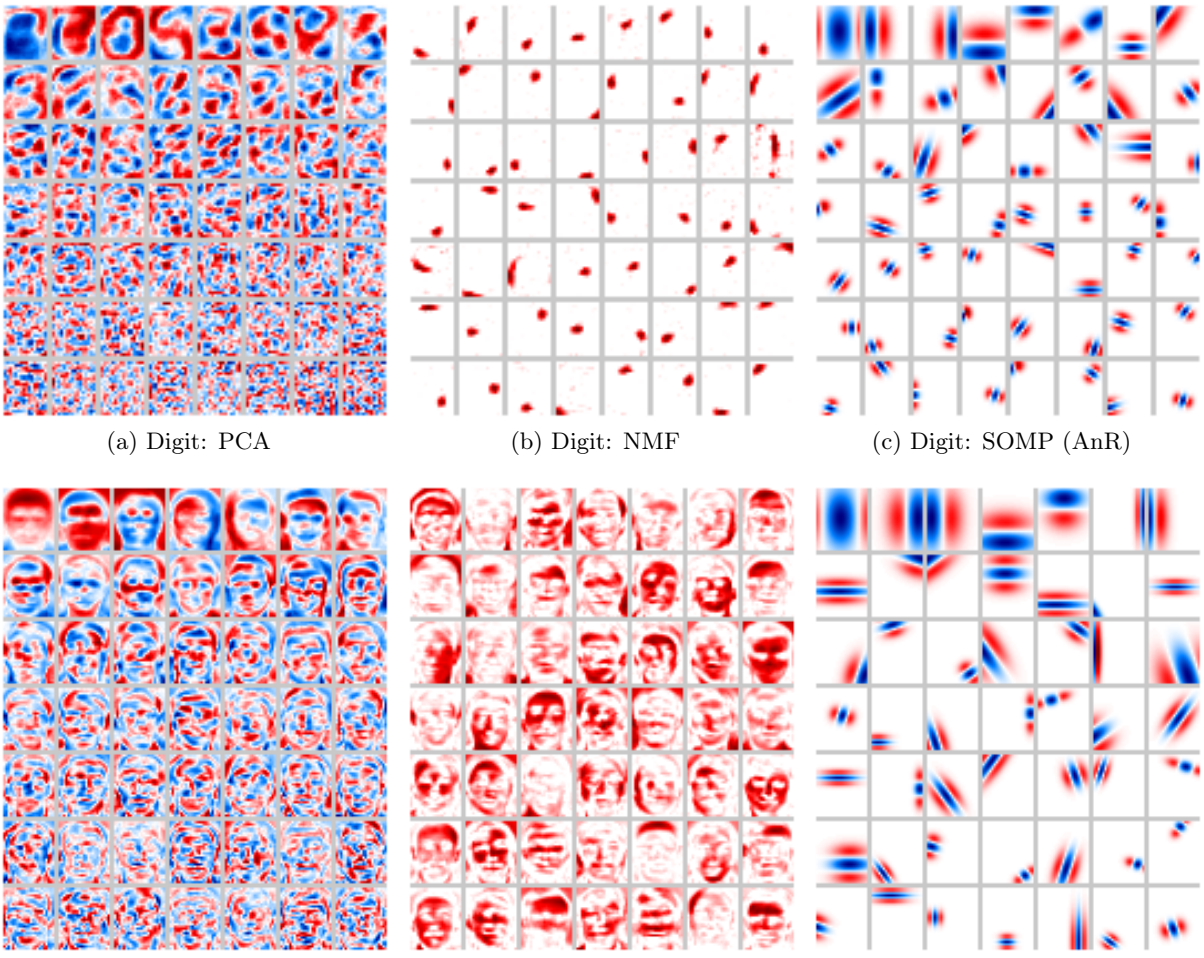
classification error rate versus the approximation error for both digit and face data. In this experiment, the dimension of the reduced space is fixed to $k = 40$. In addition, for each dataset, the values of σ and β are fixed while that of ι is varied as

- Handwritten digit dataset (Fig. 4.26a): $\sigma = 0.1$, $\beta = 5$, and $\iota \in [0.4, 2]$.
- ORL face dataset (Fig. 4.26b): $\sigma = 0.1$, $\beta = 10$, and $\iota \in [2.5, 7]$.

The approximation error is measured via the squared Euclidean norm of the residual matrix, i.e., $\sum_{c=1}^{C} \sum_{\mathbf{x} \in \Omega_c} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}^c\|_2^2$. It can be seen that, for both datasets, increasing ι leads to an increase in the approximation error and, at the same time, a decrease in the classification error rate. This demonstrates clearly the trade-off between approximation and classification in the proposed discriminative sparse coding method. A higher discrimination power has to be paid by lower representation power, and vice versa. Note that, for the approximation error in Figs. 4.26a or Fig. 4.26b, the variation in its value due to ι is quite small when compared to its average value. This is because the proposed method uses only a single coefficient vector $\boldsymbol{\alpha}^c$ for each class. A single $\boldsymbol{\alpha}^c$ cannot capture the variation among all the samples of the class it represents.

Classification performance

The proposed MML-based discriminative sparse coding method is compared with PCA, NMF, and SOMP in terms of classification performance. These methods are selected for comparison because they provide a low-rank approximation of the data and they are closely related to the proposed method (e.g., dimensionality reduction). Some samples of the basis functions obtained from PCA, NMF, and SOMP are given in Fig. 4.27 for the digit (top row) and face (bottom row) datasets. It can be observed from the figures that the basis functions obtained from PCA in Figs. 4.27a and 4.27d are of global support, whereas those from SOMP using the AnR dictionary in Figs. 4.27c and 4.27f are spatially localized. For the case of NMF, the basis functions are spatially localized for the digit dataset (Fig. 4.27b) and seem to be of global support for the face dataset (Fig. 4.27e).



(a) Digit: PCA (b) Digit: NMF (c) Digit: SOMP (AnR)

(d) Face: PCA (e) Face: NMF (f) Face: SOM (AnR)

Figure 4.27: Recovered basis functions from the handwritten digit (top row) and ORL face (bottom row) datasets: PCA (left column), NMF (middle column), SOMP (right column). For SOMP, AnR dictionary is used for both digit and face data.

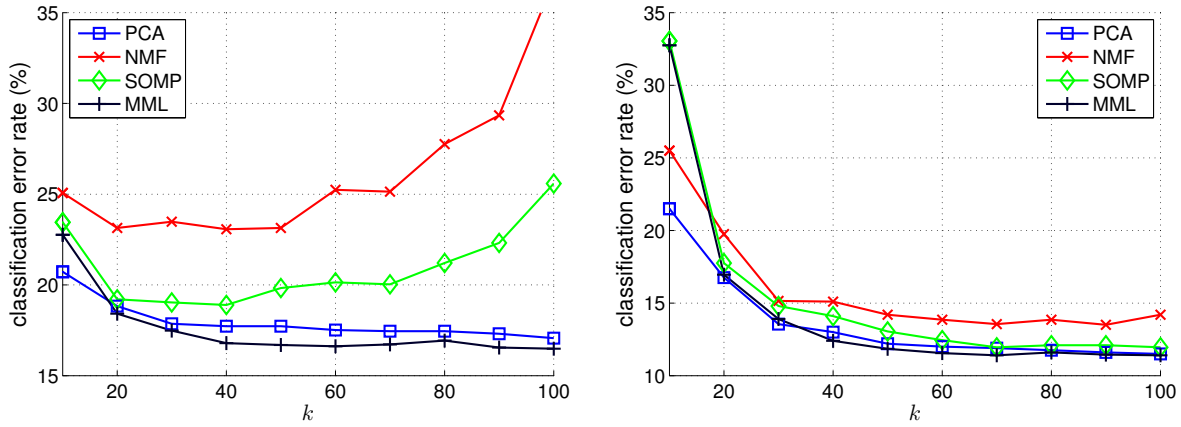
The comparison is carried out in the reduced space of dimension $k = 10, 20, \ldots, 100$. In this classification experiment, for each value of k, the classification performance is reported in terms of average error rate across 10 random realizations of the training/testing sets for both handwritten digit and ORL face datasets. For this type of experiments, the emphasis is on the classification performance then hype-parameters are selected such that overall best classification performance is obtained:

- Handwritten digit dataset: $\sigma = 0.1$, $\beta = 5$, and $\iota = 1$.

- ORL face dataset: $\sigma = 0.1$, $\beta = 10$, and $\iota = 10$.

Fig. 4.28 depicts the average classification error rates at different values of the dimension k of the reduced space for the digit and face recognition task. The proposed method is denoted by MML in the figure. It can be observed that:

- When the dimension of the reduced space is small ($k < 20$), PCA results in the best



(a) Handwritten digit (b) ORL face

Figure 4.28: Classification performance of comparison methods on the handwritten digit (a) and ORL face (b) datasets. The average classification error rate of MML is smaller than that of PCA, NMF, and SOMP for a range of dimension k.

performance. This can be explained by the global support of PCA basis functions since they can better capture important features of digit and face samples at low-dimensional space.

- In general, the performance of all methods increases with the increase in the dimension k, meaning that the discrimination power also increases with k. However, the performance of NMF and SOMP in the digit dataset degrades when $k > 50$. This is due to the inefficiency of spatially localized basis functions to high variation in the digit samples.

- Adding a discriminative term to the existing sparse coding method (SOMP) leads to an increase in classification performance. A high increase can be seen for the digit data, especially when the dimension of the reduced space is large ($k \geq 50$).

- In general, the proposed method is superior to all comparison methods. The average classification error rate of MML is smaller than that of PCA, NMF, and SOMP for a range of dimension k.

4.5 Conclusions

In this chapter, sparsity-based representation of signals/images has been presented, starting by a mathematical formulation, then by a justification for the use of the ℓ_1 regularization. The Bayesian interpretation of the framework is also given along with different strategies for the design of dictionaries, both analytical and learned from data. By promoting sparsity in the representation, important features of signals/images are usually captured by dictionary atoms and this explains for the superior performance of sparsity-based methods in some image processing tasks. Three separate problems (denoising, separation, and classification) have been tackled in this chapter by using sparse representation.

For denoising, basis pursuit denoising with dictionary being defined based on curvelets has been employed for directional denoising along graphics contours. Noise spread (NS), the output of a degradation model for graphical document images, is used as the input parameter to select

the precision parameter ϵ of the method. It has been shown both theoretically and experimentally that the proper value of ϵ turns out to have a nearly linear relationship with NS. In addition, the proposed method has been demonstrated to be superior to existing denoising methods in terms of image restoration and contour raggedness. It should be noted here that comparison using the recovery criterion is only feasible when the ground-truth images are available. Thus in this work, as a proof of concept and for the ease of evaluation, the experimental noisy images are generated from ground-truth images. To make the method applicable, in real applications, the value of NS can be estimated directly from the input images [153].

For separation, the text extraction problem has been viewed as a blind source separation problem in signal processing with text and graphics components having different morphological characteristics. Two discriminative dictionaries based on undecimated wavelets and curvelets are used to represent text and graphics components respectively. This is based on the observation that undecimated wavelets (curvelets) are good at representing text (graphics) components and bad at representing graphics (text) components. Morphological component analysis is employed for the promotion of sparse representation of text and graphics components in these two dictionaries. It has been shown that the proposed method has high recall rate of text components, overcomes the problem of text/graphics touching, and outperforms the previous benchmark. Moreover, a new technique to group text components in straight fonts into text strings has proven to be efficient.

Finally, for classification, a new discriminative sparse coding method has been proposed by modifying the conventional sparse representation framework to add a discriminative term to the model. The resulting model is a controlled trade-off between sparsity, fidelity to the data, and discrimination power. Classification is pursued in the framework of dimensionality reduction where signals are projected onto a common subspace of fewer dimensions before getting classified. The purpose here is not only discrimination but also compact representations for compression and/or coding applications. The model selection problem is tackled by using an information theoretic-based criterion (the minimum message length principle). Experimental results validate the ability to control the trade-off between approximation and classification. Moreover, the proposed method leads to superior classification performance over comparison methods of similar nature on the two common handwritten and face datasets.

Chapter 5

General Conclusion

This thesis has addressed the problem of how to represent images efficiently using dictionaries of atoms for different image processing and pattern recognition tasks. It focuses on the three specific types of dictionaries by means of the Radon transform, unit disk-based moments, and sparse representation. The intrinsic properties of each dictionary type have been discussed and are summarized in Table 5.1, where dictionaries of Radon transform and unit disk-based moments are analytical whereas those of sparse representation could be analytical or learned from data.

- *Analytical dictionaries* are characterized by mathematical models of the data. The analytical formulations lead to the existence of structure inside dictionaries. This structure sometimes results in fast implementations or other preferable properties such as orthogonality.

- *Learned dictionaries* are actually sets of realizations of the data. The learning procedure makes these dictionaries adapt to the target data. However, this adaptive property has to be paid by the loss of internal dictionary structure, which leads to the lack of fast implementation and orthogonality for this type of dictionaries.

Each type of dictionaries has been shown to be suitable for certain applications: Radon transform and unit disk-based moments for invariant recognition; sparse representation for denoising, separation, and classification. In the following, a short summary of the contents/contributions presented in this thesis regarding these three types of dictionaries is given. This is followed by a short list of open problems that can be used for future extensions.

5.1 Radon transform

The Radon transform has been used to represent patterns invariantly by employing its beneficial properties concerning RST transformations. Chapter 2 has proven theoretically that Radon transform has the property of suppressing additive white/"salt & pepper" noise. In addition, it has unified the view on possible directions for the proposal of Radon transform-based invariant pattern descriptors, leading to two novel pattern descriptors that are totally invariant to RST transformations:

- *The generic R-signature* is obtained by using an integration and then an exponentiation on the radial slices, followed by the discrete Fourier transform on the angular slices of the Radon transform data. This definition brings in a class of descriptors that has the beneficial properties of the conventional R-signature while spatially describing patterns at all directions and at different levels. This generalization gives more flexibility in definition

Table 5.1: Qualitative comparison between analytical and learned dictionaries. Dictionaries by means of Radon transform and unit disk-based moments are analytical whereas those by means of sparse representation could be analytical or learned from data.

Dictionaries	Structure	Fast computation	Adaptability	Invertibility	Orthogonality
Analytical	✓	✓/×	×	✓/×	✓/×
Learned	×	×	✓	✓	×

and, more importantly, the generic R-signature has been proven to be robust to additive noise. It has been demonstrated that the generic R-signature is superior to existing invariant pattern descriptors in terms of retrieval rate on grayscale and binary noisy datasets.

- *The RFM descriptor* is obtained by applying the 1D Fourier–Mellin and discrete Fourier transforms on the radial and angular slices of the Radon transform data respectively. It has been proven to be invariant to rotation, scaling, and translation, without the need of any normalization step. The computation of the RFM descriptor is reasonably fast and correct, based mainly on the fusion of the Radon and Fourier transforms and on a modification of the Mellin transform. It is shown to be robust to additive noise both theoretically and experimentally. It has also been demonstrated that the RFM descriptor is superior to existing invariant pattern descriptors in terms of retrieval rate on grayscale and binary noisy datasets.

5.2 Unit disk-based moments

The generalizations of existing unit disk-based orthogonal moments using harmonic functions have been pursued in Chapter 3 where the radial kernels are defined based on: Fourier series using complex exponential functions (GPCET); Fourier series using trigonometric functions (GRHFM); cosine series (GPCT); and sine series (GPST). The sets of orthogonal kernels of harmonic function-based moments have been proven to be complete in a Hilbert space of square-integrable continuous complex-valued functions. Moreover, the use of a parameter s in the definition results in four classes of moments that have beneficial properties of the original moments (PCET, RHFM, PCT, and PST) while giving more flexibility in their definitions. The usefulness of harmonic function-based moments have been demonstrated through three types of experiments:

- *Complexity*: The simple, resembling, and relating definitions of harmonic function-based kernels have resulted in an almost-constant kernel computation time, regardless of the maximal kernel order. This makes a strong contrast with Jacobi polynomial-based and eigenfunction-based methods where a higher order means a longer kernel computation time. Recursive strategies for fast computation of harmonic function-based kernels have also been proposed by exploiting the recurrence relations between harmonic functions, leading to a method that is approximately 10-time faster than direct computation and five-time faster than the current state-of-the-art strategy for fast computation of ZM kernels.

- *Representation*: Harmonic function-based methods suffer from approximation error but not from representation error. The numerical instability thus does not exist in harmonic function-based methods. Moreover, the ability to control the representation capability

according to image regions by changing the value of s draws a distinction between harmonic function-based methods and the others. Based on this ability, it is possible to have a faster reconstruction of the image function in certain image regions of interest, leading to potential applications in image compression.

- *Discrimination*: Harmonic function-based methods have been shown to generally perform better than non-orthogonal and Jacobi polynomial-based methods while having comparable performance with that of eigenfunction-based methods in rotation-invariant pattern recognition problems. Moreover, the decisive role of s in the recognition performance has been confirmed experimentally and s can be used to place emphasis of the feature vector to be extracted on certain image regions that contain discriminative information, leading to potential applications in pattern recognition.

5.3 Sparse representation

Sparsity-based representation of signals/images has been presented in Chapter 4, starting by a mathematical formulation, then by a justification for the use of the ℓ_1 regularization. The Bayesian interpretation of the framework is also given along with different strategies for the design of dictionaries, both analytical and learned from data. By promoting sparsity in the representation, important features of signals/images are usually captured by dictionary atoms and this explains for the superior performance of sparsity-based methods in some image processing tasks. Three separate problems (denoising, separation, and classification) have been tackled in this chapter by using sparse representation.

- *Denoising*: Basis pursuit denoising with dictionary being defined based on curvelets has been employed for directional denoising along graphics contours. Noise spread (NS), the output of a degradation model for graphical document images, is used as the input parameter to select the precision parameter ϵ of the method. It has been shown both theoretically and experimentally that the proper value of ϵ turns out to have a nearly linear relationship with NS. In addition, the proposed method has been demonstrated to be superior to existing denoising methods in terms of image restoration and contour raggedness.

- *Separation*: The text extraction problem has been viewed as a blind source separation problem in signal processing with text and graphics components having different morphological characteristics. Two discriminative dictionaries based on undecimated wavelets and curvelets are used to represent text and graphics components respectively. Morphological component analysis is employed for the promotion of sparse representation of text and graphics components in these two dictionaries. It has been shown that the proposed method has high recall rate of text components, overcomes the problem of text/graphics touching, and outperforms the previous benchmark. Moreover, a new technique to group text components in straight fonts into text strings has proven to be efficient.

- *Classification*: A new discriminative sparse coding method has been proposed by modifying the conventional sparse representation framework to add a discriminative term to the model. The resulting model is a controlled trade-off between sparsity, fidelity to the data, and discrimination power. Classification is pursued in the framework of dimensionality reduction where signals are projected onto a common subspace of fewer dimensions before getting classified. The model selection problem is tackled by using the minimum message length principle. The ability to control the trade-off between approximation and classification

has been validated by experiments. Moreover, the proposed method leads to superior classification performance over comparison methods of similar nature on common datasets.

5.4 Perspectives

Several issues that arise naturally from the contents of this thesis could be used as the topics for future research. While some could be addressed by straightforward extensions, the others require a considerable amount of research work before explicit conclusions can be reached. In the following, a short list of open problems is given along with some initial discussions/suggestions.

1. *The ability of the generic R-transform to encode patterns' dominant directions as discussed in Subsection 2.2.5 allows it to be employed for some other pattern recognition problems?* Intuitively, the generic R-transform could be used for applications that require the estimation of patterns' orientation. For example, a preliminary investigation on character orientation estimation using the conventional R-transform ($m = 2$) has been carried out in [95]. In addition, it is anticipated that the freedom in choosing the value of the exponent m will open up some potential applications for the generic R-transform like shape orientation estimation [247], texture analysis [106], and document image skew correction [154], etc.

2. *The extension of Radon transform and harmonic function-based moments to 3D?* Theoretical development of 3D harmonic function-based moments has been presented in Subsection 3.2.3. For 3D Radon transform, the formula in Eq. (2.1) should be slightly changed by replacing line integrals in 2D with plane integrals in 3D. In order to use these representations for different applications, such as invariant 3D pattern recognition, properties of these representations have to be made explicit a priori. For example, similar to the 2D case, the 3D Radon transform has some geometrical interpretations of the transform data that could be used for the measurement of shape properties [32].

3. *The extension of the MML-based sparse modeling framework by letting the dictionary to be learned from data, instead of being pre-defined using generating mother functions such as Gaussian, AnR, or Gabor in Subsection 4.4.4?* A dictionary learned from data means that it may be more adapted to the data and may lead to better performance, similar to the performance gain in restoration tasks due to dictionary learning. However, it should also be noted that the possible performance gain has to be paid by a loss of dictionary structure, which in turn results in the following two issues: the non-existence of fast implementation for the framework's computation procedures and the inapplicability of the framework to coding problems.

4. *Combining invariance with sparsity in a unifying representation for sparsity-based invariant pattern recognition?* Obviously, the most trivial solution for this problem is the brute force approach, in which the dictionary is composed of some generating atoms and all their RST-transformed versions. This approach has inherent limitations in both storage requirement and time complexity. A recently proposed solution uses normalizations before the sparse coding step in order to get invariance to RST transformations [11]. However, the adopted normalizations lead to issues that are similar to those due to normalizations in classical invariant pattern recognition. For this combination problem, learning both dictionary and the RST-transformed versions of its atoms directly from images as in [90] could be a good suggestion.

List of Publications

Journal articles

1. Thai V. Hoang and Salvatore Tabbone, "The generalization of the R-transform for invariant pattern representation," *Pattern Recognition*, vol. 45, no. 6, pp. 2145–2163, June 2012.

2. Thai V. Hoang and Salvatore Tabbone, "Invariant pattern recognition using the RFM descriptor," *Pattern Recognition*, vol. 45, no. 1, pp. 271–284, January, 2012.

3. Thai V. Hoang, Kwang-Hyun Park, and Z. Zenn Bien, "Development of a piano-playing robot with motion–sound mapping," *International Journal of Assistive Robotics and Mechatronics*, vol. 9, no. 3, pp. 40–51, September, 2008.

Conference papers

4. Thai V. Hoang and Salvatore Tabbone, "Generic polar harmonic transforms for invariant image description," in *Proceedings of the 18th IEEE International Conference on Image Processing* (ICIP'2011), pages 845–848, September 11–14, 2011, Brussels, Belgium.

5. Thai V. Hoang, Elisa H. Barney Smith, and Salvatore Tabbone, "Edge noise removal in bilevel graphical document images using sparse representation," in *Proceedings of the 18th IEEE International Conference on Image Processing* (ICIP'2011), pages 3610–3613, September 11–14, 2011, Brussels, Belgium.

6. Thai V. Hoang and Salvatore Tabbone, "Generic R-transform for invariant pattern representation," in *Proceedings of the 9th International Workshop on Content-Based Multimedia Indexing* (CBMI'2011), pages 157–162, June 13–15, 2011, Madrid, Spain.

7. Thai V. Hoang and Salvatore Tabbone, "A geometric invariant shape descriptor based on the Radon, Fourier, and Mellin transforms," in *Proceedings of the 20th International Conference on Pattern Recognition* (ICPR'2010), pages 2085–2088, August 23–36, 2010, Istanbul, Turkey.

8. Thai V. Hoang and Salvatore Tabbone, "Text extraction from graphical document images using sparse representation," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems* (DAS'2010), pages 143–150, June 9–11, 2010, Boston, USA.

9. Thai V. Hoang and Salvatore Tabbone, "Séparation texte/graphique à partir d'une représentation parcimonieuse," in *Actes du 11ème Colloque International Francophone sur l'Ecrit et le Document* (CIFED'2010), pages 325–340, March 18–20, 2010, Sousse, Tunisia. (in French)

10. Thai V. Hoang, Salvatore Tabbone, and Ngoc-Yen Pham, "Recognition-based segmentation of Nom characters from body text regions of stele images using area Voronoi diagram," in *Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns* (CAIP'2009), pages 205–212, September 2–4, 2009, Munster, Germany.

11. Thai V. Hoang, Salvatore Tabbone, and Ngoc-Yen Pham, "Extraction of Nom text regions from stele images using area Voronoi diagram," in *Proceedings of the 10th International Conference on Document Analysis and Recognition* (ICDAR'2009), pages 921–925, July 26–29, 2009, Barcelona, Spain.

12. Thai V. Hoang, Salvatore Tabbone, and Eric Castelli, "Un système d'indexation d'images anciennes de stèles vietnamiennes," in *Actes du dixième Colloque International Francophone sur l'Ecrit et le Document* (CIFED'2008), pages 211–212, October 28–30, 2008, Rouen, France. (in French)

13. Sunghoi Huh, Thai V. Hoang, Jens Rehr, Kwang-Hyun Park, and Z. Zenn Bien, "Development of a piano-playing robot system," in *Proceedings of the 11th International Conference on Mechatronics Technology* (ICMT'2007), pp. 274–279, November 5–9, 2007, Ulsan, Korea.

Manuscripts

14. Thai V. Hoang and Salvatore Tabbone, "The generalization of polar harmonic transforms for invariant image representation," (submitted).

15. Thai V. Hoang, Elisa H. Barney Smith, and Salvatore Tabbone, "Sparsity-based edge noise removal from bilevel graphical document images," (in preparation).

16. Thai V. Hoang and Salvatore Tabbone, "Fast polar harmonic transforms," (in preparation).

Bibliography

[1] Y. S. Abu-Mostafa and D. Psaltis, "Recognitive aspects of moment invariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 698–706, 1984.

[2] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[3] R. A. Altes, "The Fourier–Mellin transform and mammalian hearing," *Journal of the Acoustical Society of America*, vol. 63, no. 1, pp. 174–183, 1978.

[4] G. Amu, S. Hasi, X. Yang, and Z. Ping, "Image analysis by pseudo-Jacobi (p = 4, q = 3)–Fourier moments," *Applied Optics*, vol. 43, no. 10, pp. 2093–2101, 2004.

[5] G. R. Arce, J. Bacca, and J. L. Paredes, "Nonlinear filtering for image analysis and enhancement," in *The Essential Guide to Image Processing*, A. C. Bovik, Ed. Elsevier, 2009, ch. 12, pp. 263–291.

[6] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, and Y. Shkolnisky, "A framework for discrete integral transformations I – the pseudopolar Fourier transform," *SIAM Journal on Scientific Computing*, vol. 30, no. 2, pp. 764–784, 2008.

[7] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, Y. Shkolnisky, and I. Sedelnikov, "A framework for discrete integral transformations II – the 2D discrete Radon transform," *SIAM Journal on Scientific Computing*, vol. 30, no. 2, pp. 785–803, 2008.

[8] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. The ACM Press, 1999.

[9] H. S. Baird, "Document image defect models," in *Structured Document Image Analysis*, H. S. Baird, K. Yamamoto, and H. Bunke, Eds. Springer-Verlag, 1992, pp. 546–556.

[10] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[11] L. Bar and G. Sapiro, "Sparse subspace clustering," in *Proceedings of the 18th IEEE International Conference on Image Processing*, 2011, pp. 2790–2797.

[12] R. Baraniuk, E. Candès, M. Elad, and Y. Ma, "Applications of sparse representation and compressive sensing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 906–909, 2010.

[13] H. Barlow, "Possible principles underlying the transformation of sensory messages," in *Sensory Communication*. MIT Press, 1961, pp. 217–234.

[14] E. H. Barney Smith, "Modeling image degradations for improving OCR," in *Proceedings of the 16th European Signal Processing Conference*, 2008.

[15] E. H. Barney Smith and X. Qiu, "Statistical image differences, degradation features, and character distance metrics," *International Journal on Document Analysis and Recognition*, vol. 6, no. 3, pp. 146–153, 2003.

[16] A. Beck and M. Teboulle, "A fast iterative shrinkage–thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

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

[18] G. M. Bernstein and M. Jarvis, "Shapes and shears, stars and smears: optimal measurements for weak lensing," *The Astronomical Journal*, vol. 123, no. 2, p. 583, 2002.

[19] J. Bertrand, P. Bertrand, and J. P. Ovarlez, "Mellin transform," in *Transforms and Applications Handbook*, 3rd ed., A. D. Poularikas, Ed. CRC Press, 2010, ch. 12.

[20] G. Beylkin, "Discrete Radon transform," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 2, pp. 162 – 172, 1987.

[21] A. B. Bhatia and E. Wolf, "On the circle polynomials of Zernike and related orthogonal sets," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 50, pp. 40–48, 1954.

[22] M. Bober, F. Preteux, and W.-Y. Y. Kim, "Shape descriptors," in *Introduction to MPEG 7: Multimedia Content Description Language*, B. S. Manjunat, P. Salembier, and T. Sikora, Eds. John Wiley & Sons, 2002, ch. 15, pp. 231–260.

[23] C. Boncelet, "Image noise models," in *The Essential Guide to Image Processing*, A. C. Bovik, Ed. Elsevier, 2009, ch. 7, pp. 143–167.

[24] G. Borgefors, "Distance transformations in digital images," *Computer Vision, Graphics, and Image Processing*, vol. 34, no. 3, pp. 344–371, 1986.

[25] J. M. Borwein and P. B. Borwein, "On the complexity of familiar functions and numbers," *SIAM Review*, vol. 30, no. 4, pp. 589–601, 1988.

[26] N. Bouguila and D. Ziou, "Unsupervised selection of a finite Dirichlet mixture model: an MML-based approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 993–1009, 2006.

[27] F. Bowman, *Introduction to Bessel Functions*. Dover Publications, 1958.

[28] M. L. Brady, "A fast discrete approximation algorithm for the Radon transform," *SIAM Journal on Computing*, vol. 27, no. 1, pp. 107–119, 1998.

[29] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[30] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[31] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[32] D. Cailliere, F. Denis, D. Pele, and A. Baskurt, "3D mirror symmetry detection using Hough transform," in *Proceedings of the 15th IEEE International Conference on Image Processing*, 2008, pp. 1772–1775.

[33] E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities," *Communications on Pure and Applied Mathematics*, vol. 57, no. 2, pp. 219–266, 2002.

[34] E. J. Candès and F. Guo, "New multiscale transforms, minimum total variation synthesis: applications to edge-preserving image reconstruction," *Signal Processing*, vol. 82, no. 11, pp. 1519–1543, 2002.

[35] N. Canterakis, "3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition," in *Proceedings of the 11th Scandinavian Conference on Image Analysi*, 1999, pp. 85–93.

[36] R. Cao and C. L. Tan, "Text/graphics separation in maps," in *Proceedings of the 4th International Workshop on Graphics Recognition*, 2001, pp. 167–177.

[37] D. Casasent and D. Psaltis, "New optical transforms for pattern recognition," *Proceedings of the IEEE*, vol. 65, no. 1, pp. 77–84, 1977.

[38] S. S. Chandra, "Circulant Theory of the Radon Transform," Ph.D. dissertation, School of Physics, Monash University, 2010.

[39] V. Chandrasekaran, M. B. Wakin, D. Baron, and R. G. Baraniuk, "Representation and compression of multidimensional piecewise functions using surflets," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 374–400, 2009.

[40] G. Chen, T. D. Bui, and A. Krzyzak, "Invariant pattern recognition using Radon, dual-tree complex wavelet and Fourier transforms," *Pattern Recognition*, vol. 42, no. 9, pp. 2013–2019, 2009.

[41] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[42] Y. W. Chen and Y. Q. Chen, "Invariant description and retrieval of planar shapes using Radon composite features," *IEEE Transactions on Signal Processing*, vol. 56, no. 10-1, pp. 4762–4771, 2008.

[43] Z. Chen and S.-K. Sun, "A Zernike moment phase-based descriptor for local image representation and matching," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 205–219, 2010.

[44] T. S. Cho, S. Paris, W. T. Freeman, and B. K. P. Horn, "Blur kernel estimation using the Radon transform," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 241–248.

[45] C.-W. Chong, P. Raveendran, and R. Mukundan, "A comparative analysis of algorithms for fast computation of Zernike moments," *Pattern Recognition*, vol. 36, no. 3, pp. 731–742, 2003.

[46] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[47] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909–996, 1988.

[48] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.

[49] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[50] A. De Sena and D. Rocchesso, "A fast Mellin and scale transform," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–9, 2007.

[51] S. R. Deans, *The Radon Transform and Some of Its Applications*. Krieger Publishing Company, 1993.

[52] ——, "Radon and Abel Transforms," in *Transforms and Applications Handbook*, 3rd ed., A. D. Poularikas, Ed. CRC Press, 2010, ch. 8.

[53] O. Deforges and D. Barba, "A robust and multiscale document image segmentation for block line/text line structures extraction," in *Proceedings of the 12th International Conference on Pattern Recognition*, vol. 2, 1994, pp. 306–310.

[54] L. Demanet and L. Ying, "Wave atoms and sparsity of oscillatory patterns," *Applied and Computational Harmonic Analysis*, vol. 23, no. 3, pp. 368–387, 2007.

[55] S. Derrode and F. Ghorbel, "Robust and efficient Fourier–Mellin transform approximations for gray-level image reconstruction and complete invariant description," *Computer Vision and Image Understanding*, vol. 83, no. 1, pp. 57–78, 2001.

[56] M. N. Do and M. Vetterli, "The finite ridgelet transform for image representation," *IEEE Transactions on Image Processing*, vol. 12, no. 1, pp. 16–28, 2003.

[57] ——, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[58] D. S. Doermann, E. Rivlin, and I. Weiss, "Applying algebraic and differential invariants for logo recognition," *Machine Vision and Applications*, vol. 9, no. 2, pp. 73–86, 1996.

[59] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 7, pp. 797–829, 2006.

[60] D. L. Donoho and X. Huo, "Beamlets and multiscale image analysis," in *Multiscale and Multiresolution Methods*, T. J. Barth, T. Chan, and R. Haimes, Eds. Springer, 2001, pp. 149–196.

[61] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[62] D. Dori, Y. Liang, J. Dowell, and I. Chai, "Sparse-pixel recognition of primitives in engineering drawings," *Machine Vision and Applications*, vol. 6, no. 2–3, pp. 69–82, 1993.

[63] D. Dori and Y. Velkovitch, "Segmentation and recognition of dimensioning text from engineering drawings," *Computer Vision and Image Understanding*, vol. 69, no. 2, pp. 196–201, 1998.

[64] D. Dori and L. Wenyin, "Vector-based segmentation of text connected to graphics in engineering drawings," in *Proceedings of the 6th International Workshop on Structural and Syntactical Pattern Recognition*, 1996, pp. 322–331.

[65] P. Dosch and E. Valveny, "Report on the second symbol recognition contest," in *Proceedings of the 6th International Workshop on Graphics Recognition*, 2005, pp. 381–397.

[66] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[67] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[68] T. M. Dunster, "Legendre and related functions," in *NIST Handbook of Mathematical Functions*, F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, Eds. Cambridge University Press, 2010, ch. 14, pp. 351–381.

[69] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[70] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[71] M. Elad, J. Starck, P. Querre, and D. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *Applied and Computational Harmonic Analysis*, vol. 19, no. 3, pp. 340–358, 2005.

[72] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[73] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797.

[74] K. Engan, S. O. Aase, and J. H. Husøy, "Multi-frame compression: theory and design," *Signal Processing*, vol. 80, no. 10, pp. 2121–2140, 2000.

[75] H. Engels, *Numerical Quadrature and Cubature*. Academic Press, 1980.

[76] M. Fadili, J.-L. Starck, and F. Murtagh, "Inpainting and zooming using sparse representations," *The Computer Journal*, vol. 52, no. 1, pp. 64–79, 2009.

[77] C. Fefferman, "On the convergence of multiple Fourier series," *Bulletin of the American Mathematical Society*, vol. 77, no. 5, pp. 744–745, 1971.

[78] S. Fidler, D. Skocaj, and A. Leonardis, "Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 337–350, 2006.

[79] R. M. Figueras i Ventura, V. Pierre, and P. Frossard, "Low-rate and flexible image coding with redundant representations," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 726–739, 2006.

[80] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 910–918, 1988.

[81] J. Flusser, "On the independence of rotation moment invariants," *Pattern Recognition*, vol. 33, no. 9, pp. 1405–1410, 2000.

[82] J. Flusser, T. Suk, and B. Zitová, *Moments and Moment Invariants in Pattern Recognition*. John Wiley & Sons, 2009.

[83] H. Freeman and R. Shapira, "Determining the minimum-area encasing rectangle for an arbitrary closed curve," *Communications of the ACM*, vol. 18, no. 7, pp. 409–413, 1975.

[84] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[85] W. J. Fu, "Penalized regressions: the Bridge versus the Lasso," *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[86] J. Gloger, "Use of the Hough transform to separate merged text/graphics in forms," in *Proceedings of the 11th International Conference on Pattern Recognition*, vol. 1, 1992, pp. 268–271.

[87] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Prentice Hall, 2007.

[88] I. Gorodnitsky and B. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[89] W. A. Götz and H. J. Druckmüller, "A fast digital Radon transform – an efficient means for evaluating the Hough transform," *Pattern Recognition*, vol. 29, no. 4, pp. 711–718, 1996.

[90] D. B. Grimes and R. P. N. Rao, "Bilinear sparse coding for invariant vision," *Neural Computation*, vol. 17, no. 1, pp. 47–73, 2005.

[91] S. Guan, C.-H. Lai, and G. W. Wei, "Fourier–Bessel analysis of patterns in a circular domain," *Physica D: Nonlinear Phenomena*, vol. 151, no. 2-4, pp. 83–98, 2001.

[92] K. Guo and D. Labate, "Optimally sparse multidimensional representation using shearlets," *SIAM Journal on Mathematical Analysis*, vol. 39, no. 1, pp. 298–318, 2007.

[93] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[94] H. Hjouj and D. W. Kammler, "Identification of reflected, scaled, translated, and rotated objects from their Radon projections," *IEEE Transactions on Image Processing*, vol. 17, no. 3, pp. 301–310, 2008.

[95] T. V. Hoang and S. Tabbone, "Text extraction from graphical document images using sparse representation," in *Proceedings of the 9th International Workshop on Document Analysis Systems*, 2010, pp. 143–150.

[96] T. V. Hoang, S. Tabbone, and N.-Y. Pham, "Extraction of Nom text regions from stele images using area Voronoi diagram," in *Proceedings of the 10th International Conference on Document Analysis and Recognition*, 2009, pp. 921–925.

[97] P. V. C. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3 069 654, 1962.

[98] K. B. Howell, "Fourier transforms," in *Transforms and Applications Handbook*, 3rd ed., A. D. Poularikas, Ed. CRC Press, 2010, ch. 2.

[99] Y.-N. Hsu, H. H. Arsenault, and G. April, "Rotation-invariant digital pattern recognition using circular harmonic expansion," *Applied Optics*, vol. 21, no. 22, pp. 4012–4015, 1982.

[100] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[101] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, 2006, pp. 609–616.

[102] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, pp. 106–154, 1962.

[103] S.-K. Hwang and W.-Y. Kim, "A novel approach to the fast computation of Zernike moments," *Pattern Recognition*, vol. 39, no. 11, pp. 2065–2076, 2006.

[104] ISO/IEC 13660:2001, *Information technology – Office equipment – Measurement of image quality attributes for hardcopy output – Binary monochrome text and graphic images.* ISO, Geneva, Switzerland, 2001.

[105] G. Jacovitti and A. Neri, "Multiresolution circular harmonic decomposition," *IEEE Transactions on Signal Processing*, vol. 48, no. 11, pp. 3242–3247, 2000.

[106] K. Jafari-Khouzani and H. Soltanian-Zadeh, "Radon transform orientation estimation for rotation invariant texture analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 1004–1008, 2005.

[107] ——, "Rotation-invariant multiresolution texture analysis using Radon and wavelet transforms," *IEEE Transactions on Image Processing*, vol. 14, no. 6, pp. 783–795, 2005.

[108] M. H. Jansen, *Noise Reduction by Wavelet Thresholding.* Springer, 2001.

[109] L. H. Johnson, "The Shift and Scale Invariant Fourier–Mellin Transform for Radar Applications," Massachusetts Institute of Technology, Tech. Rep., 1980.

[110] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, 2002.

[111] C. Jutten and J. Herault, "Blind separation of sources, part 1: an adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1–10, 1991.

[112] A. Kadyrov and M. Petrou, "The trace transform and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 811–828, 2001.

[113] B. Kamgar-Parsi and B. Kamgar-Parsi, "Evaluation of quantization error in computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 9, pp. 929–940, 1989.

[114] M. M. Kazhdan, T. A. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proceedings of the 1st Eurographics Symposium on Geometry Processing*, 2003, pp. 156–165.

[115] B. T. Kelley and V. K. Madisetti, "The fast discrete Radon transform – I. Theory," *IEEE Transactions on Image Processing*, vol. 2, no. 3, pp. 382–400, 1993.

[116] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–497, 1990.

[117] W.-Y. Kim and Y.-S. Kim, "Robust rotation angle estimator," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 768–773, 1999.

[118] E. C. Kintner, "On the mathematical properties of the Zernike polynomials," *Optica Acta: International Journal of Optics*, vol. 23, no. 8, pp. 679–680, 1976.

[119] E. Kokiopoulou and P. Frossard, "Semantic coding by supervised dimensionality reduction," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 806–818, 2008.

[120] T. H. Koornwinder, R. Wong, R. Koekoek, and R. F. Swarttouw, "Orthogonal polynomials," in *NIST Handbook of Mathematical Functions*, F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, Eds. Cambridge University Press, 2010, ch. 15, pp. 435–484.

[121] L. G. Kotoulas and I. Andreadis, "Accurate calculation of image moments," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2028–2037, 2007.

[122] ——, "An efficient technique for the computation of ART," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 682–686, 2008.

[123] J. Kovacevic and A. Chebira, "Life beyond bases: the advent of frames (Part I)," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 86–104, 2007.

[124] C. P. Lai and R. Kasturi, "Detection of dimension sets in engineering drawings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 848–855, 1994.

[125] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1661–1666, 2000.

[126] D. X. Le, G. R. Thoma, and H. Wechsler, "Classification of binary document images into textual or nontextual data blocks using neural network models," *Machine Vision and Applications*, vol. 8, no. 5, pp. 289–304, 1995.

[127] V. F. Leavers, "Use of the Radon transform as a method of extracting information about shape in two dimensions," *Image and Vision Computing*, vol. 10, no. 2, pp. 99–107, 1992.

[128] ——, "Use of the two-dimensional Radon transform to generate a taxonomy of shape for the characterization of abrasive powder particles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1411–1423, 2000.

[129] V. F. Leavers and J. F. Boyce, "The Radon transform and its application to shape parametrization in machine vision," *Image and Vision Computing*, vol. 5, no. 2, pp. 161–166, 1987.

[130] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[131] T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[132] E. L. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed. Springer, 1998.

[133] A. Leonardis and H. Bischof, "Robust recognition using eigenimages," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 99–118, 2000.

[134] J. Li, S. K. Zhou, and R. Chellappa, "Appearance modeling using a geometric transform," *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 889–902, 2009.

[135] S. X. Liao and M. Pawlak, "On the accuracy of Zernike moments for image analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1358–1364, 1998.

[136] H. Lin, J. Si, and G. P. Abousleman, "Orthogonal rotation-invariant moments for digital image processing," *IEEE Transactions on Image Processing*, vol. 17, no. 3, pp. 272–282, 2008.

[137] C.-H. Lo and H.-S. Don, "3-D moment forms: their construction and application to object identification and positioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 10, pp. 1053–1064, 1989.

[138] Z. Lu, "Detection of text regions from digital engineering drawings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 431–439, 1998.

[139] H. Luo and R. Kasturi, "Improved directional morphological operations for separation of characters from maps/graphics," in *Proceedings of the 2nd International Workshop on Graphics Recognition*, 1997, pp. 35–47.

[140] J. Ma and G. Plonka, "The curvelet transform," *IEEE Signal Processing Magazine*, vol. 27, no. 2, pp. 118–133, 2010.

[141] P. C. Mahalanobis, "On the generalised distance in statistics," *Proceedings of the National Institute of Sciences of India*, vol. 2, no. 1, pp. 49–55, 1936.

[142] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[143] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[144] ——, "Supervised dictionary learning," in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*, 2008, pp. 1033–1040.

[145] ——, "Non-local sparse models for image restoration," in *Proceedings of the 12th IEEE International Conference on Computer Vision*, 2009, pp. 2272–2279.

[146] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[147] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed. Elsevier, 2009.

[148] P. Maragos, "Morphological filtering," in *The Essential Guide to Image Processing*, A. C. Bovik, Ed. Elsevier, 2009, ch. 13, pp. 293–321.

[149] J.-B. Martens, "The Hermite transform – theory," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 9, pp. 1595–1606, 1990.

[150] ——, "Local orientation analysis in images by means of the Hermite transform," *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1103–1116, 1997.

[151] A. M. Martínez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[152] R. Massey and A. Refregier, "Polar shapelets," *Monthly Notices of the Royal Astronomical Society*, vol. 363, no. 1, pp. 197–210, 2005.

[153] C. McGillivary, C. Hale, and E. H. Barney Smith, "Edge noise in document images," in *Proceedings of the 3rd Workshop on Analytics for Noisy Unstructured Text Data*, 2009, pp. 17–24.

[154] G. Meng, C. Pan, N. Zheng, and C. Sun, "Skew estimation of document images using bagging," *IEEE Transactions on Image Processing*, vol. 19, no. 7, pp. 1837–1846, 2010.

[155] F. G. Meyer and R. R. Coifman, "Brushlets: a tool for directional image analysis and image compression," *Applied and Computational Harmonic Analysis*, vol. 4, no. 2, pp. 147–187, 1997.

[156] R. Mukundan, S. H. Ong, and P. A. Lee, "Image analysis by Tchebichef moments," *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1357–1364, 2001.

[157] R. Mukundan and K. R. Ramakrishnan, "Fast computation of Legendre and Zernike moments," *Pattern Recognition*, vol. 28, no. 9, pp. 1433–1442, 1995.

[158] R. Mukundan and K. Ramakrishnan, *Moment Functions in Image Analysis: Theory and Applications*. World Scientific, 1998.

[159] N. Nacereddine, S. Tabbone, D. Ziou, and L. Hamami, "Shape-based image retrieval using a new descriptor based on the Radon and wavelet transforms," in *Proceedings of the 20th International Conference on Pattern Recognition*, 2010, pp. 1997–2000.

[160] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.

[161] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-20)," Department of Computer Science, Columbia University, Tech. Rep. CUCS-005-96, 1996.

[162] M. Novotni and R. Klein, "Shape retrieval using 3D Zernike descriptors," *Computer-Aided Design*, vol. 36, no. 11, pp. 1047–1062, 2004.

[163] L. O'Gorman, "Image and document processing techniques for the RightPages electronic library system," in *Proceedings of the 11th International Conference on Pattern Recognition*, vol. 2, 1992, pp. 260–263.

[164] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[165] ——, "Sparse coding with an overcomplete basis set: a strategy employed by V1?" *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1998.

[166] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals," *Proceedings of the IEEE*, vol. 69, no. 5, pp. 529–541, 1981.

[167] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[168] W. Pan, T. Bui, and C. Suen, "Text segmentation from complex background using sparse representations," in *Proceedings of the 9th International Conference on Document Analysis and Recognition*, 2007, pp. 412–416.

[169] G. Papakostas, D. Koulouriotis, and E. Karakasis, "Computation strategies of orthogonal image moments: a comparative study," *Applied Mathematics and Computation*, vol. 26, pp. 1–17, 2010.

[170] G. A. Papakostas, Y. S. Boutalis, C. Papaodysseus, and D. K. Fragoulis, "Numerical error analysis in Zernike moments computation," *Image and Vision Computing*, vol. 24, no. 9, pp. 960–969, 2006.

[171] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 4th ed. McGraw-Hill, 2002.

[172] Y. C. Pati, R. Rezaiifar, Y. C. P. R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, 1993, pp. 40–44.

[173] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[174] G. Peyré, "Sparse modeling of textures," *Journal of Mathematical Imaging and Vision*, vol. 34, no. 1, pp. 17–31, 2009.

[175] D.-S. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[176] W. Philips, "A new fast algorithm for moment computation," *Pattern Recognition*, vol. 26, no. 11, pp. 1619–1621, 1993.

[177] Z. Ping, H. Ren, J. Zou, Y. Sheng, and W. Bo, "Generic orthogonal moments: Jacobi–Fourier moments for invariant image description," *Pattern Recognition*, vol. 40, no. 4, pp. 1245–1254, 2007.

[178] Z. Ping, R. Wu, and Y. Sheng, "Image description with Chebyshev–Fourier moments," *Journal of the Optical Society of America A*, vol. 19, no. 9, pp. 1748–1754, 2002.

[179] A. D. Poularikas, Ed., *Transforms and Applications Handbook*, 3rd ed. CRC Press, 2010.

[180] W. H. Press, "Discrete Radon transform has an exact, fast inverse and generalizes to operations other than sums along lines," *Proceedings of the National Academy of Sciences*, vol. 103, no. 51, pp. 19 249–19 254, 2006.

[181] J. Radon, "On the determination of functions from their integral values along certain manifolds," *IEEE Transactions on Medical Imaging*, vol. 5, no. 4, pp. 170–176, 1986, translated by P. C. Parks from the original German text.

[182] U. Rajashekar and E. P. Simoncellis, "Multiscale denoising of photographic images," in *The Essential Guide to Image Processing*, A. C. Bovik, Ed. Elsevier, 2009, ch. 11, pp. 241–261.

[183] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3504–3508.

[184] O. Ramos-Terrades, E. Valveny, and S. Tabbone, "Optimal classifiers fusion in a non-Bayesian probabilistic framework," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1630–1644, 2009.

[185] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[186] R. Reininger and J. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Transactions on Communications*, vol. 31, no. 6, pp. 835–839, 1983.

[187] H. Ren, Z. Ping, W. Bo, W. Wu, and Y. Sheng, "Multidistortion-invariant image recognition with radial harmonic Fourier moments," *Journal of the Optical Society of America A*, vol. 20, no. 4, pp. 631–637, 2003.

[188] J. Revaud, G. Lavoué, and A. Baskurt, "Improving Zernike moments comparison for optimal similarity and rotation angle retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 627–636, 2009.

[189] J. Ricard, D. Coeurjolly, and A. Baskurt, "Generalizations of angular radial transform for 2D and 3D shape retrieval," *Pattern Recognition Letters*, vol. 26, pp. 2174–2186, 2005.

[190] G. Robbins and T. Huang, "Inverse filtering for linear shift-variant imaging systems," *Proceedings of the IEEE*, vol. 60, no. 7, pp. 862–872, 1972.

[191] F. Rodriguez and G. Sapiro, "Sparse representations for image classification: learning discriminative and reconstructive non-parametric dictionaries," University of Minnesota, IMA Preprint Series 2213, Tech. Rep., 2008.

[192] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[193] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. McGraw-Hill, 1976.

[194] ——, *Real and Complex Analysis*, 3rd ed. McGraw-Hill, 1987.

[195] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, 1994, pp. 138–142.

[196] S. Sardy, A. G. Bruce, and P. Tseng, "Block coordinate relaxation methods for nonparametric wavelet denoising," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 361–379, 2000.

[197] P. Schmid-Saugeon and A. Zakhor, "Dictionary design for matching pursuit and application to motion-compensated video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 880–886, 2004.

[198] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 550–571, 2004.

[199] Y. Sheng and J. Duvernoy, "Circular Fourier–radial Mellin descriptors for pattern recognition," *Journal of the Optical Society of America A*, vol. 3, no. 6, pp. 885–888, 1986.

[200] Y. Sheng and L. Shen, "Orthogonal Fourier–Mellin moments for invariant pattern recognition," *Journal of the Optical Society of America A*, vol. 11, no. 6, pp. 1748–1757, 1994.

[201] M. Shensa, "The discrete wavelet transform: wedding the à trous and Mallat algorithms," *IEEE Transaction on Signal Processing*, vol. 40, no. 10, pp. 2464–2482, 1992.

[202] H. Skibbe, Q. Wang, O. Ronneberger, H. Burkhardt, and M. Reisert, "Fast computation of 3D spherical Fourier harmonic descriptors – a complete orthonormal basis for a rotational invariant representation of three-dimensional objects," in *Proceedings of the IEEE International Workshop on 3-D Digital Imaging and Modeling*, 2009, pp. 1863 –1869.

[203] K. Skretting and J. H. Husøy, "Texture classification using sparse frame-based representations," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–11, 2006.

[204] R. Souvenir and K. Parrigan, "Viewpoint manifolds for action recognition," *EURASIP Journal on Image and Video Processing*, vol. 2009, pp. 1–13, 2009.

[205] J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, 2002.

[206] J.-L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Transaction on Image Processing*, vol. 14, no. 10, pp. 1570–1582, 2005.

[207] J.-L. Starck, F. Murtagh, and J. M. Fadili, *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity.* Cambridge University Press, 2010.

[208] D. Strong and T. Chan, "Edge-preserving and scale-dependent properties of total variation regularization," *Inverse Problems*, vol. 19, no. 6, pp. S165–S187, 2003.

[209] F. Su, T. Lu, R. Yang, S. Cai, and Y. Yang, "A character segmentation method for engineering drawings based on holistic and contextual constraints," in *Proceedings of the 8th International Workshop on Graphics Recognition*, 2009, pp. 280–287.

[210] S. Tabbone, O. R. Terrades, and S. Barrat, "Histogram of Radon transform. A useful descriptor for shape retrieval," in *Proceedings of the 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[211] S. Tabbone, L. Wendling, and J.-P. Salmon, "A new shape descriptor defined on the Radon transform," *Computer Vision and Image Understanding*, vol. 102, no. 1, pp. 42–51, 2006.

[212] C. L. Tan and P. O. Ng, "Text extraction using pyramid," *Pattern Recognition*, vol. 31, no. 1, pp. 63–72, 1998.

[213] M. R. Teague, "Image analysis via the general theory of moments," *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.

[214] C.-H. Teh and R. T. Chin, "On digital approximation of moment invariants," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 3, pp. 318–326, 1986.

[215] ——, "On image analysis by the methods of moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 496–513, 1988.

[216] W. M. Thorburn, "The myth of Occam's razor," *Mind*, vol. XXVII, no. 3, pp. 345–353, 1918.

[217] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[218] A. N. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems.* V. H. Winston & Sons, 1977, (F. John, Translation Editor).

[219] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch, "Text/graphics separation revisited," in *Proceedings of the 5th International Workshop on Document Analysis Systems*, 2002, pp. 200–211.

[220] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

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

[222] S. C. Verrall and R. Kakarala, "Disk-harmonic coefficients for invariant pattern recognition," *Journal of the Optical Society of America A*, vol. 15, no. 2, pp. 389–401, 1998.

[223] F. Wahl, K. Wong, and R. Casey, "Block segmentation and text extraction in mixed text/image documents," *Computer Graphics and Image Processing*, vol. 20, no. 4, pp. 375–390, 1982.

[224] C. Wallace, *Statistical and Inductive Inference by Minimum Message Length.* Springer, 2005.

[225] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.

[226] Q. Wang, O. Ronneberger, and H. Burkhardt, "Rotational invariance based on Fourier analysis in polar and spherical coordinates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1715–1722, 2009.

[227] X. Wang, B. Xiao, J.-F. Ma, and X.-L. Bi, "Scaling and rotation invariant analysis approach to object recognition based on Radon and Fourier–Mellin transforms," *Pattern Recognition*, vol. 40, no. 12, pp. 3503–3508, 2007.

[228] Y. Wang, K. Huang, and T. Tan, "Human activity recognition based on R transform," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[229] J. B. Weaver, Y. Xu, D. M. Healy, and L. D. Cromwell, "Filtering noise from images with wavelet transforms," *Magnetic Resonance in Medicine*, vol. 21, no. 2, pp. 288–295, 1991.

[230] C.-Y. Wee and P. Raveendran, "On the computational aspects of Zernike moments," *Image and Vision Computing*, vol. 25, no. 6, pp. 967–980, 2007.

[231] J. Weickert, "Coherence-enhancing diffusion filtering," *International Journal of Computer Vision*, vol. 31, no. 2–3, pp. 111–127, 1999.

[232] D. Wipf and B. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[233] J. Wood, "Invariant pattern recognition: a review," *Pattern Recognition*, vol. 29, no. 1, pp. 1–17, 1996.

[234] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[235] B. Xiao, J.-F. Ma, and X. Wang, "Image analysis by Bessel–Fourier moments," *Pattern Recognition*, vol. 43, no. 8, pp. 2620–2629, 2010.

[236] Y. Xin, M. Pawlak, and S. X. Liao, "Accurate computation of Zernike moments in polar coordinates," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 581–587, 2007.

[237] M. Yaghoobi, T. Blumensath, and M. E. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2178–2191, 2009.

[238] Z. Yang and S.-i. Kamata, "Fast polar and spherical Fourier descriptors for feature extraction," *IEICE Transactions on Information and Systems*, vol. E93-D, no. 7, pp. 1708–1715, 2010.

[239] P.-T. Yap, X. Jiang, and A. C. Kot, "Two-dimensional polar harmonic transforms for invariant image representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1259–1270, 2010.

[240] P. Yip, "Sine and cosine transforms," in *Transforms and Applications Handbook*, 3rd ed., A. D. Poularikas, Ed. CRC Press, 2010, ch. 3.

[241] D. Yu and H. Yan, "An efficient algorithm for smoothing, linearization and detection of structural feature points of binary image contours," *Pattern Recognition*, vol. 30, no. 1, pp. 57–69, 1997.

[242] F. Zernike, "Beugungstheorie des schneidenver-fahrens und seiner verbesserten form, der phasenkontrastmethode," *Physica*, vol. 1, no. 7-12, pp. 689 – 704, 1934.

[243] D. Zhang and G. Lu, "Shape-based image retrieval using generic Fourier descriptor," *Signal Processing: Image Communication*, vol. 17, no. 10, pp. 825–848, 2002.

[244] ——, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[245] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2691–2698.

[246] Z. Zhang, T. Lu, F. Su, and R. Yang, "A new text detection algorithm for content-oriented line drawing image retrieval," in *Proceedings of the 11th Pacific Rim Conference on Multimedia*, 2010, pp. 338–347.

[247] J. D. Zunic, P. L. Rosin, and L. Kopanja, "On the orientability of shapes," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3478–3487, 2006.

[248] P. E. Zwicke and I. Kiss, "A new implementation of the Mellin transform and its application to radar classification of ships," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 2, pp. 191–199, 1983.

Résumé

La pertinence d'une application de traitement de signal relève notamment du choix d'une "représentation adéquate". Par exemple, pour la reconnaissance de formes, la représentation doit mettre en évidence les propriétés salientes d'un signal; en débruitage, permettre de séparer le signal du bruit; ou encore en compression, de synthétiser fidèlement le signal d'entrée à l'aide d'un nombre réduit de coefficients. Bien que les finalités de ces quelques traitements soient distinctes, il apparait clairement que le choix de la représentation impacte sur les performances obtenues.

La représentation d'un signal implique la conception d'un ensemble génératif de signaux élémentaires, aussi appelé *dictionnaire* ou *atomes*, utilisé pour décomposer ce signal. Pendant de nombreuses années, la conception de dictionnaire a suscité un vif intérêt des chercheurs dans des domaines applicatifs variés: la transformée de Fourier a été employée pour résoudre l'équation de la chaleur; celle de Radon pour les problèmes de reconstruction; la transformée en ondelette a été introduite pour des signaux monodimensionnels présentant un nombre fini de discontinuités; la transformée en contourlet a été conçue pour représenter efficacement les signaux bidimensionnels composées de régions d'intensité homogène, à frontières lisses, etc.

Jusqu'à présent, les dictionnaires existants peuvent être regroupés en deux familles d'approches: celles s'appuyant sur des *modèles mathématiques* de données et celles concernant l'*ensemble de réalisations* des données. Les dictionnaires de la première famille sont caractérisés par une *formulation analytique*. Les coefficients obtenus dans de telles représentations d'un signal correspondent à une *transformée du signal*, qui peuvent parfois être implémentée rapidement. Les dictionnaires de la seconde famille, qui sont fréquemment des dictionnaires *surcomplets*, offrent une grande flexibilité et permettent d'être adaptés aux traitements de données spécifiques. Ils sont le fruit de travaux plus récents pour lesquels les dictionnaires sont générés à partir des données en vue de la représentation de ces dernières.

L'existence d'une multitude de dictionnaires conduit naturellement au problème de la sélection du meilleur d'entre eux pour la représentation de signaux dans un cadre applicatif donné. Ce choix doit être effectué en vertu des spécificités bénéfiques validées par les applications envisagées. En d'autres termes, c'est l'usage qui conduit à privilégier un dictionnaire. Dans ce manuscrit, trois types de dictionnaire, correspondant à autant de types de transformées/représentations, sont étudiés en vue de leur utilisation en analyse d'images et en reconnaissance de formes. Ces dictionnaires sont la *transformée de Radon*, les *moments basés sur le disque unitaire* et les *représentations parcimonieuses*. Les deux premiers dictionnaires sont employés pour la reconnaissance de formes invariantes tandis que la représentation parcimonieuse l'est pour des problèmes de débruitage, de séparation des sources d'information et de classification.

Cette thèse présentent des contributions théoriques validées par de nombreux résultats expérimentaux. Concernant la transformée de Radon, des pistes sont proposées afin d'obtenir des descripteurs de formes invariants, et conduisent à définir deux descripteurs invariants aux rotations, l'échelle et la translation. Concernant les moments basés sur le disque unitaire, nous formalisons les stratégies conduisant à l'obtention de moments orthogonaux. C'est ainsi que quatre moments harmoniques polaires génériques et des stratégies pour leurs calculs rapides sont introduits. Enfin, concernant les représentations parcimonieuses, nous proposons et validons un formalisme de représentation permettant de combiner les trois critères suivant : la parcimonie, l'erreur de reconstruction ainsi que le pouvoir discriminant en classification.

Mots-clés: représentation de l'image, transformée de Radon, moments basés sur le disque unitaire, représentation parcimonieuses, reconnaissance de formes invariantes, débruitage d'images, séparation d'images, classification.