HAL
open science

Linear Combination of multiresolution descriptors: Application to Graphics Recognition

Oriol Ramos Terrades

▶ **To cite this version:**

Oriol Ramos Terrades. Linear Combination of multiresolution descriptors: Application to Graphics Recognition. Human-Computer Interaction [cs.HC]. Université Nancy II, 2006. English. NNT : . tel-00109597

HAL Id: tel-00109597

https://theses.hal.science/tel-00109597v1

Submitted on 24 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Linear Combination of Multiresolution Descriptors: Application to Graphics Recognition

Memoria de tesis presentada per
Oriol Ramos Terrades a la
Universitat Autònoma de Barcelona i la
Université de Nancy 2
per a la obtenció del títol de
Doctor en Informàtica.

Bellaterra, September 7, 2006

Advisor: **Dr. Ernest Valveny**
Universitat Autònoma de Barcelona
Dept. Ciències de la Computació & Computer Vision Center
Co-Advisor: **Dr. Salvatore Antoine Tabbone**
Université Nancy 2
Laboratoire Lorraine en Recherche Informatique et ses Applications, LORIA

This document was typeset by the author using LaTeX 2_ε.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona and Université Nancy 2, LORIA.

Copyright © 2006 by Oriol Ramos Terrades. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the author.

ISBN 84-933652-9-7

Printed by Ediciones Gráficas Rey, S.L.

als meus pares,
a les meves germanes
i a la petita Maria

Agraïments

És sorprenent com canvia la percepció de la velocitat temps. Sembla que fos ahir quan entrava per primera vegada al CVC i tanmateix, ja han passat cinc anys (1826 dies i més de 150 mil·lions de segons) des d'aquell dia. I quan miro enrere s'em fa difícil veure d'un cop d'ull totes les persones que d'una manera o un altre han contribuït a que en aquest moments estigui escrivint aquestes línies. No em sento capaç de resumir tot aquest temps. Des de la perspectiva de les emocions viscudes, i compartides, segur que per a més d'un li seria molt més amè que no l'objecte principal d'aquest treball de recerca. Tanmateix, espero que aquestes línies serveixin per agrair un cop més tot el suport rebut en tot aquests temps.

Poques coses he tingut clares durant tot aquest temps. Una d'elles és que tot té un principi, un recorregut i un final. Des d'aquesta perspectiva, puc assegurar que la meva tesis no hauria començat si, d'una banda en Juanjo no m'hagués donat la oportunitat de començar-la en el CVC i si, d'altre banda, l'Ernest no hagués acceptat en dirigir-me. Sobre el recorregut de la tesis que dir... Un fet inqüestionable és que lliga el principi amb el final. De quina manera? Bé, aquest és un altre tema. Gairebé no m'atreveixo a afirmar res més al respecte. Ha estat un recorregut, de vegades similar a un passeig aleatori, que hauria pogut estar molt diferent al que realment a estat. De ben segur, el seu final no hauria estat tal com el veieu, sense les orientacions de l'Ernest. Si hi ha res que s'hauria pogut fer millor, i no s'ha fet, segur que és per la meva culpa.

Bien sure, cette thèse n'aurait pas été la même, si je n'étais pas allé à Nancy il y a trois ans pour un séjour au LORIA pour travailler avec le membres du équipe QGAR et plus particulièrement avec Salvatore-Antoine, qui aprés cette éxperience est devenu mon co-directeur de thèse et avec qui j'ai eu l'occasion de parier quelques bières. On a eu pas mal des discussions sur les descripteurs et il m'a appris plein choses. Par ailleurs, Nancy, est une ville connu surtout pour l'Art nouveau et sa Place Stanislas. Cependant, il a fallu que quelques mois plus tard j'y revienne avec mon collègue Joan pour y découvrir quelques trésors cachés comme le Blitz et d'autres...

Han estat moltes hores en el CVC, en aquella taula de la cantonada de la sala O-112 compartida per no pocs companys de "batalla" i presidida per una foto ben genuïna sobre l'extintor. No escric noms, sou molts els que heu passat pe aquest despatx i els que quedeu i no en vull deixar cap. No es fàcil mantenir un ambient de concentració havent-hi tant gent que entra i surt al cap del dia. Tanmateix, crec que aquest ambient hi és, compensat a més, per necessaris moments d'esbarjo. Poc m'hauria imaginat el que dóna de si quatre ninots de galetes per fer un pessebre o

les possibilitats que dóna un programa com el paint. Però aquest bon ambient no és només exclusiu del despatx, és general de tot el centre. Potser afavorit per les seves dimensions reduïdes que permet que tots ens coneguem. O potser afavorit per la sana, i mai prou freqüent, tradició de portar croissanets per esmorzar i celebrar qualsevol bon motiu amb caramels i bombons. Converses sobre el sexe dels àngels a l'hora de dinar, més o menys interessants, però sempre entretingudes i tantes altres coses que fan del CVC un bon lloc per venir a treballar.

Però hi ha tota una vida fora la tesis (i el CVC), moltes coses per fer, per veure i molta gent per conèixer. Qui m'ho hauria dit el dia que vaig anar a una reunió informativa al teatre de la plaça cívica organitzada per D-Recerca, i altres associacions de investigadors pre-doctorals que mesos més tard m'hauria "enganxat" a aquest moviment. La empenta i la convicció d'aquells, que després han estat companys, van ser els culpables. Parlo de la Maria Villarroya, la Teresa Solchaga, el Mario, la Mª del Mar, el Jordi Cabana i la Sarai. Després van venir la MJ, la Laia, la Cecilia i el Txus i els que ara continuen el Jordi de Mier, la Noemi, la Elena i la Emi i tantes d'altres persones que no he tingut la sort de conèixer però igual de convençuts.

I en l'àmbit estatal, la feina dels membres de la Federación de Jovenes Investigadores-Precarios. Para algunos, si llega, unas siglas, una camiseta naranja o una taza con el lema: "Investigar es trabajar". Para mi, un colectivo de investigadores de ámbitos muy dispares unidos por unas ideas, convicciones, proyectos comunes y pasionados por su trabajo, por investigar. Personas que destacan por su calidad humana y su compromiso personal. Realmente me siento afortunado de haber compartido parte de mi tiempo con parte de ellos como son Joaquin de Navascués, Maria José Serván, Cristina Muñoz. Marta V. y Ana Torralbo en la comisión de documentación. De compartir momentos más que intensos con Marta G., Virgilio, Pere, Pablo P., Oscar R., Jaime A., Cristina C., Aurelia, Irene y Mª del Mar. Conocer a personas incombustibles com Andrés B. , Leni, Tatiana, Elvira o Jaime M. y aquellos que han venido despues tanto o más incombustibles como los anteriores: Xosé, Nacho, Marisa y David F.

I més enllà de la recerca i sou tots. Sou uns quants els que heu patit i sofert els meus neguits i alegries. Han estat realment uns anys intensos en molts aspectes que no els hagués viscut de la mateixa manera de no haver-los compartit amb els de

"sempre". Aquells que setmana rera setmana heu estat testimonis del meu recorregut. Compartint el meu temps, entre xerrades, cerveses i somnis més o menys profunds que han fet que sempre que estava amb vosaltres em trobés a casa, entre amics. Sou el Nico, el Ruben, el Jose, la Judith, la Noelia, les "Annes" i les "Elis". I com no, el Jordi que des d'Osca... bé, millor no endisar-nos en records boirosos i muntanyencs.

Però tot i així, hi ha persones que han estat singulars. Hi ha el Gerard, l'Aura i el Virgilio amb qui he pogut xerrar no poques vegades del que feia, o deixava de fer, en la tesis. En certs moments d'incertesa he pogut discutir amb ells i la seva opinió m'ha estat de molt valor. Però també hi ha la Esther i la Anabel que mes enllà d'alguna revisió i lectura d'alguns dels treballs o la inestimable ajuda en la composició d'alguna figura, en estat en bona mesura un bon suport moral i energètic. Finalment, les meves germanes, la Marta, la Cristina i la Natalia i, molt especialment, els meus pares l'Enric i la Mª Dolors. Sense la seva ajuda, paciència i alè es poc probable que ni tan sols hagués arribat a la situació d'iniciar una tesis i molt menys d'acabar-la. Durant tots aquests anys totes cinc han estat un referent i una fons d'inspiració inacabable.

A tothom,

gràcies.

Resum

En el camp de l'Anàlisi de Documents voldríem ser capaços de processar automàticament qualsevol tipus de document digital i d'extreure la informació rellevant. És a dir, voldríem conèixer la configuració del document, identificar cadascuna de les seves parts i reconèixer els seus continguts; per a poder fer cerques entre les components del document, però també, per fer cerques entre documents diferents. Aquest és un problema difícil que ha motivat diferents línies de recerca a diferents nivells. S'ha desenvolupat tot una sèrie de tècniques destinades a pre-processar la imatge per augmentar la seva qualitat, reduint el soroll dels sistemes d'adquisició i minimitzant els efectes de la degradació dels documents. També trobem molts treballs en la segmentació destinats a separar les àrees d'interès de la resta del document. Finalment, des de finals dels anys 60 fins a l'actualitat s'han proposat molts tipus descriptors que pretenen representar i identificar aquestes àrees d'interès.

En aquesta tesis ens hem centrat en el darrer d'aquests problemes, la descripció de formes però tambéen la fusió de classificadors per a aplicar-los a una de les apliacions de l'Anàlisi de Documents, el reconeixement de símbols gràfics. En el reconeixement de formes, moltes aplicacions han de fer front al problema de descriure un conjunt gran i complex de formes per a reconèixer-les, o per a recuperar-les de gran bases de dades. En alguns casos, a més del gran nombre de formes, podem trobar altres dificultats com són la semblança entre formes o la variabilitat de classes de símbols. En aquest casos, un punt clau en el procés de reconeixement de formes és la definició de descriptors de gran capacitat de discriminació. Malauradament, un sol tipus de descriptors no sol ser suficient per aconseguir resultats satisfactoris i per tant, hem de combinar la informació provinent de diferents fonts per a millorar el comportament global del sistema de reconeixement. Aquesta combinació de la informació la hem realitzat a través de la fusió de classificadors.

En relació a la descripció de formes, tradicionalment els símbols gràfics s'han representat mitjançant descriptors estructurals, construïts a partir d'una representació vectorial. Els mètodes de vectorització són sensibles al soroll i a les distorsions dels símbols esboçats. Podem intentar evitar aquest problema definint gramàtiques o construint models deformables dels símbols. Una altra possibilitat, la que hem seguit en aquest treball, és fer servir descriptors que no necessiten d'una representació vectorial. En el context de la descripció de formes hem proposat un descriptor basat en la transformada de crestetes —en anglés "ridgelets"— que, gràcies a que hem unificat

la terminologia i hem introduït un vocabulari per explicar i classificar els descriptors, podem definir com: multiresolució, polar, 2D, que conserva la informació i invariant a les similituds. D'altre banda, la propietat de multiresolució de la transformada de crestetes fa que obtinguem una representació en diferents nivells de resolució que ens permet dividir-la en grups de coeficients de crestetes que es poden considerar com a descriptors. D'aquesta manera, hem entrenat un classificador per a cada descriptor, i hem proposat unes regles de combinació lineals, IN i DN, que minimitzen l'error de classificació per aquells classificadors que compleixin un conjunt de restriccions, relatives a la distribució i dependència dels classificadors.

Aquests enfocs teòrics han estat avaluats a partir d'un conjunt d'experiments que ens han donat els següents resultats: Els descriptors de crestetes descriuen millor els símbols que altres descriptors més genèrics. Els mètodes IN i DN redueixen l'error de classificació en relació a d'altres mètodes de referència. Per últim, el mètode IN aplicat als descriptors de crestetes, en combinació amb classificadors de tipus "boosting" aconsegueix uns encerts de reconeixement propers als 100% en les proves definides per a la base de dades de símbols gràfics del GREC'03.

Paraules clau: *descriptors de forma, fusió de classificadors, descriptors multiresolució, transformada de crestetes, operador d'agregació lineals.*

Résumé

Dans le domaine de l'analyse de documents on voudrait être capable de traiter automatiquement n'importe quel genre de documents numériques et d'extraire l' information la plus importante. Plus précisément, on voudrait connaître la configuration du document,identifier chacune de ses parties et reconnaître ses contenus, pour faire des requêtes par le contenu du document lui-même mais aussi, parmi des documents différents. Ceci est une problème difficile qui a suscité un nombre important de travaux à différents niveaux. On a développé un ensemble de techniques destinés à pré-traiter les images numériques afin d'augmenter leurs qualités, en réduisant le bruit provenant des systèmes d'acquisition et en minimisant les effets de la dégradation des documents. On trouve aussi, beaucoup de travaux destinés à la segmentation de zones d'intérêts du fond du document. Finalement, depuis les années 60 à aujourd'hui un nombre important des descripteurs on été proposé pour représenter ces zones d'intérêts.

Dans ce thèse, nous avons travaillé sur la description des formes et la fusion de classificateurs pour les appliquer à la reconnaissance de graphiques. Dans la reconnaissance de formes, beaucoup d'applications sont confrontées au problème de description de grands ensembles de formes complexes pour les reconnaître, mais aussi pour les identifier dans des grandes bases de données. En plus du nombre important de formes on doit également faire face aux problèmes de similitude des formes ou de variabilité des classes des symboles. Dans ces cas, un point clé dans le processus de la reconnaissance des formes est la définition de descripteurs ayant une grande capacité de discrimination. Malheureusement, un seul descripteur ne suffit pas pour obtenir des résultats satisfaisants et donc, nous devons combiner l'information provenant de différentes sources pour améliorer le comportement global du système de reconnaissance. Cette combinaison est réalisée par un mécanisme de fusion des classificateurs.

Par rapport aux descriptions des formes, traditionnellement les symboles graphiques ont été représentés par des descripteurs structurelles, construits à partir d'une représentation vectorielle. Les méthodes de vectorisation sont sensibles aux bruits et aux distorsions des symboles ébauchés. On peut essayer de contourner ce problème en définissant une grammaire de descripteurs ou en construisant des modèles déformables des symboles. Une autre possibilité, celle poursuivie dans ce mémoire, est d'utiliser des descripteurs que n'ont pas besoin d'une représentation vectorielle. Dans le contexte de la description des formes on a proposé un descripteur basé sur la transformation de ridgelets qu'on peut définir comme: multiresolution, polaire, en 2D et qui préserve l'information

d'invariance aux similitudes. D'un autre coté, malgré qu'on puisse considérer ce descripteur comme un seul, il nous offre une représentation des formes permettant de la décomposer en groupes de coefficients de ridgelets qui sont chacun définis comme un descripteur. De cette manière, pour chaque descripteur, nous avons entraîné des classifieurs qui sont combinés linéairement en utilisant des règles de combinaison: IN (Indépendant et Normale) et DN (Dépendant et Normal), que minimisent l'erreur de classification pour ces classifieurs par rapport à un ensemble de contraintes.

Ces développements théoriques ont été validés à partir d'un ensemble de résultats expérimentaux. Les descripteurs ridgelets décrivent mieux les symboles que d'autres descripteurs plus classiques. Les règles de fusion IN et DN réduisent l'erreur de classification par rapport aux autres méthodes de références. Enfin, la méthode IN appliquée aux descripteurs de ridgelets, en combinaison avec des classificateurs du genre "boosting", aboutie à un taux de reconnaissance d'environ 100% sur la base de données définies au workshop GREC'03.

Mots clés: *Descripteurs de forme, Transformée de ridgelets, fusion de classificateurs, descripteurs multiresolution, opérateurs linéaires d'agrégation.*

Abstract

In the field of Document Analysis we would like to be able to automatically process any kind of digital document. We mean extracting the document layout and identifying each of its parts, recognising its contents and organising them in order to make searches of its components, through the document itself, but also through different documents. This is a challenger problem that has motivated different lines of research in the field of Document Analysis at different levels: Pre-processing techniques have been developed to upgrade the quality of the document image, reducing noise from the input devices and minimizing the effects of the degradation of documents. A deep study in segmentation has been carried out in order to separate the regions of interest from the document background. Finally, many descriptors have been proposed for representing and identifying these regions of interest since the end of 60s until now.

In this thesis, we have focused on, this last problem, the shape description description and also on classifier fusion, to apply them to one of the application fields in the Document Analysis: the graphics recognition. In shape recognition, many applications have to face the problem of describing a large number of complex shapes for recognition or retrieval in large databases. Besides the large number of shapes, we can find other challenges for shape description, such as the similarity among some of the shapes or the variability of the shape classes. In these cases, one of the key issues is the design of highly discriminant shape descriptors. Unfortunately, one kind of descriptor is not usually enough to achieve satisfactory results and hence, we have to combine the information from different sources to improve the global performance of the recognition system. We have carried out this combination of information using classifier fusion.

Concerning shape description, traditionally graphics have been represented using structural descriptors, which are based on a vectorial representation of the shape. Vectorization is quite sensitive to noise and to distortions of sketched symbols. We can try to overcome this problem using grammar descriptors or deformable models of shapes. Another possibility, which is the followed in this dissertation, is to propose descriptors that do not need a vectorial representation of the symbol. Thereby, in the context of shape description, we have proposed a descriptor based on the ridgelets transform which, thanks to we have unified the terminology used in shape description and the introduced vocabulary, we can define as: 2D, polar and multi-resolution descriptor information preserving and invariant to similarities. On the other hand,

although ridgelets descriptor can be considered as a single descriptor, it offers a shape representation divided into groups of coefficients, which permit us to consider them as single descriptors. Thus, for each descriptor, we have trained a classifier and we have proposed two linear combination rules, IN and DN, that minimize the classification error of classifiers verifying a set of constraints concerning the dependence and the distribtuion of classifers.

These theoretical approaches have been evaluated through an experimental evaluation in ridgelets descriptors, classifier fusion and applying the classifier fusion methods to ridge lets descriptors, obtaining the following results: Ridgelets descriptors have proven to represent graphics symbols better than general purpose descriptors. IN and DN methods reduce the misclassification rates regarding other reference fusion methods. Finally, the IN method applied to ridgelets descriptor, in combination of boosting algorithms, has reached recognition rates near to 100% in the test defined for the GREC'03 database.

Keywords: *shape descriptors, classifier fusion, multi-resolution descriptors, ridgelets transform, linear aggregation operators.*

Contents

Agraïments **i**

Resum **v**

Résumé **vii**

Abstract **ix**

1 Introduction **1**
1.1 Overview of Document Analysis 1
1.1.1 Shape Recognition 4
1.1.2 Classifier Fusion 6
1.2 Objectives and contribution of the present dissertation 7
1.2.1 Goals of the work 8
1.2.2 Framework of the thesis 10
1.3 Organization 11

2 Shape Descriptors **13**
2.1 Introduction 13
2.2 Definitions 17
2.3 Taxonomy of Descriptors 18
2.3.1 Primitives of Descriptors 20
2.3.2 Multiresolution methods 25
2.3.3 Structural descriptors 26
2.4 Discussion 27

3 Ridgelets descriptors **29**
3.1 Introduction 29
3.2 Theoretical Framework 31
3.2.1 Radon Transform 32
3.2.2 Multiresolution Analysis 36
3.3 Ridgelets Transform 38
3.3.1 Computation of Ridgelets coefficients 39
3.4 Image representation 40
3.4.1 Definition of a shape model 41

3.4.2 Definition of a similarity measure 43
3.4.3 Definition of a combination rule 44
3.5 Local Norm descriptors based on ridgelets transform 44
3.6 Discussion 47

4 Classifier Fusion 49
4.1 Introduction 49
4.2 Classifier Fusion approaches 50
4.2.1 Bayesian approach 51
4.2.2 Additive models: logistic regression approach 52
4.3 The problem of classifier fusion: definitions 53
4.4 Optimal Linear Combination Rules: IN and DN 56
4.4.1 IN method 57
4.4.2 DN method 64
4.5 Discussion 66

5 Experimental Evaluation 69
5.1 Introduction 69
5.2 Descriptors 71
5.3 Ridgelets descriptors 73
5.3.1 Robustness to resolution 73
5.3.2 Invariance to similarity transforms 74
5.3.3 Robustness to degradation and vectorial distortion 76
5.4 Combining Classifiers 78
5.4.1 Using synthetic data 78
5.4.2 Using Shape Databases 84
5.5 Combining ridgelets descriptors 92
5.5.1 Invariance to similarity transforms 94
5.5.2 Robustness to degradation and vectorial distortion 95
5.5.3 Discussion 97

6 Conclusions 99
6.1 Summary of the Contributions 99
6.2 Discussion and Conclusions 100
6.2.1 Ridgelets descriptors 100
6.2.2 Classifier Fusion 101
6.3 Open issues 102
6.4 Final Conclusion 103

Bibliography 105

List of Tables

2.1 Pavlidis' taxonomy. 14
2.2 Zhang and Lu's taxonomy. 16

4.1 Examples of combination rules 52
4.2 Definition of random variables for binary classifiers. 53

5.1 Recognition rates at each decomposition level (columns) for each test . 75
5.2 Recognition rates for degraded and distorted images using ridgelets and ART descriptors 77
5.3 Sets of experiment using two shape databases and two sets of descriptors 84
5.4 Confusion matrices for IN and *mean* rules. 91
5.5 Invariance to similarities 95
5.6 Test with degradation and vectorial distortions tests using the PIN method. 96

List of Figures

1.1 Examples of documents processed by document analysis systems. . . . 2
1.2 Scheme of a document analysis process for technical drawings 4
1.3 Scheme of the shape recognition process. 5
1.4 Architectures for combining features to return a single value 6

2.1 Mehtre et al.'s taxonomy. 15
2.2 Example of primitive and descriptors 18
2.3 Example of the documents processed in Lladós [1997] 19
2.4 Skeleton and Contour of a Circumference 21

3.1 Radon transform of *Electrical J* 32
3.2 Pseudopolar grid for $N = 8$ 34
3.3 Geometric distortion of pseudo-polar coordinates in rotated shapes . . 35
3.4 Fast Wavelet Transform, FWT. 37
3.5 Extension of FWT to bivariate functions. 38
3.6 Ridgelets coefficients of the graphic symbol *Electrical J*. 41
3.7 Representation of ridgelets coefficients. 42
3.8 Coefficients from F_1^3. 46

4.1 Straight lines, $A\lambda_{\mathcal{N}} + B\lambda_D = C$, crossing the set of feasible $\lambda_{\mathcal{N},D}$: $0 \leq \lambda_{\mathcal{N}}, \lambda_D \leq 1$. 62
4.2 Surface defined by function (4.17) 64

5.1 Models of GREC'03 database 70
5.2 Numerals of the MNIST database 70
5.3 Some examples of degraded images. 76
5.4 Some examples of distorted images. (a) Without distortion. (b), (c) and (d) correspond to increasing distortion degrees. 76
5.5 Validation of the classifier fusion method 82
5.6 Misclassification rates using combination rules: IN, max, median mean and DN 83
5.7 Example of distributions of symbol *Electrical J* 84
5.8 Binary Classifiers 88
5.9 Differences between DAB and $DAB-\sigma$ classifiers 89
5.10 Mean of misclassification rates for multiclass classification. 90

5.11 weights of CR and IN methods 97

List of Algorithms

3.1 Computing ridgelets coefficients using *FSS* algorithm. 39
3.2 Combination rule of ridgelets descriptors 45
4.1 IN method 58
4.2 DN method 65
5.1 Generalized version of Adaboost Schapire and Singer [1999] 86
5.2 $CNormal$ Classifier 87
5.3 Pyramidal IN method 93
5.4 Pyramidal DN method 94

Chapter 1

Introduction

In this Chapter we will give a general overview of the shape recognition process. We will put it in the context of the research done inside the Document Analysis field, giving the motivation and pointing up those aspects that we will treat in this dissertation:*shape recognition* and *classifier fusion*. We will summarize the objectives of this work as well as the contents of each chapter.

1.1 Overview of Document Analysis

In the field of Document Analysis we would like to be able to automatically process any kind of digital document —see Fig. 1.1. We mean extracting the document layout and identifying each of its parts, recognizing its contents and organizing them in order to make searches of its components, through the document itself, but also through different documents. Layout analysis is necessary to extract high level information from documents like magazines or newspapers. Thereby, we try to identify regions like headlines, figures (and their captions), charts, etc. to use them for indexing and retrieval purposes because the relevant information of such documents is usually condensed in these regions. Besides, once we have detected and correctly identified these regions we must apply expert tools to obtain relevant information. Depending on the complexity of the object to be recognized we will have to divide it into smaller entities to be processed separately. After the processing of documents, the amount of extracted information can be considerable. Thus, we can require appropriate indexing tools able to organize and deal with this heterogeneous information.

The process of Document Analysis depends on the document treated. Thereby, type-written documents were the first kind of documents treated by researchers. We can find many works from the 80s and 90s proposing Optical Character Recognition (OCR) methods for this kind of documents. Nowadays, we can purchase OCR software able to correctly identify almost all isolated characters in type-written doc-

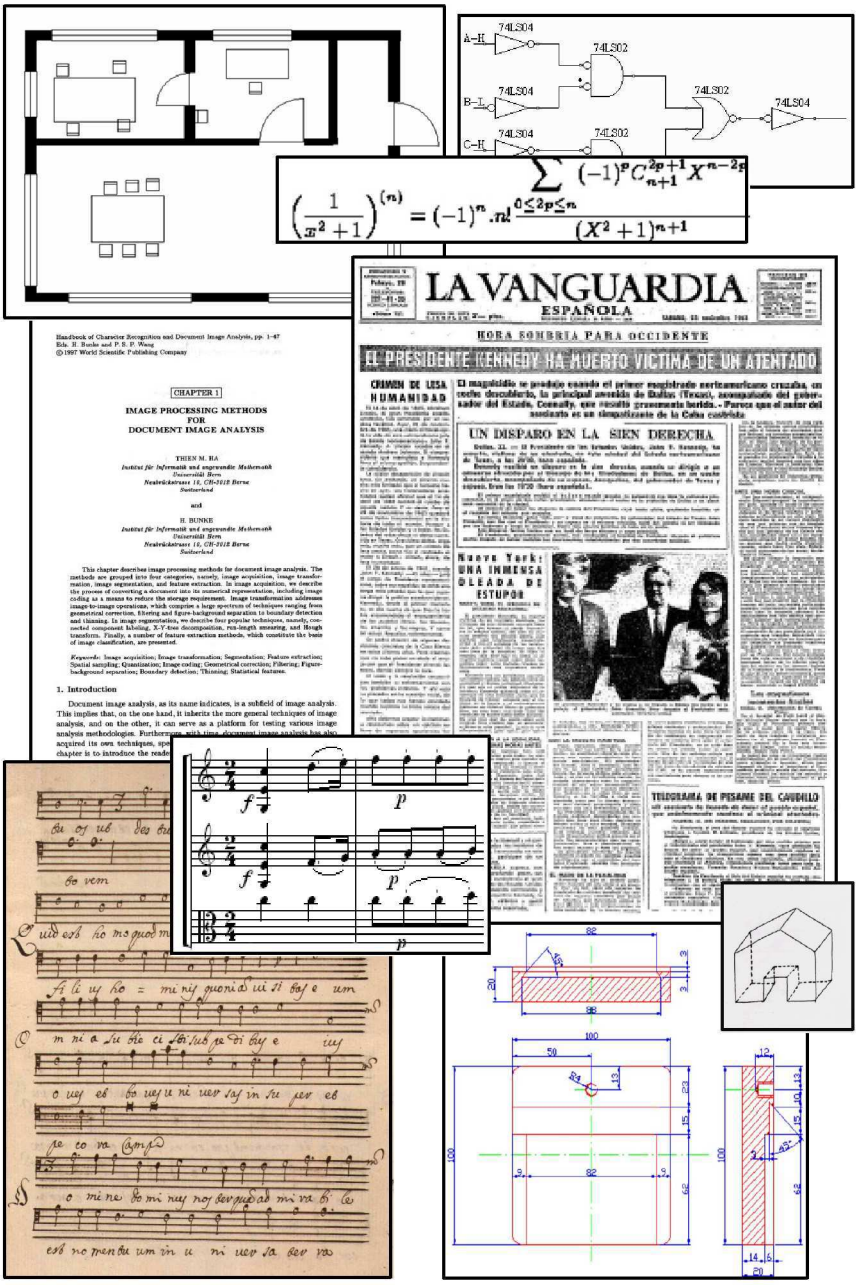


Figure 1.1: Examples of documents processed by document analysis systems.

uments. This fact has provoked research interest to move on to other challenging documents, like historical, ancient, hand-written, sketched and technical documents. We can group these last types of documents in three families, not necessarily disjoint, of documents: ancient documents, technical documents and sketched documents —we can see some examples of these types of documents in figure 1.1. Ancient, or historical, documents are digitized from paper support. They are usually degraded by the effect of time and most of them are manuscripts. Besides, notation or writing rules of some kind of documents like mathematical scripts or musical scores have evolved along the centuries, making its comprehension more difficult even for human experts. On the contrary, the main difficulty in technical documents is information organization rather than document degradation. We mean that in text documents writing is linearly organized. We read from left to right (or from right to left, depending on the culture). However, in technical documents such as architectural maps, electronic circuits or engineering designs information is organized in two dimensions, making harder the extraction of semantic information. Besides, the range of semantic entities is considerably larger than for text scripts in most of the cases. For text writing, we have symbols from an alphabet (consisting of 24 to 30 letters in Latin languages) whereas for non-Latin languages and for technical drawings the number of symbols to recognize can be bigger. Finally, sketched documents are essentially technical hand-written sketches. The research in this application domain has been motivated by the challenges of new technologies. New input devices such as tablet-PC, digital pens, PAD, etc. have motivated new uses and new needs that must combine these new technologies with the use of paper. Their main characteristic is the use of temporal information for recognition purposes. Some of their difficulties consist of shape segmentation and variability between users which make difficult the learning process. In an online framework, we can incorporate the information of the order in which the symbol has been created, making easier the task of segmentation.

Each of these challenges has motivated different lines of research in the field of Document Analysis at different levels. On the one hand, pre-processing techniques have been developed to upgrade the quality of the document image, reducing noise from the input devices and minimizing the effects of the degradation of documents. On the other hand, a deep study in shape segmentation has been carried out. In particular, in technical documents, shape segmentation is usually achieved after vectorising the image. Finally, many descriptors have been proposed for shape recognition and for shape interpretation —for a review of these interests of research Tabbone [2005] can be consulted. If we focus in the domain of technical documents, in figure 1.2, we have depicted a scheme of shape recognition process. We start by applying segmentation methods to detect the regions of interest and to extract the shape —after pre-processing the document to reduce as much as possible the noise and to increase, thus, the quality of image. Then, we input the unknown shape into the shape recognition system which compares the query shape with a shape database containing the set of *shape models*. However, this scheme is extremely simple for most of the documents to analyze and a recurrent dilemma is raised: We need to segment shapes to correctly recognize them but we also need to recognize to correctly segment shapes. In other words, in most cases we are not able to correctly segment a shape if we do

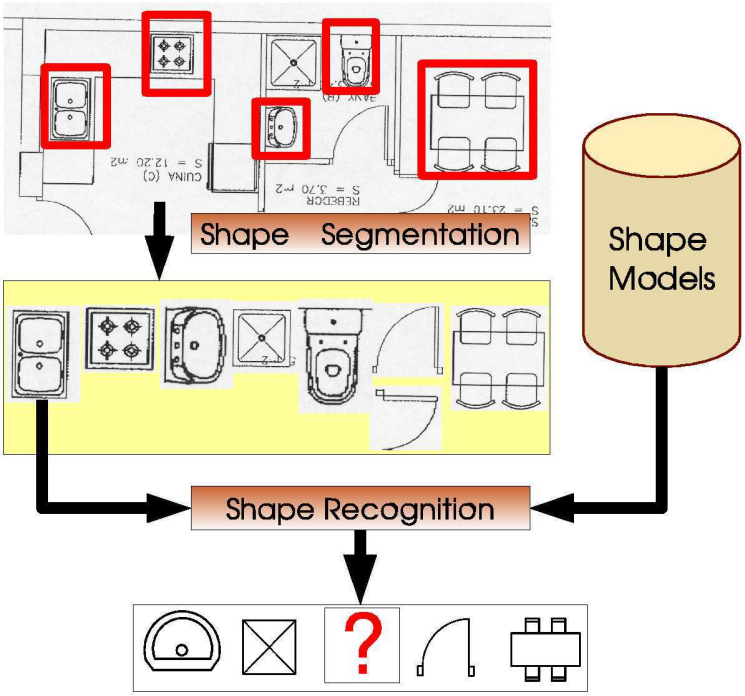


Figure 1.2: Scheme of a document analysis process for technical drawings

not know which is the shape to segment. In this sense, spotting techniques try to recognize shapes, even though the segmentation is not perfect. They try to detect regions of interest where there is a high probability to find a shape, then they supply these regions to a recognition engine which will try to recognize the shapes within the region of interests or, in case that the recognition process has failed, they will refine the shape segmentation in a later step.

Our work has been motivated by specific problems in the domain of technical documents. However, among all these problems, challenges and lines of research, we have focused on a general shape recognition framework without considering shape segmentation and preprocessing steps.

1.1.1 Shape Recognition

This dissertation is about shape recognition. On the one hand we have an unknown shape, the query, and on the other hand we have a set of candidates, the models —Fig. 1.3. The recognition process consists of identifying which is the most similar model to the query shape. In the general context of pattern recognition we can distinguish two stages: *shape description* and *shape classification*. The shape description

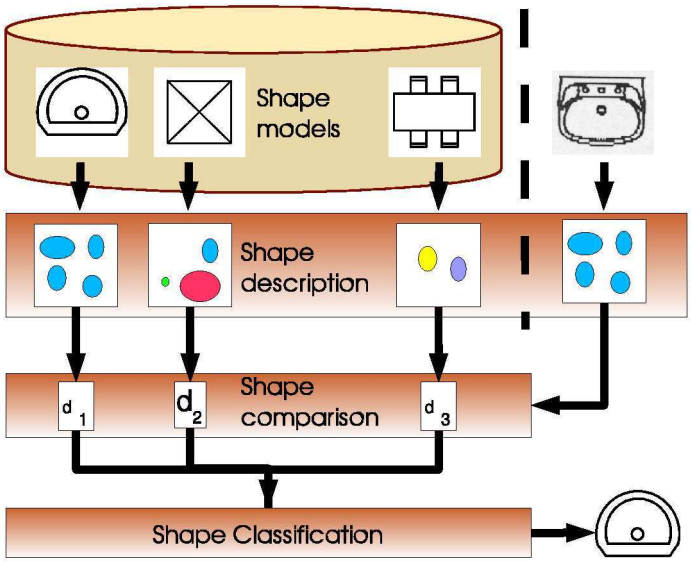
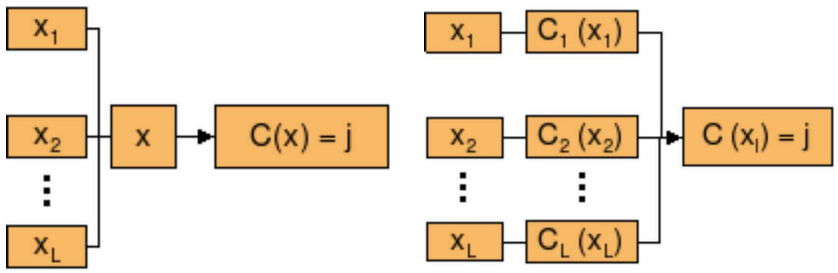


Figure 1.3: Scheme of the shape recognition process.

stage is characterized to be a process which only depends on the properties of the shape, whereas in the classification stage we usually take into account information from other shapes. In other words, in the shape description step we extract the relevant information which will be used, later, in the classification stage. In particular, these two stages can be applied to the shape recognition process. However, in shape recognition a third stage, namely *shape comparison* has to be defined. Classification is usually done through the comparison between a query shape and a set of shapes used as models. Thereby, this comparison should be included in the classification stage, although some shape recognition methods include the definition of the similarity measure in the shape description stage —as we will see in chapter 2. Besides, *structural* methods focus their interest on looking for efficient matching algorithms. That include, at the same time, comparison and classification stages. We are interested in distinguishing the shape description from the shape comparison and shape classification stages. Then, these three stages can be described as follows:

Shape description: The shape description is an abstraction process consisting of obtaining the relevant information from the shape. This is a critical stage because later, classification will depend on the goodness of the extracted features. Ideally, all shapes from the same model should have the same features.

Shape comparison: We have to define a similarity measure to compare shapes using the extracted features. Not all similarity measures can be applied to any descriptor. Depending on the properties of descriptors some measures will be more suitable than others.



(a) Feature Fusion (b) Classifier Fusion

Figure 1.4: Architectures for combining features to return a single value

Shape Classification: The classifier must be able to decide the model corresponding to the shape. If the extracted features are good enough a simple classifier based on the proximity of the shape to the models should work, typically a threshold value. However, in real problems we need more complex classifiers able to deal with the variability, noise, occlusions and similarities between models.

1.1.2 Classifier Fusion

Unfortunately, one kind of descriptor is not usually enough to achieve satisfactory results in general pattern recognition problems. Thus we must combine the information from different sources to improve the performance of single descriptors. In this sense, there are several starting points to face this problem. We can find strategies consisting of merging features from different descriptors into a single descriptor and then, training high performance classifiers like neural networks [Lippmann, 1988], boosting-based classifiers or Support Vector Machine [Burges, 1998]. However, for general purpose problems in which the number of classes begins to be high and the shapes to be recognized can be counted by thousands, these expert classifiers begin to fail. Moreover, when the classes to recognize may change depending on the problem handled or according to user requirements, neural networks, boosting algorithms and support vector machine become rigid methods because they should be trained again, which is usually time-expensive. Therefore, we have to find strategies that easily permit us to adapt to the user needs, avoiding to train again the system each time the problem conditions change. One possibility, the one we have followed in this work, is to use different classifiers, one for each descriptor, and combine them in such a way that the global performance increases —see figure 1.4 for an example of this strategy compared to the combination of features, explained before. This strategy is known in the community under some different names: *aggregation operators*, *combination rules* or *classifier fusion*, depending on the context and the characteristics of classifiers. We like the name of *classifier fusion* and we will use it in most of the times, despite sometimes we will also use indifferently the terms of aggregation operator and combination rule to refer to the same concept.

Then, this dissertation is also about classifier fusion. Classifier fusion methods started to be developed in the 90s. A raised question was whether it is reasonable to combine the output of several classifiers to improve global classification rates and, if plausible, which is the best way to combine these outputs. The first question has affirmatively answered through the research done in machine learning and experimental results obtained during these last years [Kittler, 2000, Kittler et al., 1998, Schapire, 1990, Tax et al., 2000]. We find that fusion methods improve individual classifiers performance. However, the question concerning which of fusion method is the best one among all of them remains open. Besides, experimental evidence leads to think that there is not a "best" fusion method, that it will depend on the application. In this sense, for example, a discussion on when it is better to average or to multiply the classifiers is given in [Tax et al., 2000] under strict probabilistic conditions but the final behavior will depend on the application. Works like those by Stejic et al. [2005] propose a genetic algorithm that learns which aggregation operator should be used every time. In general, classifier fusion methods have been grouped in two classes according to the architecture used to combine them: *serial* and *parallel*. Of course, there is a third architecture, an *hybrid* architecture, which combines serial and parallel classifier configurations [Jain et al., 2000, Zouri, 2004].

In Serial combination, classifiers are organized into ordered stages. Each stage is composed of a single classifier that takes into account the output of the precedent classifier to confirm, or reject, its response. Each classifier filters some candidates to be the model reducing in such way recognition errors at each stage.

In the Parallel architecture, on the contrary, we evaluate all classifiers at the same time and, afterwards, we combine classifier outputs to return a single candidate (or a ranked list of candidates). In these systems, the choice of the aggregation operator is critical.

Hybrid architecture combines both parallel and serial configurations to take advantage of the goodness of each one. The difficulty relies on the design of such an architecture.

We have chosen a parallel architecture to fusion classifiers because we are interested in the aggregation operators used to combine them. In particular, the chapter 4 has been devoted to find linear combination rules minimizing the misclassification rates.

1.2 Objectives and contribution of the present dissertation

The original proposal of the research project was related to the search of graphic entities in technical documents using content-based information. Thus, we had to find suitable descriptors for this kind of shapes, which could be rotated and scaled in any

part of the document. Besides, the number of symbols could be big so that descriptors should find an agreement between the size of the descriptor and its description capabilities to facilitate later indexing tasks.

The global problem required the help of a wide range of methods from the fields of shape recognition, segmentation, vectorization, classifiers and indexing. We had to focus our work on some of these parts and we decided to intensify our research in the shape recognition field to find out suitable descriptors for graphic symbols. However, the lack of a general theory of shape description required a review and a test of existing techniques in feature extraction methods. In addition, existing descriptors do not seem convenient when the number of symbols in the database to recognize increase dramatically and we have explored new descriptors like multiresolution descriptors based on ridgelets transform. Multiresolution descriptors offer a hierarchical description of shapes according to their resolution. It seemed that this kind of descriptors could be useful to define an indexing structure, which facilitates the task of recognition when the number of symbols increase. Furthermore, the capacity of ridgelets transform to detect linear singularities made this transform more suitable than other multiresolution transforms. Nevertheless, this descriptor took with him some intrinsic problems, namely the size of descriptor and the representation of symbols at different levels of resolution. On the one hand, we have studied different strategies to reduce the size of descriptors. On the other hand, it was not evident how to combine all resolutions of ridgelets descriptors and therefore, we have decided to explore classifier fusion. We have studied linear rules as a way to combine all resolutions and afterwards, we have compared them with other popular combination rules like *max* and *median* in a framework for the general problem of combination of classifiers. Furthermore, in this framework, we had to do some assumptions about classifiers, which motivated a slight study of classifiers and their behavior

As a result, this dissertation treats about two different issues in shape recognition: shape representation using multiresolution descriptors and the fusion of classifiers to combine the information of multiple descriptors to improve single performance and increase recognition rates.

1.2.1 Goals of the work

In this section, we will pinpoint the main goals achieved in this dissertation. These contributions turn around two axes: *shape descriptors* and *classifier fusion*.

Shape descriptors When we have reviewed the bibliography, we have realized that there is not an unified terminology. Due to the complexity of the shape recognition process, the division into three stages done above, namely *shape description*, *shape comparison* and *shape classification* is not standard and depends on each author. Thereby, there are several terms to denote similar concepts but being slightly different. Some feature extraction methods consider the similarity measure as a part of the method, meanwhile others not. Hence, our review of the state of art starts by the proposal of a set of definitions consistent with

the decomposition of the shape recognition process in these three stages. More specifically:

- We have formalized and unified the concept of feature vectors, signatures and other terms used to refer to the results of feature extraction methods under the name of *descriptor*.
- We have generalized the notion of *primitive*, specific for *structural* descriptors, proposing a definition that can be applied to any descriptor.
- We have introduced the notion of *primary* feature extraction method in order to decompose a descriptor in more elemental entities. Such decomposition should not be seen like the decomposition of natural numbers in prime numbers, in the sense that a descriptor can be decomposed in a unique sequence of *primary* feature extraction methods. On the contrary, it must be seen as a flexible manner to decompose a descriptor identifying common elements among them for their classification.

After, we have reviewed existing methods and organized this part of the chapter according to these definitions. In particular we have defined a taxonomy based on the characteristics of *primitives*, on the properties of *feature extraction* methods and on the characteristics of *descriptors*.

Ridgelets descriptors are particular multiresolution descriptors. We have considered ridgelets transform because this transform is suitable to detect linear singularities in multivariate functions. Thereby, in the context of Graphic Recognition, this kind of singularities correspond to lines. Concerning other graphic entities, like arcs, theoretical results assert that ridgelets transform is not worse than other multiresolution transforms like wavelets [Candès, 1998]. The study of ridgelets transform has involved:

- The study of Radon transform. Ridgelets transform requires the use of Radon transform. Thus we have implemented Radon transform seeking to reduce the time complexity of this transform. A modified version of the Fast Slant Stack algorithm [Averbuch et al., 2001] that corrects some geometrical distortions when the shape is rotated has been proposed.
- Definition of a ridgelets-based descriptor. We have constructed a shape descriptor based on the ridgelets coefficients. In this sense, we have modified the ridgelets representation to make descriptors robust to noise and vectorial distortion. Besides, we have normalized it to make it invariant to shape shifts and scale.
- Multiresolution descriptors have different number of scales depending on the shape resolution. In order to compare shapes at different resolutions we have introduced the notion of *Decomposition Level*, *DL*, grouping in the same *DL* all scales that appear when we increase the resolution of the shape.
- Ridgelets descriptors are not invariant to symbol rotation. So, we have proposed a similarity measure achieving invariance for each scale.

- We have explored different techniques to reduce the size of ridgelets descriptors. Some proposals consist of applying $\mathcal{R}$-Signature descriptor [Tabbone et al., 2006, 2001] to each ridgelet scale and defining Local Norm descriptors [Costa and Cesar Jr., 2001] based on ridgelets transform

Classifier Fusion: Although the original problem was raised in the context of combination of multiresolution descriptors, we have attacked it in the general framework the combination of classifiers. Hence, the work in this area has involved:

- Raising the problem of combining ridgelets descriptors as a general problem of classifiers fusion. In this context, we have proposed a probabilistic framework where we have modeled two-class classifiers as random variables.
- Focusing on linear combination rules, we found the optimal weight for each classifier that minimizes the classification error under some constraints about classifier distribution and dependence/independence of classifiers. For independent classifiers we have found an explicit formula for the cases where classifier distribution can be approximated by Normal and Dirac distributions. For dependent classifiers, we have raised an optimization problem to be solved for classifiers whose linear combination follows a a normal distribution.
- Extending the optimal solution for binary classifiers to multi-class classifiers through the operator $\arg\max$.

Combination of ridgelets descriptors: We have finished our dissertation applying the scheme of classifier fusion to the ridgelets descriptor. We have constructed two parallel classifiers. The first classifier uses the weights obtained under the assumption of *Independence* of classifiers whereas in the second one the weights have been obtained solving the optimization problem corresponding to the *Dependence* assumption.

1.2.2 Framework of the thesis

We had to make some practical decisions in order to pass from the abstract shape recognition problem, raised at the beginning of this chapter, to a practical shape recognition problem. In this sense, some of the decisions taken were related to the following issues: the application domain, the theoretical framework and the experimental framework.

Application Domain: We have mainly worked with graphic symbols, specially when we have studied shape descriptors, because the origin of this work was to look for suitable shape descriptors for Graphic symbols. However, in the context of classifier fusion we have extended our domain to datasets of handwritten numerals.

Theoretical framewok: Our mathematical background has influenced the theoretical approach. Multiresolution descriptors are based on wavelets theory which

is an active interest of research in functional analysis community. In this context, the shape image is taken as a bivariate function and all the mathematical machinery of this field has been applied. Furthermore, a probabilistic approach has been adopted in the classifier fusion approach, and as for multiresolution analysis we have applied all the probabilistic terminology we have needed.

Experimental framework: Two slightly different experimental frameworks have been developed according to the proposals for ridgelets descriptors and classifier fusion. Ridgelets descriptors are applied to segmented images without any need of symbol vectorization. For this reason, we have compared ridgelets descriptors with other descriptors that do not need to be vectorized and have to be applied on segmented shapes, too. The main measure to evaluate their performance is the recognition rate or, alternatively, the classification error. Concerning classifier fusion we have continued working with segmented, and non vectorized, shapes. Furthermore, recognition rate has been the measure used to evaluate classifier performance. However, we have compared our proposal of classifier fusion with other classifier fusion approaches. For that reason, several classifiers have been trained. Finally, we have complemented the set of experiments using synthetic data to simulate classifiers satisfying the theoretical assumptions done in the theoretical framework and other classifiers not verifying them.

1.3 Organization

This dissertation has been organized similarly to this chapter, which more or less follows the temporal evolution of the work with two exceptions: The review of shape descriptors, that has been done during all the time but have been summarized in the next chapter, and the experimental evaluation that has been grouped at the end of this work, before the conclusions chapter. More specifically, in each chapter we will find the following;

- In chapter 2, we review the different shape descriptors proposed in the literature. First, we introduce some definitions related to descriptors and feature extraction methods, permitting us to fix the vocabulary that we will use along this work. Later, we introduce the most used shape descriptors in the graphics recognition domain, grouping them according to the properties of the descriptors. We have centered our attention in descriptor properties being the case that we have introduced several times the same descriptor depending on the property discussed at each moment. This discussion has been oriented to introduce, in the next chapter, a ridgelets descriptor.

- Chapter 3 is devoted to ridgelets descriptors. In the first part we introduce the mathematical framework that we need to explain the ridgelets descriptor, *i.e.* Radon transform and Multiresolution Analysis theory. Then, we continue with the implementation of ridgelets transform and the definition of a descriptor based on this transform. We finalize this chapter by enumerating some of the

problems of ridgelets descriptors, namely the size of the ridgelets descriptor and the combination of descriptors from all resolutions. In this chapter, we also propose some solutions for reducing the size of ridgelets descriptors, through the definition of *Local Norm* descriptors.

- In chapter 4, we face the classifier fusion problem. We start with a short review of classifiers and classifier fusion techniques and then we develop a probabilistic framework to propose optimal solutions for linear combination of two-class classifiers.

- The experimental evaluation of ridgelets descriptors and the optimal linear combination rule is done in chapter 5. In this chapter we evaluate ridgelets descriptors using symbol databases used in graphics recognition community to compare ridgelets descriptors to other existing shape descriptors. The evaluation of the combination rule is done in a similar way, using existing symbol databases, but also using different classifiers to determine its influence in the combination rules. We also carry out some experiments using synthetic data, which has permitted us to experimentally validate the theoretical framework developed in the previous chapter. We finalize this chapter by applying the classifier fusion methods introduced in chapter 4 to the ridgelets descriptors introduced in chapter 3.

- We finalize our work by summarizing the main contributions and the obtained results. Besides, we point out some future lines of research in shape recognition and classifier fusion.

Chapter 2

Shape Descriptors

In this chapter we will review some of shape descriptors proposed in the last years. We will propose the definitions of *descriptor* and *primitive* and we will introduce the notion of *primary* feature extraction method. With these definitions, we will classify shape descriptors according to their properties pointing out their strengths and weaknesses, concluding with a discussion about the proposed taxonomy for graphic descriptors.

2.1 Introduction

In the previous chapter we have divided the shape recognition process into three stages, namely shape description, shape comparison and shape classification. In the shape description stage, we describe the shape by extracting relevant features. In the comparison stage, we compare shapes using these descriptors. Finally, in the classification stage, we decide the class where the shape belongs to. However, if we try to identify each of these stages in the existing methods we will realize that such correspondence can not be done in a straightforward way. Most of the difficulties to do this correspondence relies in the use of different terminologies to denote the different elements taking part in the recognition process.

Since the first studies at the end of the sixties and beginning of the seventies, we can periodically find in the literature several survey works which try to summarize the new advances in shape descriptors, briefly explaining some of their main properties [Loncaric, 1998, Mehtre et al., 1997, Pavlidis, 1978, Rui and Huang, 1999, Smeulders et al., 2000, Trier et al., 1996, Zhang and Lu, 2004]. We will briefly summarize the surveys of Pavlidis [1978], Mehtre et al. [1997] and Zhang and Lu [2004] in order to illustrate the historical evolution of the proposed taxonomies as well as the terminology used and its meaning in the shape recognition process.

One possible taxonomy can be that introduced by Pavlidis [1978] who divide *algo-*

	Internal	External
Scalar	2D Fourier Transform Moments	Fourier Transform
Space	Medial Axis Transform (*skeleton*) Primary convex subsets	Chain code syntactic descriptors tree grammars

Table 2.1: Pavlidis' taxonomy.

rithms for shape analysis in several binary classes: *external* and *internal* algorithms; *scalar* and *domain* transforms; and *information preserving* and *information nonpreserving* methods. *External* methods refer to methods defined over the local boundary, whereas *internal* methods are defined over the whole shape. He also makes another distinction between *scalar* and *domain* transforms. This last kind of methods transforms one image to another image whereas scalar methods compute scalar features from input images. Finally, Pavlidis talks about *information preserving* and *information nonpreserving* methods depending on whether it is possible to reconstruct the original image from the shape descriptor. According to these three criteria, he defines the following four classes of algorithms: *external scalar* transforms, *internal scalar* transforms, *external space domain* techniques, *internal space domain* techniques — table 2.1.

If we advance through the history of shape descriptors, we can observe some changes. A first remark is that there were important advances in shape descriptors from the end of the seventies to the middle of the nineties. The number of shape descriptors increased and the number of groups of descriptors, too. Mehtre et al. [1997] classified shape descriptors as *boundary based* methods and *region based* methods —which correspond to external and internal methods in Pavlidis' taxonomy. The tree structure used to organize the groups of shape descriptors illustrates the difficulty of defining a suitable taxonomy of shape descriptors —Fig. 2.1. As example, in boundary-based methods we find descriptors like chain code, area, perimeter, compactness and Fourier descriptors, like in Pavlidis' taxonomy. However, we can remark that some of these descriptors like area, elongation and compactness, are also classified like *complete*, *geometric*, *spatial* and *region* shape descriptors. Therefore, the classification is not unique. Besides, we must emphasize the different terminologies used. Pavlidis talks of *algorithms for shape analysis*, whereas Mehtre et al. talk of *shape measures*, *shape recognition techniques* and *shape description*.

Later, Zhang and Lu [2004] defined a slightly different taxonomy of that proposed by Methre. They still differentiate between *contour* and *region* based descriptors but they simplify Methre et al.'s classification by only differentiating between *structural* and *global* descriptors —in other contexts global descriptors are also known as *statistical* descriptors. If we observe the descriptors reviewed by Zhang and Lu we can remark that they are essentially the same descriptors than in Mehtre et al.'s review. It means that advances in shape description have been done by improving the existing methods. Besides, if we analyze the methods reviewed by Zhang and Lu we can remark that some of them refer to shape comparison: *Hausdoff distance* or

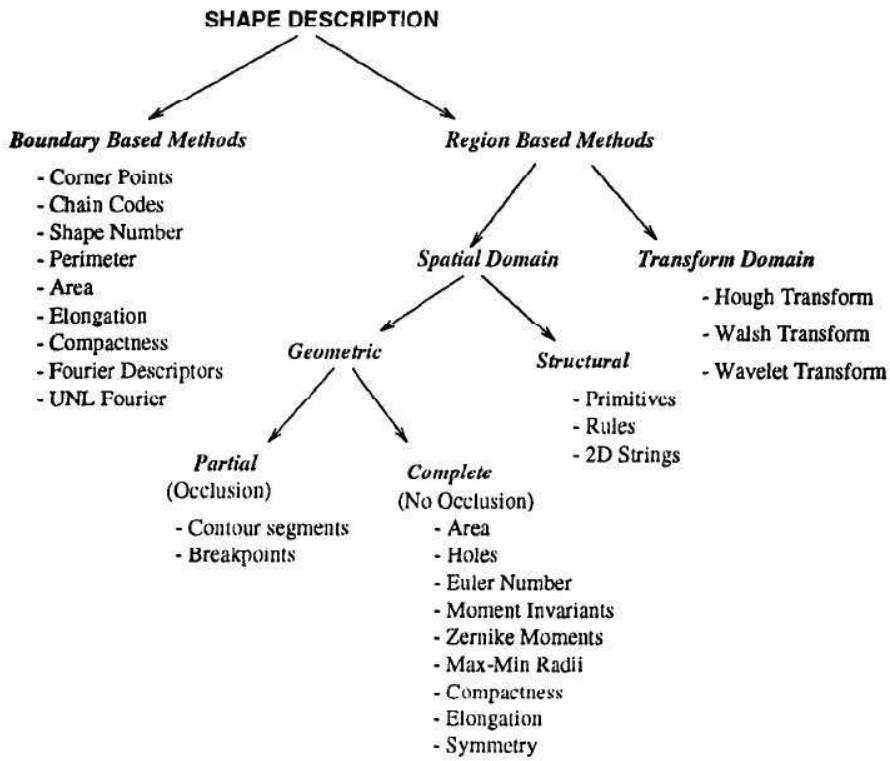


Figure 2.1: Mehtre et al.'s taxonomy.

elastic matching, which mean that they include the shape comparison into the shape description stage.

Therefore, there is not a single, common and widely accepted term to define the result of the description of shapes since the first applications of shape representation in the seventies. Moreover, depending on the author's idea and background, different terms to denote the same concept have been used until now. We can talk of *feature vectors*, *descriptors* or *signatures* depending on the application, mathematical properties or data representation. The problem is not only terminological. All these terms have slightly different meanings and/or properties. We can find some examples of that. For instance, Zhang and Lu [2004] define *Shape signature* as "a representation by a one dimensional function derived from shape boundary points". Meanwhile in other contexts a *signature* is a n-dimensional vector of complex or real values [Tabbone et al., 2001]. Both definitions are quite different from a theoretical point of view. Besides, the concept of *feature vector* implicitly implies a n-dimensional vector where each component is a feature and they are often used in statistical approaches. Usually, *primitives* are segments, arcs of circumference or splines represented by their parameters, and they are used in *structural* or *syntactic* approaches.

	Contour	Region
Structural	Chain code Polygon B-Spline Invariants	Convex Hull Media Axis Core
Global	Perimeter Compactness Eccentricity Shape signature Hausdorff Distance Fourier Descriptors Wavelet Descriptors Scale Space Autoregressive Elastic Matching	Area Euler Number Eccentricity Geometric Moments Zernike Moments Pseudo-Zernike Moments Legendre Moments Generic Fourier Descriptors Grid Method Shape Matrix

Table 2.2: Zhang and Lu's taxonomy.

But the different terminology is not the only difficulty to study the different existing feature extraction methods. As the description of the shape is usually the basis of shape classification, one question arises: Where does shape representation finish and where does shape comparison and shape classification start?

Depending on the terminology and the underlying definition used, we can fix the border between these stages in one point or another. We have observed the difficulty to fix this border in the reviewed literature. Zhang and Lu [2004] distinguish between *shape representation* and *description techniques*, but differences between both concepts remain unclear. It seems that a *description technique* involves not only the features used to represent shapes but also the similarity measure or classification methods to match them. Meanwhile *shape representation* only refers to the features. Mehtre et al. [1997] use the concept of *shape measures* to refer to descriptors and similarity measures to compare images in a Content-Based Image Retrieval (CBIR) system. Some of the descriptors reviewed in their work are invariant moments, Zernike moments or Fourier descriptors, which have been considered as *shape representations* by Zhang and Lu [2004]. On the other hand, Loncaric [1998] and Trier et al. [1996], in their respective surveys of shape analysis techniques and feature extraction methods for character recognition, essentially review shape descriptors without taking into account the measures used to match them or the classifiers involved in the recognition process. Except for the so-called *syntactic* approaches, which are methods based on grammars. The problem lies in the fact that a *syntactic* approach provides not only a *shape representation* but also the definition of a measure —the parser— involved in the recognition process. It is obvious that each grammar requires its specific parser, but the *descriptor* is the definition of the grammar and the matching parser is the similarity measure.

Then, we have considered that we should unify the terminology used to explain shape descriptors. In this sense in section 2.2 we have proposed the definitions of con-

cepts like *descriptor*, *feature extraction method* and *primitive* to clarify the different parts taking part in the recognition process. In addition, because shape description is a complex task, we have introduced the concept of *primary feature extraction method*, which will help us to decompose *feature extraction methods* into more "elemental" entities in section 2.3.

2.2 Definitions

All these differences in the ways to review shape description may provoke confusion to a newcomer who is faced to several terms which are very similar in meaning and whose differences are often linked to computational representation, mathematical context or historical reasons. Therefore, we have proposed the definition of these concepts: *descriptor*, *primitive* and *feature extraction method* that we will use through all this work. The definition of *descriptor* will include the terms of *feature vector* and *signature*, whereas the definition of *primitive* will generalize the usual notion in structural approaches.

Definition 1 (Descriptor). *Is a set of features, extracted from* primitives*, used for pattern recognition tasks.*

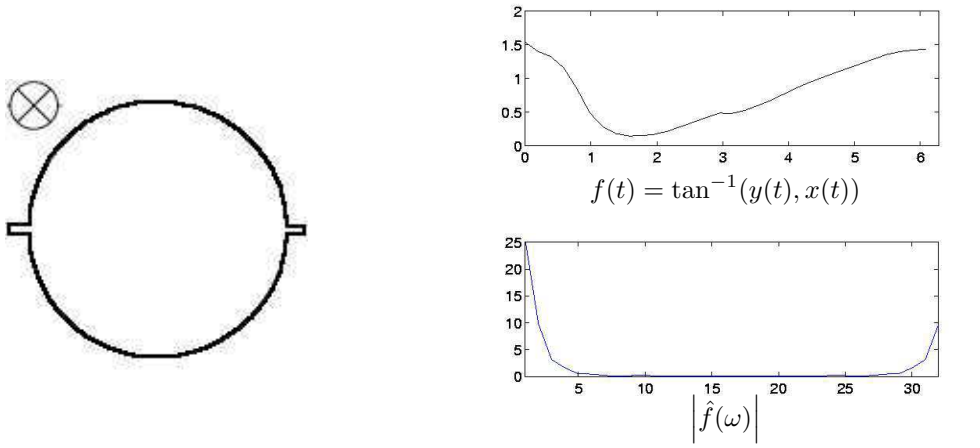
This definition is coherent with the division of shape recognition in shape description, shape comparison and shape classification. As a result of the description of a shape, we obtain a descriptor which later is used in the shape comparison stage. The matching process does not characterize the shape and hence, it does not appear in the definition of descriptor. Besides, we have to remark that we do not find any reference to feature representation. This fact, will permit us to state that *feature vectors*, *grammars* and *signatures* are descriptors with independence of the structure used to represent features. Finally, to completely understand the definition of descriptor we have to explain what a *primitive* is. In this sense, we have proposed a definition of *feature extraction method* that will permit us to introduce the concepts of *primitive* and *descriptor* in a more natural way:

Definition 2 (Feature Extraction Method). *A feature extraction method (FEM) is a map: $D : X \rightarrow Y$ such that:*

- *The elements $x \in X$ are* primitives
- *For any $A \subset X$, $y_A = D(A)$ is a* descriptor.

This mathematical definition of feature extraction method states primitives and descriptors as input and output of the method, respectively. Let us illustrate these concepts with the help of an example. Let us suppose that we have the shape shown in figure 2.2. Then, several descriptors can be defined:

- We can extract the shape contours and consider the external contour as a primitive. The closed curve is a periodic function and hence, the Fourier transform



(a) Shape contour (b) Fourier descriptor based on the contour

Figure 2.2: Example of primitive and descriptors. (a) Example of a primitive (b) Example of a descriptor, the Fourier transform (below) of the $\tan^{-1}$ (above)

can be applied to it, being the Fourier coefficients the descriptors —as depicted in figure 2.2.

- We can vectorize the image and define an adjacency graph. In this case, the primitives will be the vectors and the descriptor, the graph.
- We can compute Zernike moments, being the image the primitive and the coefficients of Zernike functions, the descriptor.

These last examples show that we can find a wide heterogeneous pool of elements that could be called *primitives*. Not only vectors and circumference arcs, as it has traditionally been considered in structural approaches, but also pixel curves, raster images, polylines, splines, etc. Besides, the definition of *descriptor* ties with one of the reasons for being of FEMs, *i.e.* we need descriptors for something: recognition, retrieval, spotting, indexing, etc. In this context, if we have a set of elements that we can use for some pattern recognition purpose, then we probably have a descriptor. On the other hand, if the features extracted from our data are not useful for the problem that we have to solve, then we may seriously consider if we actually have something that could be called a descriptor.

2.3 Taxonomy of Descriptors

In the reviewed surveys, the properties of descriptors are the basis of the groups of descriptors. In a similar way, we can group classifiers according to the properties of

the elements involved in the definitions of the previous section. Some examples of these properties are:

Primitive: According to the geometry of the primitives of FEM, we can distinguish between 1D descriptors and 2D descriptors. 1D descriptors will essentially correspond to contour-based and skeleton-based descriptors while 2D descriptor will correspond to region-based descriptors.

FEM: As Pavlidis, we will differentiate between FEM that permit us to reconstruct primitives from descriptors —*information preserving*— and that do not permit us this reconstruction —*information nonpreserving*. Besides, we can also group descriptors according to whether they are *invariant*, or not, to affine transforms.

Descriptors: Some properties linked to a descriptor are related to its structure. In this sense we can distinguish some particular kind of descriptors, such as *multiresolution* and *structural* descriptors.

Nevertheless, we can easily remark that classifying shape descriptors using either Zhang and Lu's groups, or the ones defined according to these new properties, is not straightforward. We can easily find methods that can be included in several categories. For instance, a graph descriptor based on Hough/Radon transform [Lladós, 1997]. This descriptor is used to detect walls in architectural hand-drawing documents — Fig. 2.3. We can identify these graphical entities because they are composed of two parallel lines joined by obliques parallel lines. On the one hand, this descriptor can be considered as structural —because the resulting descriptor is represented into a string structure— but Hough transform is a *global* and *region-based* transform, following Mehtre et al. [1997]'s taxonomy —Fig. 2.1. Using the new terminology we can describe this last descriptor as: *1D, information preserving and structural descriptor*. We can define it as 1D because Lladós apply the Hough transform to a vectorized image. Then, we have considered it information preserving because we consider that Hough and Radon transform are essentially the same transform [Deans, 1982] —and the Radon transform is *information preserving*. Finally, as a structural descriptor because the descriptor is represented using a string structure.

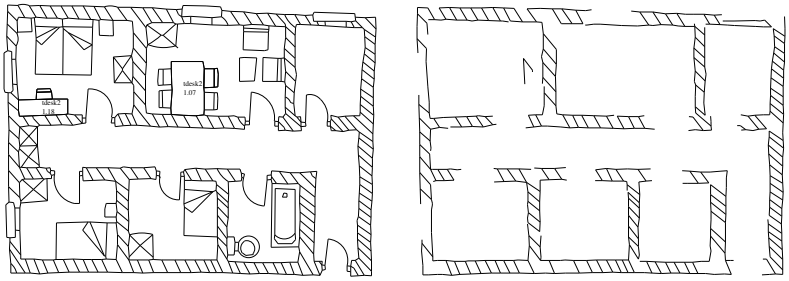


Figure 2.3: Example of the documents processed in Lladós [1997]

We have described this descriptor in this way because we have considered it as the composition of two FEM. Firstly, we have applied the Hough transform to the

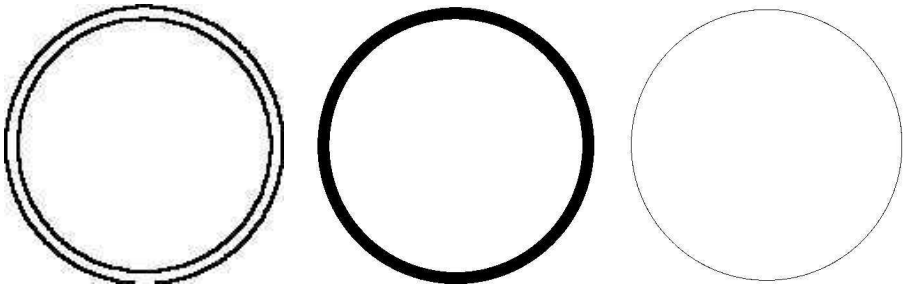
vectorized image. Secondly, we have defined a graph structure based on the Hough representation. The properties of the descriptor comes from the properties of these two FEM. In this context, we can define *primary* FEM as a a FEM used as a part of complex shape descriptors. Thereby, each of these two FEM can be considered as *primary* FEM. However, we must emphasize that the decomposition of FEM into *primary* elements have nothing to do with the decomposition of natural numbers into "prime" numbers. In the case of descriptors, we can decompose FEM in several ways depending on each situation. For us, the term of *primary* FEM is a conceptual tool used to analyze shape descriptors.

The remainder of the chapter is devoted to the review of shape descriptors paying special attention to those close to the multiresolution descriptor based on the ridgelets transform, that will be introduced in the next chapter. In this sense, we have structured the review following a scheme similar to that done in the reviewed surveys, but considering the properties of primitives, FEMs and descriptors. First, we group descriptors according to whether their primitives are 1D or 2D. Then, we have classified them depending on the FEM used to compute the descriptors. For 1D descriptors we have defined four groups: *Fourier*, *Stochastic*, *Curvature* and *Geometric Invariant*. Meanwhile for 2D descriptors we have defined three groups: *polar*, *moments* and *local norm*. For all groups, we have discussed about their invariance to affine transforms. After this review according to primitive properties we will review shape descriptors from the perspective of descriptor properties. In this sense, we have distinguished descriptors whose FEM is a multiresolution transform and descriptors that take into account the structure of the shape to represent features.

2.3.1 Primitives of Descriptors

A constant issue in the surveys from Pavlidis' to our current days is the distinction between *countour-based* and *region-based* descriptors. We have also followed this distinction but changing the names to 1D and 2D descriptors, respectively. The basis for changing the names is the definition of primitive. These names are linked to the geometric properties of primitives in a more general way than *contour-based* and *region-based* terms are. We will justify it with an example. There are feature extraction methods that can be applied to shape contours but also to other primitives as the *skeleton*. Concerning the skeleton, if we look at the skeleton as a primitive, which are the differences between a closed curve from a shape contour and a closed curve from the skeleton ? —Fig. 2.4. If we consider the external contour of an image, and then we consider the skeleton of the same shape, can anyone remark any significant difference in these two representations? We believe that both representations, the skeleton and the external contour of the circumference, are essentially equal and hence, we group descriptors computed using both of them as 1D descriptors.

According to Definition 1, we will say that a descriptor is *1D*, if its primitives can be represented as a curve. We will consider that primitives are samples of an unknown underlying curve that can be mathematically expressed as a combination of continuous functions. Most of feature extraction methods based on mathematical



(a) Contour (b) Circumference (c) Skeleton

Figure 2.4: Skeleton and Contour of a Circumference

transforms make similar assumptions, some of them imposing more restrictive conditions like derivability. The cases where these hypothesis do not hold are object of a deep study for each kind of descriptor and they are out of the scope of the current discussion. Hence, we will basically think about descriptors obtained after computing the contours, the skeleton or a vectorial description of the shape. Conversely, we have defined 2D descriptors as those descriptors whose primitives are essentially two dimensional regions.

1D descriptors

There are several transforms that can be applied to 1D primitives which can be expressed in two ways, as univariate or bivariate functions. Some examples of descriptors based on curves described by univariate functions are Fourier descriptors of Zahn and Roskies [1972], curvature-based descriptors like those of Asada and Brady [1984] and Berreti et al. [2000] or autoregressive (AR) descriptors of Das et al. [1990]. Examples of descriptors represented by bivariate functions are Fourier descriptors of Bartolini et al. [2005], wavelets [Khalil and Bayoumi, 2000] and AR descriptors of Sekita et al. [1992]. We will continue by sketching some descriptors according to the type of transform used: *Fourier*, *Stochastic*, *Curvature* and *Geometric Invariant*.

- **Fourier** descriptors are probably the most applied descriptors in shape recognition problems. Fourier transform is an *information preserving* transform. Only when the modulus is computed, information from primitives is lost. Without the phase information we can not recover the original curve. There are several ways to apply the Fourier transform to a planar curve: Zahn and Roskies [1972] propose a descriptor based on the Fourier transform of the cumulative angular function of the shape contour. Tabbone et al. [2006] apply 1D Fourier transform to the R-Signature (which is obtained from the Radon transform of the image) to reach invariance to rotation and Bartolini et al. [2005] apply this transform to the shape contour and use phase information to compute similarity between shapes. A strength of Fourier descriptors is that they permit to get a global description of curves without requiring a large number of coefficients.

However, they lose discriminant capability when the similarity between shapes is important because slight differences are confused with noise.

- **Stochastic** descriptors are *information preserving* descriptors based on autoregressive methods. These methods consist of computing the parameters of the curve using regression techniques like the least squares method. They are defined over closed curves and hence, they are usually applied to contour curves. The coefficients that fit contour curves are then used to derive invariant descriptors to similarity transforms. Das et al. [1990] and Sekita et al. [1992], propose bivariate AR models, instead of AR models based on univariate function representing the shape boundary, to overcome some shape representation problems [Kashyap and Chellappa, 1981]. With a bivariate function convex and non-convex shapes are treated in the same way. One of the drawbacks of stochastic methods is that the number of coefficients needed to describe the shape must be high for complex shapes and must be chosen empirically.

- **Curvature** descriptors are essentially computed from the curvature function, which is based on the second derivative of a planar curve. It means that the curvature function determines the planar curve, except in its position and orientation, *i.e.* it is an *information preserving* FEM. Changes of curvature in shapes are considered to be dominant features and they have been object of deep study since the beginning of the eighties. Asada and Brady [1984] introduced the concept of *curvature primal sketch* that is a multiresolution description of shape based on curvature. Besides, Berreti et al. [2000], compute the curvature function and extract local maxima points to construct a descriptor based on minimal and maximal curvature points. Finally, the CSS descriptor, which is based on a multiresolution description of the curvature function, is introduced in the MPEG-7 standard [Manjunath et al., 2002], because it is shown that this kind of descriptors usually have good performance for general shape description purpose. However the computation of the second derivative makes this kind of descriptor sensitive to noise.

- **Geometric Invariant** descriptors are inspired by Mundy and Zisserman [1992]'s geometric invariant theory which aims at constructing affine invariant descriptors. They suggest different ways to construct invariants that have been applied to construct descriptors based on wavelets transform, Tieng and Boles [1995, 1997], Khalil and Bayoumi [2002] and more recently, El Rube et al. [2006]. For more details, Forsyth et al. [1991] and Khalil and Bayoumi [2001] can be consulted.

2D descriptors

As we have said 2D descriptors are those whose primitives are essentially regions. Classical descriptors in this group are: area, perimeter, eccentricity which were first used in the seventies [Zhang and Lu, 2004]. But there are also more descriptors than these that can be grouped according to their properties. Some examples are

F-signature [Tabbone et al., 2001], who try to compute shape "internal forces"; zoning descriptors and $\mathcal{R}$-signature that could be considered to be special cases of a more generic descriptors namely *Local Norm* descriptors, which have been introduced by Costa and Cesar Jr. [2001]; Moments descriptors like invariant geometric moments [Hu, 1962], Legendre or Zernike [Teh and Chin, 1988, Yap and Paramesran, 2005]; multiresolution descriptors like wavelets or ridgelets; Fourier descriptors, Radon descriptors or ART from MPEG-7 standard [Kim et al., 1999, Manjunath et al., 2002]. We have proposed to classify all these descriptors into three different groups: *polar*, *moments* and *local norm* descriptors.

- **Polar descriptors**. Document images are usually expressed in Cartesian coordinates. However some descriptors are based on a polar representation of images, although the change of artesian-to-polar coordinates is a time-consuming process. These descriptors have proved to perform well in shape recognition tasks but they have important drawbacks. One of them is the definition of the coordinates origin. The change of artesian-to-polar, and polar-to-artesian, is based on the distance of points to the coordinate origin. The same shape can be represented in a very different manner depending on the definition of the coordinate origin. Examples of methods to determine the coordinate origin are: the center of gravity, the center of the bounding box or the center of the minimal enclosing circle. Each of these methods will get a different polar description of the shape. A second drawback is the effect of shifts in a polar description. While a shift in a artesian system follows a linear map, a shift in a polar system follows a sinusoidal function. This fact makes difficult getting shape invariance to translation. A third drawback is the time complexity of this type of transforms. There are methods like that of Averbuch et al. [2001] who propose a pseudo-polar transform in order to speed up the change of coordinates, proposing concentric squares instead of concentric circles to represent shapes in a pseudo-polar space. However, this transform introduces geometrical distortions due to the approximation of circles by squares.

  The most representative descriptors in this group are: Polar Fourier transform, Fourier-Mellin transform, Radon/Hough transform, Zernike Moments and Angular Radial transform. We will treat Radon and Hough descriptors as the same kind of descriptors, since Deans [1982] justifies that we can compute Hough transform from Radon one. Hough transform has been classically used to detect segments on images [Fränti et al., 2000, Leavers, 1992]. There exist generalized versions of Hough transform trying to detect other simple shapes like circles or general curves [Ballard, 1981] but their use is not generalized. Radon descriptors are $\mathcal{R}$-Signature [Tabbone and Wendling, 2002, Tabbone et al., 2006] and the ridgelets transform introduced in chapter 3. Polar Fourier transform simply consists of computing 2D Fourier transform in polar coordinates. An example is the Generalized Fourier descriptor [Zhang, 2002]. On the contrary, the Fourier-Mellin transform computes the Fourier transform in the angular parameter whereas in the radial parameter is the Mellin Transform (which is a kind of moment function in a complex variable) [Adam et al., 2001]. Angular Radial transform decomposes a shape in an orthogonal basis which is defined

by the multiplication of a radial function and an angular function [Kim et al., 1999]. Both functions, angular and radial are defined by a parameter that determines the ART coefficients. This descriptor has been included in the MPEG-7 Standard. Finally, Zernike Moments are defined by the same angular function as ART descriptors, but the radial function is a real-valued polynomial [Chong et al., 2003].

- **Moment-based descriptors**. Another family of *2D* descriptors widely used in shape recognition are moment-based descriptors. The most well-know are invariant geometric moments described by Hu [1962]. Some algebraic combinations can make them invariant to similarity transforms.

  Legendre and Zernike moments are other moment-based descriptors used to construct invariant shape descriptors. Definitions of these moments are based on the Legendre and the Zernike polynomials, which are defined over the unitary disk. These bivariate polynomials can also be expressed in a complex way and mathematical theory proves that we can express any complex (or bivariate) function defined over the unitary disk as a Legendre or Zernike polynomial of infinite degree. Then, we can obtain an approximation of the original function, *i.e.* the shape, by truncating this infinite polynomial. The polynomial degree is the maximum moment order. Some works concerning Legendre and Zernike Moments are [Teh and Chin, 1988, Yap and Paramesran, 2005], to cite some of them.

  Zernike moments have proved to be more discriminant and robust to noise than geometric and Legendre moments. However, they are computationally more time expensive. Some comparative studies have been carried out in this direction [Chong et al., 2003].

- **Local Norm descriptors** are a group of descriptors constructed after computing a norm over the primitives. Roughly speaking when we think of Local Norm descriptors, we should imagine that we divide the space of primitives into disjoint sets. Afterwards, we compute a norm over these sets. So the name of *local norm*. These descriptors, introduced by Costa and Cesar Jr. [2001], are not as known as other descriptors like moment and Fourier descriptors, but they have some properties that make them interesting for shape description. Later, in section 3.5 we will explain in detail local norm descriptors as we apply them to ridgelets descriptors. Other examples of these descriptors are zoning and $\mathcal{R}$-Signature [Tabbone and Wendling, 2002] that can be understood as a Local norm descriptor based on the Radon transform.

  Invariance to similarity transforms will depend on the definition of each Local Norm descriptor. For instance, R-signature is invariant to translation but not to rotation whereas ridgelets local norm descriptors are invariant to shape rotation but not to shifts. More specifically, invariance to similarities will depend on the definition of the space in which we will compute each norm. If the space is invariant to a particular transform, then the descriptor will be invariant, too.

  This kind of descriptor is useful to compact and reduce the description of the shape. We concentrate the information of a set of features on a single value, its

norm. However, we probably lose discrimination capability. Therefore, we must find the trade off between discrimination and the size of descriptor.

2.3.2 Multiresolution methods

During our daily activities we do not usually pay the same degree of attention to all tasks we must carry out. If we held meticulous attention to all of them we would surely finish our working day exhausted. It would be too energy consuming. In a similar way, not all pattern recognition tasks need the same degree of information to achieve their goals. Some of them can be done simply with a shape sketch whereas other need as much information as possible. Multiresolution descriptors were born to solve these needs, by describing shapes at different resolutions depending on the problem to handle. In the literature, we can find two different approaches: *Scale Space* and *Wavelets* methods.

Space scale method basically consists of smoothing the shape by convolving it with a Gaussian kernel. This kernel depends on a parameter, σ *parameter*, which is usually referred to as the *scale parameter*. The underlying hypothesis is that the most relevant features will be preserved at rough scales, when the Gaussian kernel is bigger. Therefore, descriptors based on this method usually extract features following a pyramidal algorithm from the roughest scale to the finest one to complete the shape description. In *1D* descriptors, scale space descriptors are obtained by convolving the contours of the shape by a Gaussian kernel (first and second derivative of a Gaussian filter). They are known as Curvature Scale Space (CSS) and an example is the "primal sketch curvature" [Asada and Brady, 1984]. In *2D* descriptors a 2D Gaussian kernel is applied over the whole image. An inherent drawback of this kind of multiresolution approach is that the size of data is always the same, in spite of information reduction. It means that for rough scales the size of the descriptor is the same as for finest scales. However, in descriptors like the CSS included in the MPEG-7 Standard local maxima are extracted and used as descriptors, reducing in such way the size of the descriptor [Manjunath et al., 2002].

Other multiresolution descriptors are *wavelets* descriptors. By wavelets descriptors we mean all descriptors derived from the Multi-Resolution Analysis (MRA) theory. As MRA theory is one of the basis of ridgelets transform, later in section 3.2.2 we will explain it in detail. For a deep study we recommend [Mallat, 1989, 1999]. *1D* wavelets have been applied to contours of shapes with a relative degree of success. Some approaches are those from Khalil and Bayoumi [2000, 2001, 2002] which combine wavelets with the geometric invariant theory and works from [El Rube et al., 2006]. Concerning *2D* descriptors, wavelets have also been used directly on images to detect horizontal and vertical lines. When lines have other orientations the performance of $2D$ wavelets decreases. In graphics recognition problems, we can find lines oriented in any direction.

One of the main problems in this kind of descriptors is how to select the most suitable range of scales to represent the shape. Depending on the selected range of scales, recognition performance can vary significantly.

2.3.3 Structural descriptors

By *Structural* descriptors we denote the descriptors that take into account the structure of the shape. We mean the logical relationships (perpendicularity, adjacency, crossing,...) between the elements composing the shape. According to the definition of descriptor introduced at the beginning of this chapter —section 2.2— the vectors, or curves, extracted in this previous step will be the primitives of the structural descriptor, which ties with the classical notion of primitives that we can find in the literature. Depending on the primitives extracted from the shape, we will be able to extract different kinds of relationships. For instance, if we extract vectors we will be able to compute parallelism, adjacency or angle between them. Besides, if we extract polylines, or other complex curves, we will be able to extract information like inclusion or intersection. If primitives are regions, like in logos or other solid shapes, we can define relationships between the regions.

The first works in structural pattern recognition methods date from the end of 60s and the beginning of 70s. Recently, Conte et al. [2004] have reviewed graph matching methods defining two different taxonomies: matching algorithms and application of graph-based techniques. Nevertheless, according to our discussion in section 2.2 this classification can not be applied to descriptors as they have been defined there. Conte et al. paid attention to graph matching algorithms and their application, whereas we are interested in the descriptors themselves and more specifically in the existing structures to represent the relationships between graphical entities. Shapes are described by these set of relationships extracted from primitives. Each different shape is called a prototype and matching algorithms try to identify the correct shape comparing its descriptor with the descriptors extracted from their prototype.

We find more suitable the classification done by Lladós [1997], when he explains structural methods for graphics recognition. He distinguishes between syntactic and prototype-based descriptors. Syntactic descriptors are those determined by a Grammar [Groen et al., 1985, Lee, 1992, Messner and Bunke, 1996]. Grammar descriptors are based on formal language theory introduced by Chomsky in the middle of 50s. A grammar is a condensed representation of a large set of prototypes. From a finite set of elements and a set of rules we are able to produce a large set of prototypes in a similar way that in human language, in which we have an alphabet and the language grammar rules which permit us to produce words. This kind of representation is suitable when the number of prototype patterns is big, when common substructures among patterns are large and when the knowledge available about the structure will facilitate the grammar inference [Bunke, 1990]. If some of these factors do not hold then it will be better to use a prototype-based descriptor. Prototype-based descriptors are usually represented using strings and graph structures. They permit us to use graph theory —graph and sub-graphs isomorphisms— to compare and classify shapes that can be even partially occluded [Bunke, 1982, Fahmy and D., 1993, Habacha, 1991].

However, the computational cost in time of both kinds of representations, grammars and graphs, is high. That is one of the reasons that explains that research in this field has been focused on matching methods that try to speed up the algorithms.

2.4 Discussion

In the reviewed literature we have observed that there is not a fixed terminology to refer to the different elements involved in the shape recognition process. This fact not only makes difficult for a newcomer to understand the difference between existing descriptors but also a comparative analysis of them. Consequently, we have proposed to unify the terminology by proposing definitions of *descriptors*, *primitives* and FEM. These definitions are general enough to be applied to any kind of descriptors, independently of application or implementation issues.

Besides, shape description is a complex task that have motivated a rich variety of descriptors making difficult the task of grouping them into disjoint class. Thereby, if we want to understand descriptors behavior, we need to describe them in a flexible way. In this sense, we have proposed the notion of *primary* FEM. Thus, decomposing a descriptor in these *primary* components will help us. Firstly, to understand descriptor properties and, secondly, to propose new descriptors improving the performance of the existing ones.

In this context, we have linked some properties of descriptors to one of the three elements that take part in the process of shape description. We have characterized descriptors according to the geometry of their primitives —distinguishing between 1D and 2D descriptors. Besides, we have connected structural and multiresolution properties to the descriptor itself and, finally, the invariance and the information preserving properties have been linked to the FEM. The review of shape descriptors has been done according to these properties.

We have not pretended to do an exhaustive review of all kind of shape descriptors. We have commented the more significant descriptors for the goals of our work and we have seen how to describe them using the chosen properties and the groups of descriptors defined in this chapter. However, there are other descriptors we have not talked about as for example, those based on the Medial Axis Transform (MAT)[Blum, 1967] —also known as *skeleton*. We have seen the skeleton as a basis for other descriptors. However, the first descriptors based on this transform date from the end of sixties [Blum, 1967] and more recently [Siddiqi et al., 1999] has proposed shock graphs, which are based on the skeleton. Another example of descriptors that we have not commented are *boundary* approximation methods. These descriptors include polygonal approximation, splines, but also vector approximation or autoregression methods. For more detailed surveys about shape descriptors, referenced works like Loncaric [1998], Mehtre et al. [1997], Zhang and Lu [2004] can be consulted.

This chapter had two objectives, the first one was to offer the reader a global overview of shape descriptors, focused on 2D descriptors. The second objective was to introduce the required vocabulary to introduce a descriptor based on the ridgelets transform in the next chapter. Besides, we need to introduce the concept of *primary* FEM to articulate the description of descriptors in a flexible way. Thanks to these vocabularies we can describe the ridgelets descriptor as a multiresolution 2D descriptor, polar, information preserving.

Chapter 3

Ridgelets descriptors

In this chapter we will introduce ridgelets transform as a suitable multiresolution transform to extract features from shapes. We will first explain Radon transform and multiresolution analysis as the theoretical framework of ridgelets transform. Then, we will construct a descriptor based on ridgelets transform, proposing besides a similarity measure invariant to rotation. Finally, we will derive local norm features based on ridgelets as a useful descriptor to reduce the size of ridgelets descriptors.

3.1 Introduction

General purpose descriptors do not usually perform satisfactory enough in graphics recognition tasks. Traditionally, graphic symbols have been represented using structural descriptors [Lladós, 1997, Sánchez, 2001, Valveny, 1999], which are based on a vectorial representation of the shape. Vectorization is quite sensitive to noise [Chen et al., 1996, Nagasamy and Langrana, 1990, Tombre et al., 2000] and to distortions of sketched symbols. We can try to overcome this problem using graph-edit distances Habacha [1991], Jiang et al. [2000], Messner and Bunke [1996], grammar descriptors [Sánchez, 2001] or deformable models [Valveny, 1999]. Another possibility is to propose descriptors that do not need a vectorial representation. In this context, we can find 2D descriptors like those reviewed in the previous chapter. For instance, the F-signature [Tabbone et al., 2001] or different evolutions of the $\mathcal{R}$-Signature [Tabbone and Wendling, 2002, Tabbone et al., 2006]. The multiresolution descriptor based on ridgelets transform introduced in this chapter can be included in this second group of descriptors.

Structural approaches have traditionally been used in graphics recognition because the difference between symbols can be expressed in terms of the difference in the spatial relationships between the lines and the arcs of circumference composing

the graphic symbol. On the contrary, we can think of lines as singularities and hence, we can apply methods to detect these singularities. For example, edge detection methods [Canny, 1986] or two dimensional wavelets transform [Mallat, 1999]. However, separable wavelets can only detect isolated points. Their performance is similar to directional gradients and they are not able to describe properly more complex geometries like lines. Therefore, other singularity detectors should be investigated. In this sense, non-separable wavelets seems to enhance singularity detection. However, this kind of transform is computationally expensive and particular constraints must be applied to overcome this shortcoming. In that direction, Gabor wavelets bank filters have been used in the texture analysis domain [Manjunath et al., 2000].

Another extension of wavelets to 2D is the ridgelets transform, which was defined in the context of neuronal nets and functional approximation by Candès [1998]. His goal was to design a system with the minimal number of neurons to fit any (but reasonable) function. Candès concludes that, in the two dimensional case, the ridgelets transform is better than Fourier and wavelets transforms to approximate functions with singularities along lines, whereas its performance for other kinds of singularities is not worse. This property makes it suitable to describe images with linear singularities [Candès and Donoho, 1999, Donoho, 2000, Flesia et al., 2001]. Ridgelets transform takes the original intensity image and returns a multiscale sparse representation in the polar space where high values correspond to the parameters (θ, t) of the lines in the original image, while far away of these points the response is low (near to 0).

In fact, although it is an independent image transform, originated in the context of non-separable wavelets, the ridgelets transform can also be explained as the combination of two primary FEM: Radon/Hough transform and wavelets transform. From this point of view, the ridgelets descriptors can be defined as multiresolution, 2D, polar, information preserving and invariant to shifts and scale but not invariant to rotation. In this context, as the Radon/Hough transform have proved to be a suitable transform to detect lines [Fränti et al., 2000, Lladós, 1997, Tabbone and Wendling, 2002], we can hope that ridgelets transform, too. Besides, as ridgelets transform belongs to the family of wavelet transforms, it is also a multiresolution transform with several scales of decomposition, and it can be used for indexing purposes. Therefore, the ridgelets transform can be used as the basis for general pattern recognition tasks such as shape recognition, indexing, browsing, etc, where the main shape feature are lines. To sum up, the ridgelets transform combines advantages of both transforms, the ability to detect lines, from the Radon transform, and the multiscale property of wavelets to work at several levels of detail.

We will continue by introducing the theoretical framework needed to explain ridgelets transform. So, we will review in more detail Radon/Hough transform and wavelets transform, in the context of multiresolution analysis. Afterwards, we will construct a ridgelets basis like in Donoho [2000] to propose a descriptor based on ridgelets transform. A similarity measure to compare ridgelets descriptors will be proposed in section 3.4.2 and we will finalize by proposing a new ridgelets descriptor, based on local norm features, to reduce the size of ridgelets descriptor in section 3.5.

3.2 Theoretical Framework

Although we apply Radon transform to images, it can also be applied to scalar functions defined in a high dimensional space. The *Radon Space* is the mathematical object used to study the properties of the Radon transform. In particular, the definition of Radon Space for a two dimensional space is:

Definition 3 (Radon Space). *We call Radon space, $\mathcal{R}$, to the functional space, squared integrable, defined in the surface of the cylinder $\mathbb{R} \times \mathcal{S}^1$ and verifying the* antipodal symmetry*:*

$$F(t, \theta) = F(-t, \theta + \pi) \tag{3.1}$$

The Radon Space is a Hilbert space and most of the properties of finite vectorial spaces are still valid. For instance, we can define an inner product that will permit us to measure distances between functions as follows:

For f and g functions from the functional space of square-integrable functions in V, $\mathrm{L}^2(V)$ —where V denotes a real and finite vectorial space of dimension n— we define the inner product of f and g as:

$$\langle g, f \rangle = \int_V f(x) g(x) \mathrm{d}x$$

This inner product induces a norm given by:

$$\|f\|^2 = \int_V |f(x)|^2 \, \mathrm{d}x$$

and hence the distance between two functions will be given by $d(f, g) = \|f - g\|$. Besides, because the Radon space is a vectorial space we can find a discrete set of functions forming an orthogonal basis, such that all the functions $f \in \mathrm{L}^2(V)$ are determined by a linear combination of the functions basis. The coefficient of each function in the basis is computed by projecting the function f to the one dimensional space spanned by the orthogonal function, *i.e.* the coefficients are given by the inner product between the function f and all the functions of the basis. In this context, univariate wavelets functions are an orthogonal basis of square-integrable real functions, whereas the wavelets extension to bivariate functions is a set of bivariate basis functions of square-integrable functions defined in a plane. The definition of ridgelets descriptors lies in the construction of an orthogonal basis of ridgelets functions [Donoho, 2000]

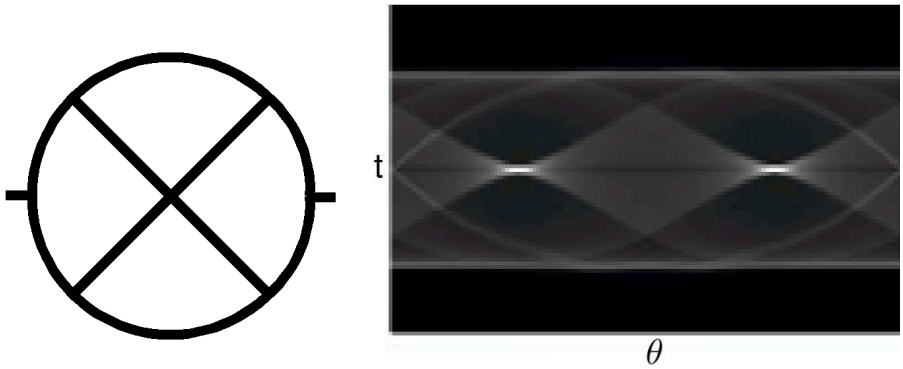


Figure 3.1: Radon transform of a graphical symbol: *Electrical J*: (a) Symbol image. (b) Radon Transform of image in (a) corresponding to angles in $[0, \pi)$

3.2.1 Radon Transform

We will continue by explaining some properties of the Radon transform that will be useful in the definition of ridgelets descriptors. Besides, we will propose a modification of the Fast Slant Algorithm proposed by Averbuch et al. [2001] to compute the Radon transform, which will correct some geometrical distortions of the original algorithm making it useful for shape recognition tasks.

For a L^2 function $f(x)$, let Rf denote the Radon transform of f, which is defined as the integral of the function along a line $\mathcal{L}_{t,\theta}$, and is expressed using the Dirac delta δ:

$$Rf(t,\theta) = \int f(x_1, x_2)\delta(x_1 \cos\theta + x_2 \sin\theta - t)dx, \tag{3.2}$$

where the phase space is $\mathbb{R} \times [0, 2\pi)$. In binary images, the value of the Radon transform at a point (t_0, θ_0) corresponds to the total number of pixels along the line $\mathcal{L}_{t_0,\theta_0}$. In figure 3.1 we can see an example of a binary image of a linear graphic symbol and its Radon Transform. Higher values can be found at angles $\pi/4$ and $3\pi/4$, which correspond to the orientation of the main lines of the symbol. The two horizontal lines in the Radon transform are due to the circle.

The Radon transform has some interesting properties when a similarity transform (translation, rotation or scaling) is applied to a given image. These properties allow to recover the Radon transform of the transformed image from the Radon transform of the original image. As they are partially inherited by the ridgelets transform, they can be used in order to define invariant descriptors to such transforms. We have summarized below these properties:

Rotation Let G_α be the rotation by angle α, which is applied to an image $f(x)$.

Then, the Radon transform of the rotated image can be expressed as:

$$R\left(f \circ G_\alpha(x)\right)(t,\theta) = Rf(t,\theta+\alpha). \tag{3.3}$$

Translation Let $T_v(x) = x + v$, $v = (v_1, v_2) \in \mathbb{R}^2$, be a translation of an image $f(x)$. Then the Radon transform of the translated image is:

$$R\left(f \circ T_v(x)\right)(t,\theta) = Rf(t + t'(\theta), \theta), \tag{3.4}$$

where $t'(\theta) = v_1 \cos(\theta) + v_2 \sin(\theta)$.

Scaling Let $H_a(x) = ax$, $a > 0$, be the scaling of an image. Then, the Radon transform of the scaled image is:

$$R\left(f \circ H_a(x)\right)(t,\theta) = \frac{1}{a} Rf(at,\theta), \tag{3.5}$$

Thus, a rotation of an image involves an horizontal shift (a shift in the angular parameter) in the Radon space, whereas a translation in the image space results in a non-linear shift in the t (vertical) parameter of the Radon space. Finally, scaling an image involves another scaling in the Radon space but only in the vertical parameter. In particular, expression (3.3) will play an important role in the definition of a similarity measure between images, in section 3.4.2, in order to achieve invariance to rotation.

There are several implementations of Radon transform. One of them consists of applying the definition of Radon transform [Rosenfeld and Kak, 1982] but it could be time expensive for some applications. Hence, other possibilities have been explored and one of them is the Fast Slant algorithm proposed by Averbuch et al. [2001]. This algorithm is based on the computation of 2D Fourier transform in concentric squares. However, this representation introduces some geometric distortion that can difficult recognition tasks. So, we have suggested a slight variation of this algorithm computed in concentric circles. Consequently, we will start the next part of this section explaining the *FSS* algorithm of Averbuch et al. and then, we will explain how we can correct the geometric distortion.

Implementation of Radon Transform

Fourier Slice theorem gives us the relationship between Radon transform and Fourier transform [Rosenfeld and Kak, 1982]. Let us denote $\mathcal{F}_1$, $\mathcal{F}_2$ the univariate and bivariate Fourier transform and P the polar-to-cartesian operator. Then:

$$\mathcal{F}_1 \circ R = P \circ \mathcal{F}_2 \tag{3.6}$$

This expression suggests us a two steps method to estimate the Radon transform of an image. Firstly, we can calculate Polar Fourier transform, the right side in equation

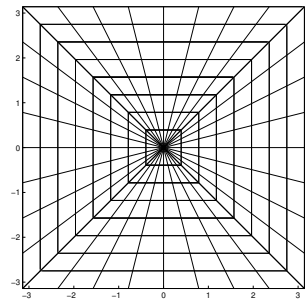


Figure 3.2: Pseudopolar grid for $N = 8$

(3.6) and after that, we can apply inverse 1D Fourier transform to recover Radon transform.

From the 2-dimensional discrete Fourier's expression, we get:

$$\hat{f}_d(\omega_1, \omega_2) = \sum_{j,k} f[j,k] e^{-i(k\omega_1 + j\omega_2)} \tag{3.7}$$

we can change this expression to polar coordinates by writing $\omega_1 = \xi \cos\theta$ and $\omega_2 = \xi \sin\theta$. Then, we replace in (3.7) by sampling $\xi_n = \frac{n}{N}$, $n = 0, \ldots, N-1$ and $\theta_m = \frac{2\pi}{M} m$, $m = 0, \ldots, M-1$.

$$\hat{f}_d[n,m] = \sum_{j,k} f[j,k] e^{-\frac{in}{N}\left(k \cos\frac{2\pi m}{M} + j \sin\frac{2\pi m}{M}\right)} \tag{3.8}$$

It is not clear how to calculate equation (3.8) in a fast way. However, Averbuch et al. define a pseudopolar grid —see figure 3.2— where circumferences are approximated by squares in order to have frequencies aligned in two separate panels: the first one, Ω^1, is composed of frequencies with angles in $[-\frac{\pi}{4}, \frac{\pi}{4})$ and the second one, Ω^2, composed of those frequencies with angles in $[\frac{\pi}{4}, \frac{3\pi}{4})$.

$$\Omega^1 = \left\{ \left(\frac{n\pi}{N}, \frac{2}{M} m \frac{n\pi}{N} \right) \quad -N \le n < N, -\frac{M}{2} \le m < \frac{M}{2} \right\}$$
$$\Omega^2 = \left\{ \left(\frac{2}{M} m \frac{n\pi}{N}, \frac{n\pi}{N} \right) \quad -N \le n < N, -\frac{M}{2} \le m < \frac{M}{2} \right\}$$

We are going to develop Averbuch et al.'s approach with panel Ω^1. For the set of frequencies in Ω^1, $\cos\theta \neq 0$, we can write:

$$\hat{f}_d[n,m] = \sum_{j,k} f[j,k] e^{-i\xi \cos\theta (k + j \tan\theta)}$$

But $\xi \cos\theta = \frac{n\pi}{N}$ and $\tan\theta = \frac{2m}{M}$. Thus:

$$\begin{aligned}\hat{f}_d[n,m] &= \sum_{j,k} f[j,k] e^{-i\frac{\pi}{N}n\left(k+j\frac{2m}{M}\right)} \\ &= \sum_j \left(\sum_k f[j,k] e^{-i\frac{\pi}{N}nk} \right) e^{-i\frac{2\pi}{N}nj\frac{2m}{M}}\end{aligned}$$

Setting $\alpha_m = \frac{2m}{NM}$, we rewrite this last expression as:

$$\hat{f}_d[n,m] = \sum_j \left(\sum_k f[j,k] e^{-i\frac{\pi}{N}nk} \right) e^{-i2\pi nj\alpha_m}, \tag{3.9}$$

where external sum is a Fractional Fourier transform with parameter α_m, that can be computed with logarithmic cost, Bailey and Swarztrauber [1991]. Panel Ω^2 can be treated in an analogous way, replacing cos by sin and tan by cot.

In this way, we can compute the right side of equation (3.6). Finally, we only have to apply 1D inverse Fourier transform to each column and we retrieve the Radon Transform.

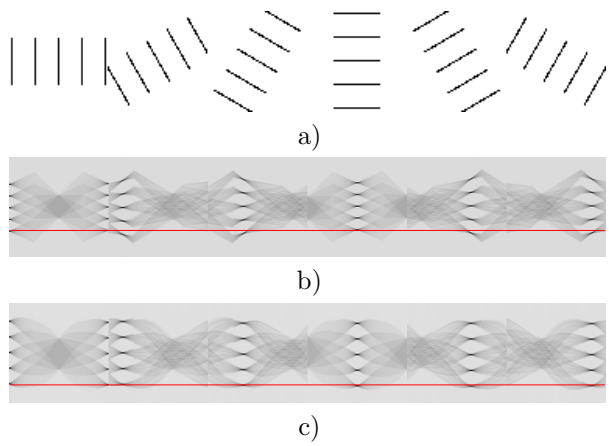


Figure 3.3: Geometric distortion of pseudo-polar coordinates in rotated shapes: a) rotated images. b) Radon transform using original *FSS*. c) corrected version

This discrete implementation results in a Radon transform that is geometrically exact to the continuous version in the same way that circles are similar to squares. There is some distortion from the ideal transform due to the use of a pseudo-polar grid instead of using a real polar grid. Figure 3.3.b shows what we mean. As we move away from the origin, the distance between local maxima of image transform and theoretical local maxima grows. The red line corresponds to the position in t direction where all maxima should be. However, for rotated images, local maxima

are farther from the origin. Furthermore, slice parameter t and angular parameter θ, are not uniformly sampled for all couples (t, θ).

We have developed a slightly different implementation of the Fast Slant Stack algorithm, where we correct the sampling of t parameter and we increase geometrical precision by replacing the last 1D inverse Fourier transform in equation (3.6) by its fractional version with parameter α according to the angular value in each column. In the pseudo-polar grid, for each angle θ, t samples are equispaced but distances between them depends on the angle. Fractional Fourier transform allows us to obtain an uniform t sampling for all θ, as we can see in figure 3.3.c, where all local maxima are located at the same value of parameter t, independently of the angle. In such way we compute Fourier transform in concentric circles although it has not been uniformly sampled in the angular parameter. As angular sampling is not important for our applications, we leave it as in Averbuch et al. approach.

3.2.2 Multiresolution Analysis

Ridgelets coefficients are obtained after applying wavelets to the Radon transform of the shape. Thus, we must explain wavelets transform in a more general framework, *i.e.* multiresolution analysis (MRA). Multiresolution analysis theory has been developed within wavelet theory. In fact it is difficult to state the border between both. One may say that multiresolution analysis theory has been inspired and motivated by wavelets study and multiresolution analysis can not be understood without wavelets. In this section, we will summarize the theory of multiresolution analysis to state some definitions and to fix the intuition about multiresolution approaches. For a detailed explanation of multiresolution analysis *A Wavelet Tour of Signal Processing*, can be consulted [Mallat, 1999].

The idea we should keep in mind when we think about wavelets is that we are approximating the original shape by an addition of details grouped according to their relevance. The name of *multiresolution* has been motivated by this characteristic of approximating the image at different levels of resolution. The notion of resolution in this field is also connected to the usual meaning of resolution when images are acquired by some input devices such as scanners. Objects in a scanned document at low resolution, are described without many details. Similarly, an image approximated at a low resolution level offers a rough approximation of the original one. In other words, we can imagine we have a kind of *magnifying glass* that we will use to examine our image. This *magnifying glass* has different zoom levels permitting us to examine our image at different resolutions. At a given resolution we will shift our over the whole image obtaining an approximation at that resolution. Thus, the *magnifying glass* will be expressed as a mathematical function, φ in $\mathrm{L}^2(V)$. The shifts and zooms of φ form a parametric family of functions that will span the space $\mathrm{L}^2(V)$. The φ function is named *scaling* function and it will determine the multiresolution approximation.

Fixing the resolution at level 0 the set of functions $\{\varphi_{0,k}(x) = \varphi(x-k) | k \in \mathbb{Z}^n\}$ will span a vectorial space $V_0 \subset \mathrm{L}^2(V)$. Then, we will project the image-function f over V_0 by computing the inner product $a_0[k] = \langle f, \varphi_k \rangle$. Hence, the function: $f_0 =$

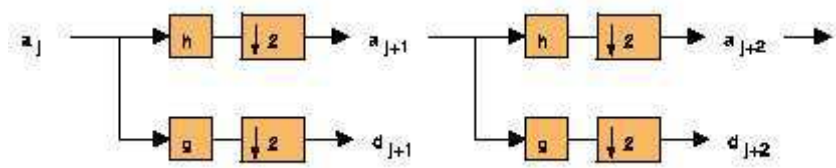


Figure 3.4: Fast Wavelet Transform, FWT is computed with a cascade of filtering with h and g followed by a factor 2 subsampling Mallat [1999]

$\sum_k a_0[k]\varphi_k(x)$ is the nearest element of V_0 to f and is named the approximation of f at resolution 0. If we dilate the function φ by multiplying their arguments by 2, then we change the resolution of φ. Afterwards, the set $\{\varphi_{1,k}(x) = \varphi(2x-k)|k \in \mathbb{Z}^n\}$ will span to a vectorial sub-space $V_1 \subset \mathrm{L}^2(V)$. The function f_1 obtained after projecting f over V_1 is another approximation of f.To see the relation between both approximations we will return to one dimensional case, in which the multiresolution analysis was initially developed. It can be proven that $V_0 \subset V_1$ and, more in general:

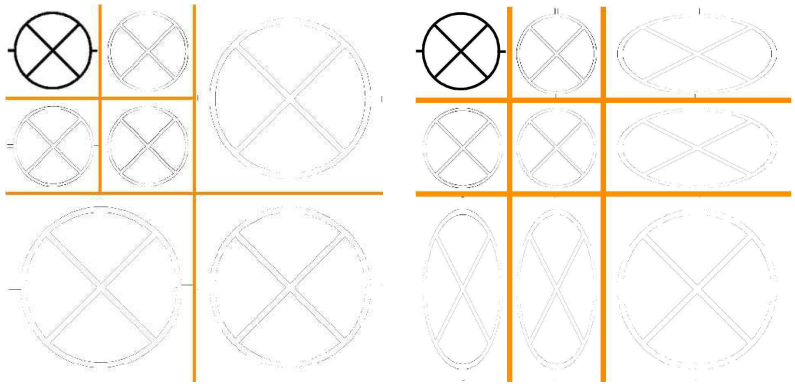
$$\forall j \in \mathbb{Z}, \quad V_j \subset V_{j+1}$$

The bigger is the resolution, the bigger is the similarity between f and its approximation. The difference between space V_j and V_{j+1} is another vectorial space, namely W_j, which is the space composed by the details of f at resolution 2^j. W_j is generated by the integer shifts of the wavelet function $\psi(x)$. In this case, the details are obtained after projecting f over the wavelets functions, ψ, i.e.. $d_j[k] = \langle f, \psi_k \rangle$. Hence, we can obtain a decomposition of functional space, $\mathrm{L}^2(V)$, into orthogonal vectorial sub-spaces:

$$\mathrm{L}^2(V) = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus W_{j_0+2} \oplus \cdots$$

To sum up, we can divide the functional vectorial space $\mathrm{L}^2(V)$ into a infinite sequence of vectorial sub-spaces. The first sub-space of this sequence is the *resolution* space which is generated by integer shifts of the *scale* function φ at a given resolution 2^{j_0}, whereas the other sub-spaces are the details sub-spaces W_j generated by the wavelet function ψ_j. Thereby, the wavelets coefficients are both the sequence $\{a_{j_0}[k]\}_k$ and $\{d_j[k]\}_{k,j \geq j_0}$. Besides, the values a_{j_0} and d_j can be quickly obtained by convolving the function f with a low-pass, h, and a high-pass, g, filters to compute the projections to V_{j_0} and W_j, respectively —figure 3.4. Both filters, h, and g are obtained from the scale and the wavelet function, respectively.

We can easily extend wavelet transform to higher dimensions by applying a MRA to each dimension. On the left-hand side of figure 3.5 we can see the cascade 2D FWT introduced by Mallat [1999]. At each iteration, we reduce the resolution of the original



(a)Cascade 2D FWT (b) Tensorial 2D FWT

Figure 3.5: Extension of FWT to bivariate functions.

image by applying at each dimension one iteration of the 1D FWT. On the contrary, on the right-hand side of the figure 3.5, we can observe the 2D FWT used by Donoho [2000] to compute the ridgelets basis. In this case, first, we apply the 1D FWT to one dimension of the image and then, we apply the 1D FWT to the other dimension —each rectangle delimited by the orange lines corresponds to wavelets coefficients at a given scale.

3.3 Ridgelets Transform

Ridgelets transform is a suitable transform to detect linear singularities in any direction. It will permit us to describe shapes, and more in particular graphic symbols, in terms of lines without the need to vectorize documents. The ridgelets transform was first defined by Candès Candès and Donoho [1999], in the context of neuronal nets and functional fitting. In his work, the activation function, ψ, is a wavelet which is used to define a ridgelets function, $\rho_{a,t,\theta}$ as follows: for each positive a, any $t \in \mathbb{R}$ and $\theta \in [0, 2\pi)$, we define $\rho_{a,t,\theta} : \mathbb{R}^2 \to \mathbb{R}^2$ as:

$$\rho_{a,t,\theta}(x_1, x_2) = a^{-1/2}\psi((x_1 \cos\theta + x_2 \sin\theta - t)/a).$$

This function is constant along lines $x_1 \cos\theta + x_2 \sin\theta = t$ and transverse to the "ridges" —lines—, it is a wavelet. Candès concludes that ridge functions have better properties than traditional activating functions and that we can approximate some kind of functions, including images, by ridge functions. Intuitively, we can think of images as composed of a superposition of ridges.

One interesting property of the ridgelets transform is coefficient sparsity. Higher coefficients are concentrated around the parameters θ and t corresponding to the

longer lines in the image. Thus, sparsity permits us to localize and to separate linear singularities into the parameter space. This is the main property that distinguish this wavelet from usual separable wavelets (Haar, Daubechies, Meyer,...) [Mallat, 1999]. On the contrary, one of the shortcomings of the continuous ridgelets transform is its discretization. However, Donoho overcomes this problem by constructing an orthonormal basis which he calls *orthonormal ridgelets basis* [Donoho, 2000]. He proposes a family of functions that form an orthogonal basis of $\mathrm{L}^2(\mathbb{R}^2)$ and then, he justifies the connection between this basis and the ridgelets functions defined by Candès in his thesis. On the forthcoming, we will use the orthonormal ridgelets basis constructed by Donoho, but keeping in mind the intuitive idea of the continuous ridgelets transform.

Taking the definition of the Donoho basis, we can see the ridgelets coefficients of an image f as the wavelets coefficients of the Radon transform of f. That means that we must define an isometry, $\mathcal{I}^{-1}$, based on the Radon transform and an orthogonal wavelet basis W_λ. Then, we can compute ridgelets coefficients with this formula:

$$WRf(\lambda) = \left[\mathcal{I}^{-1}f, W_\lambda\right] \tag{3.10}$$

where $WRf(\lambda)$ denote the λ ridgelets coefficients of image f and $[\cdot,\cdot]$ is the inner product defined in the Radon Space —definition 3.

3.3.1 Computation of Ridgelets coefficients

Algorithm: *Ridgelets coefficients*

Input: Image, f.
Output: ridgelets transform, WRf_λ.
begin:
$F = FSSf$, Computing Radon transform of f;
$\mathcal{W}_{W_\lambda}F$ decomposing F into wavelets W_λ
return ridgelets coefficients of image f.
end:

Algorithm 3.1: Computing ridgelets coefficients using *FSS* algorithm.

Donoho proves the existence of orthonormal ridgelets basis in a constructive way, *i.e.* the proof also gives the main outline of the algorithm for computing ridgelets —Fig. 3.1. The algorithm, which is deeply explained in Flesia et al. [2001], basically consists of computing the Radon Transform using the *Fast Slant Stack* algorithm and then, applying an usual wavelets decomposition. First, the Radon transform of f is computed using the modified version of the *Fast Slant Stack* algorithm, *FSS*

introduced in section 3.2.1. Then, taking Meyer [Auscher et al., 1991] and Lemarié-Meyer [Lemarie and Meyer, 1986] wavelets in slice and angular parameters, respectively, W_λ, the Radon transform of f is decomposed into its wavelets coefficients. More explicitly, the formula of the wavelet basis, W_λ, of the Radon Space is given by the following expression:

$$W_\lambda(t,\theta) = (\psi_{j,n}(t)w_{k,m}(\theta) + \psi_{j,n}(-t)w_{k,m}(\theta+\pi))/2. \tag{3.11}$$

After applying wavelets to the Radon space, we get a family of four-indexed ridgelets coefficients, $WRf_{j,k}[n,m]$, where j and k correspond to the scale parameters of the Meyer and Lemarié-Meyer wavelets, while n and m correspond to the shift parameters. Then, summarizing this procedure, we can rewrite formula (3.10) in this way:

$$WRf_{j,k}[n,m] = [FSSf, W_{j,k}(t-n,\theta-m)]\,. \tag{3.12}$$

This last expression could be interpreted as the projection of $FSSf$ over the subspaces defined by functions $W_{j,k}$. As the parameters of the Radon space are in the cylinder $[-1,1)\times[0,2\pi)$, we can see each subspace as a particular sampling of this cylinder, where the resolution of sampling depends on the scale parameters (j,k). We can visually represent this partition of the Radon space in figure 3.7, where each rectangle corresponds to one of these subspaces. In figure 3.6 we can see an example of decomposition of a particular image for the first twelve subspaces —numbered 1 to 12 in Fig. 3.7— each sub-image corresponds to one of 12 subspaces. In those images, gray-scale values are the values of the ridgelets coefficients corresponding to parameters θ (x-axis) and t (y-axis). Dark values represent high response, *i.e.* local maxima, of the ridgelets transform at coordinates (t,θ) and hence, it means that we have linear singularities at local maxima of ridgelets coefficients.

3.4 Image representation

With the decomposition procedure explained previously, we get a set of coefficient matrices, each matrix corresponding to a pair (j,k) of scale parameters in the wavelets basis. In this section, we describe how we can organize all these matrices to get a multiscale representation of images which depends on a single index. The main idea underlying this representation scheme is the *decomposition level, DL*. The number of matrices depends on the size of the original image, while the size of each matrix (and so, the level of detail) depends on the scale parameters (j,k). All matrices corresponding to the same scale parameter j share the same size and sampling along the direction of parameter t in the Radon space, while all matrices corresponding to the same scale parameter k share the same size and sampling along the direction of

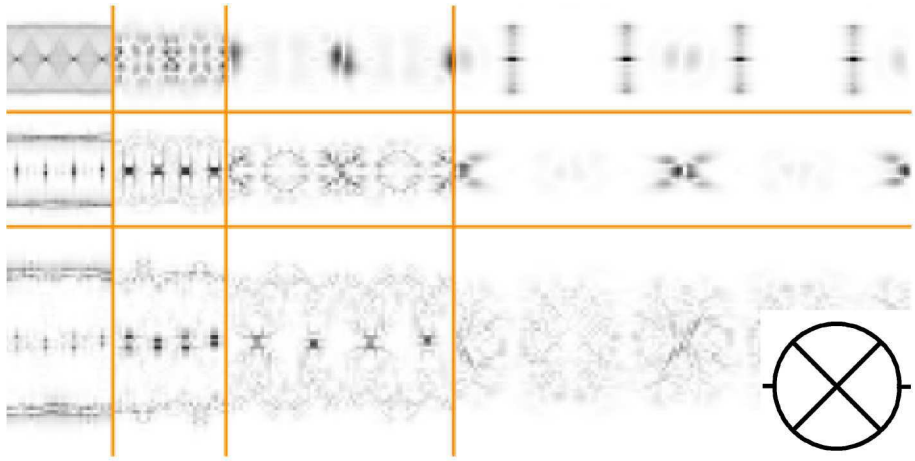


Figure 3.6: Ridgelets coefficients of symbol Electrical J. The black points correspond to high response in ridgelets transform.

parameter θ in the Radon space. We will use all these facts to define the concept of decomposition level.

As we can see in figure 3.7, we have numbered each matrix depending on its scale parameters (j, k). A decomposition level includes all matrices corresponding to given scale parameters j and k, like the shaded band in figure 3.7. We can say that a decomposition level L includes all information at the resolution provided by scale parameters j_L and k_L. For example, if we compute the ridgelets coefficients for an image of size 64×64 pixels, we will obtain 12 matrices corresponding to scale parameters $j = 1, 2, 3$ and $k = 1, 2, 3, 4$. However, with a 128×128 image, we have $j = 1, 2, 3, 4$ and $k = 1, 2, 3, 4, 5$. A new set of matrices —the shaded band numbered from 13 to 20— is added to the representation. This new set of matrices, corresponding to $j = 4$ and $k = 5$ is what we call a decomposition level, *DL*.

Thus, the representation of an image consists of a small number of decomposition levels —from 3 to 7, depending on image size— and each level L includes all matrices of coefficients generated when adding scale parameters j_L and k_L. As each decomposition level is related to a given resolution at both parameters t and θ, we have a general multiscale representation of any intensity image.

3.4.1 Definition of a shape model

For recognition purposes, we need to build a model of every pattern, based on the general representation explained in the last section. This model must be invariant to similarity transform and it must take into account the usual degradations and distortions which can be found in images —in our case, linear graphic symbols. Then, before applying the ridgelets transform to an image, we must normalize it in order

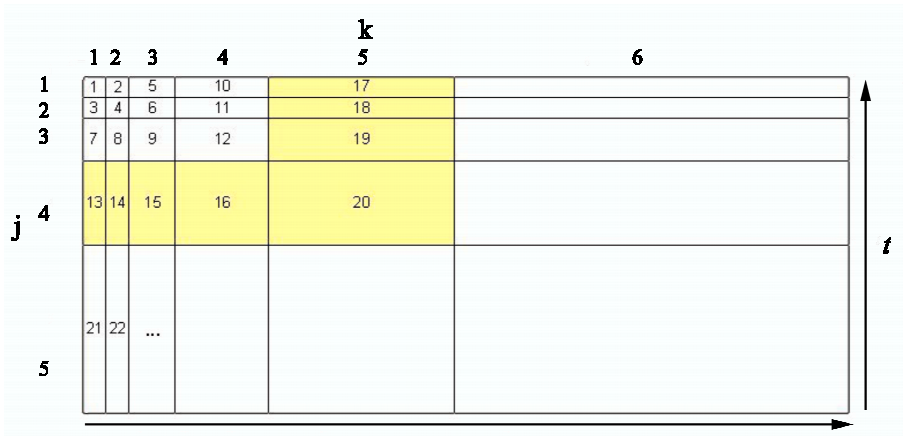


Figure 3.7: Representation of ridgelets coefficients. Indexes of couples (j, k). The shaded band corresponds to indexes in level 2.

to get such invariance. First, we apply a denoising step, a morphological opening, to reduce binary noise produced by input devices. Then, we find the minimal enclosing circle [Skyun, 1991] and we center then the image at the enclosing circle center, and scale it to the nearest dyadic radius using nearest neighbor interpolation. Using the nearest dyadic radius for scaling does not assure that all normalized images have the same size. However, this is not a problem thanks to the multiscale nature of the ridgelets transform. In the case that two images have different sizes, they will be decomposed in a different number of decomposition levels. Then, as it will be explained later, we will use only the common levels for comparing, and we will discard the finest levels, only available in the largest image. To take into account degradations and distortions, we will build the model of a symbol allowing some variability in the ridgelets coefficients of the symbol. Each line of a symbol must give a high response in the matrix of coefficients at a given location of the Radon space. However, due to noise and distortion, this response will not always be the same, nor will be at the same exact location. We can build the model of a symbol by taking each coefficient of the ideal representation, $WRf_{j,k}$, as a random variable $X_{j,k}$, that defines an area of influence around the ideal location of coefficients. Its density is defined by this expression:

$$X \sim f_X = \frac{|WRf|}{\|WRf\|_{L^1}}, \tag{3.13}$$

in which we have left scale parameters out to simplify the notation. Then, to allow variability, we smooth the description of a symbol by adding to X, a normal distribution, centered at 0, W_Σ: $Y = X + W_\Sigma$. It is known that the density of Y is obtained by the convolution of X and W_Σ. The covariance matrix Σ will be related to the degree of distortion allowed to the symbol.

3.4.2 Definition of a similarity measure

We also need to define a similarity measure to compare different images. As the representation is not invariant to rotation, we will use expression (3.3) to define a similarity measure invariant to rotation. This similarity measure is defined at each decomposition level, based on the inner product in the Radon space.

Before defining this measure in the ridgelets domain, we must return to the Radon space, $\mathcal{R}$. Following the ridgelets theory of Donoho [2000], we can take the inner product $[\cdot,\cdot]$, and its associated norm $\|\cdot,\cdot\|$:

$$[F_1, F_2] = \frac{1}{4\pi}\int_0^{2\pi}\int_{-\infty}^{\infty} F_1(t,\theta)F_2(t,\theta)dtd\theta, \tag{3.14}$$

F_1 and F_2 are in the Radon space —and refer to the Radon transforms of the symbols to compare. If we normalize F_1 and F_2 to norm 1, computing the distance between symbols is equivalent to computing the inner product:

$$[F_1, F_2] = 1 - \frac{\|F_1 - F_2\|^2}{2}.$$

Hence, using $[\cdot,\cdot]$ and expression (3.3), we can define a circular similarity measure, d, invariant to rotation:

$$d(F_1, F_2) = \max_{\alpha\in[0,2\pi)} \frac{[F_1, F_2 \circ G_\alpha]}{\|F_1\|\,\|F_2\|}, \tag{3.15}$$

where we denote by $F_2 \circ G_\alpha$ the rotation by angle α of symbol f_2. This distance is similar to the circular distance defined in Fränti et al. [2000], but it can be seen as a kind of correlation measure: $d(F, F) = 1$. Similar symbols will return values near to 1.

As we have explained in section 3.3.1, the ridgelets transform divides the Radon space into orthogonal subspaces. Thus, we can induce a metric $\mathcal{R}$ on each subspace and therefore, for each subspace indexed by parameters (j, k), expression (3.15) remains valid. However, equality in expression (3.3) is lost because the wavelet transform is not invariant to shifts in finest scales. Then,

$$WR(f \circ G_\alpha)_{j,k}[n, m] \approx WRf_{j,k}[n, m + \alpha_{j,k}], \tag{3.16}$$

From these definitions, we can define a similarity measure between two symbols S_1 and S_2 at a given decomposition level, $DL = s$, as the average of the similarity measure d between both symbols for all pairs of scales (j, k) belonging to s:

$$\bar{d}_s(S_1, S_2) = \frac{1}{|I_s|}\sum_{(j,k)\in I_s} d(X^{(1)}_{j,k}, X^{(2)}_{j,k}), \tag{3.17}$$

where d is the distance defined in (3.15), I_s is the set of scales belonging to the decomposition level s and $X_{j,k}$ is the matrix of coefficients for scale parameters (j, k). By construction, $\bar{d}_s$ is in the interval $[0, 1]$.

3.4.3 Definition of a combination rule

So far, we have defined a similarity measure for each DL. However, when we compare two shapes, we usually need to return a single value stating whether two shapes belong, or not, to the same class. In this section we have designed a combination rule, CR, which is based on a voting scheme after taking into account two evidences obtained from the analysis of some preliminary experiments with ridgelets descriptors in Ramos Terrades and Valveny [2004]:

1. For all the decomposition levels, the ridgelets coefficients of scales: $WRf_{DL+1,s}$, where $s = 1, \ldots, DL - 1$ (the "horizontal" ridgelets coefficients in figure 3.7) yield better recognition rates than the ridgelets coefficient of scales: $WRf_{s,DL+2}$, where $s = 1, \ldots, DL+1$ (the "vertical" ridgelets coefficients in figure 3.7). Then, in the computation of the distance defined in equation 3.17, we have only used the "horizontal" scales of the decomposition level.

2. The first decomposition level, $DL = 0$, reaches better recognition rates than the other decomposition levels. Then, we will give a greater weight to this level in case of equality.

For that, first we have built the ridgelets model, M, of every class of shapes in the database at each DL —as explained at the beginning of section 3.4. So, we take an ideal image of every symbol, we normalize it to shifts and scaling, we compute its ridgelets coefficients and, finally, we smooth the coefficients at each scale. In such way, we have obtained a symbol description at several decomposition levels, robust to vectorial distortion and degradation. Then, the CR algorithm sketched in Algorithm 3.2, works as follows: For each unknown image, WRf, we have also normalized it and we have computed its ridgelets coefficients. Then, at each DL, we compute the distance between the image and all the symbol models and we keep the most similar at each DL. Finally, we vote among all DLs, returning the model with the highest score. In case of equality between two, or more symbols, the winner is the model identified at level 0 (we have done it by multiplying by 1.1 the model provided in $DL = 0$).

3.5 Local Norm descriptors based on ridgelets transform

The size of ridgelets descriptors constructed in the last section is relative big for pattern recognition purpose. The usual size for descriptors is around 16 or 30 features

Algorithm: CR

Input: WRf: ridgelets descriptors,
M: list of ridgelets models.
Output: m_0: the nearest model
d_0: similarity value

```
begin:
    for all decomposition levels: s = 1, ..., DL,
        for all models in M: m,
            Compute d̄_s(WRf, M_m);        (using 3.17)
        endfor
        Get the nearest model: m_s = arg max_m d̄_s;
        v_s(m_s) = 1 otherwise 0;
    endfor
    return: m_0 = arg max_m 1.1 v_1 + Σ_{s=2}^{DL} v_s;
end:
```

Algorithm 3.2: Combination rule of ridgelets descriptors

for shape. However, if we work with images of size of 16x16 the size of ridgelets descriptors is around 1024 coefficients, most of them with values near zero. We have explored local norm descriptors to reduce the size of ridgelets descriptors.

We have already introduced local norm descriptors in chapter 2, but now we will explain them with more detail with the help of an example: Zoning. In zoning, we divide an image of a shape into a squared grid and compute in each cell the area, or volume, of the shape. In this case we apply the L^1-norm and the result is a volume or an area depending on whether the shape is a binary or an intensity image. Considering the whole shape image, it could be seen as an element of a vectorial space of dimension $N \times M$ (if the image size is $N \times M$). Let us denote this vectorial space by V. Therefore, after defining a grid in the image, each cell is a vectorial subspace, namely F_l. These ideas can be formalized with the following definition:

Definition 4 (Local norm features). *Let V be a vectorial space and $\{F_l\}_l$, a family of orthogonal vectorial sub-spaces of V,* i.e. $V = \bigoplus_l F_l$. *Let g be an element of V. We can compute its $L^\alpha(F_l)$ norm for $1 \leq \alpha < +\infty$:*

$$v_g(l) = \|g\|^\alpha_{L^\alpha(F_l)} = \int_{F_l} |g|^\alpha \, dx \tag{3.18}$$

The set of values $v_g(l)$ computed over each subspace is the local norm descriptor. Besides, different local norm descriptors can be obtained by defining these subspaces in multiple ways. Thereby, concerning ridgelets coefficients, we have seen that they are indexed by four parameters: $WRf_{j,k}[n,m]$, where j and k correspond to the scale parameters, while n and m correspond to the shift parameters. These coefficients are obtained after projecting Radon coefficients over the subspaces $V_{j,k}$ defined by

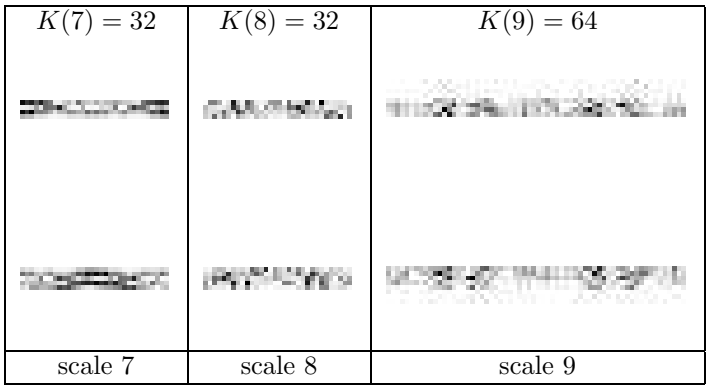


Figure 3.8: Coefficients from F_1^3. They are distributed on three different matrices of sizes $J(4) \times K(7)$, $J(4) \times K(8)$ and $J(4) \times K(9)$. Here $J(4) = 64$

functions $W_{j,k}$. We can index functions $W_{j,k}$ by the pairs (j, k), according to the representation of figure 3.7, where each rectangle is a matrix of coefficients. We have grouped these subspaces taking into account the concept of *decomposition level*, *DL*, which is consistent with image resolution. Each *decomposition level DL* is composed of a succession of spaces $V_{j,k}$, drawing a *turned "L"* in figure 3.7. According to the experimental results explained in Ramos Terrades and Valveny [2004], we will only consider some of these subspaces $V_{j,k}$ to define the subspace V_{DL} associated to a given decomposition level *DL*. Then we define V_{DL} as:

$$V_{DL} = \{V_{j,k} | j = DL, \quad k = 1, \ldots, DL\} \tag{3.19}$$

Given the vectorial spaces V_{DL} defined in equation (3.19), we can split them up into a set of subspaces F_l^{DL} by grouping basis $W_{j,k}$ in convenient sets. Then, the parameter l will run over rows of matrix coefficients while we will consider all columns in order to obtain a representation stable under rotation. Thus, we can see a strip, of width 4 —in figure 3.8— of ridgelets coefficients. In this way, the first local norm feature at the third decomposition level will be obtained after computing the L^{α}-norm of these ridgelets coefficients. Then F_l^{DL} can be defined as:

$$F_l^{DL} = \left\{ W_{DL,s}(t-n, \theta-m) \in V_{DL} \left| \begin{array}{l} 0 \leq n - l < 4, \\ 0 \leq m < K(s) \end{array} \right. \right\}$$

where $l = 0, 1, \ldots, \frac{J(DL)}{2} - 1$ and $J(DL)$ and $K(s)$ are the size along j and k directions of the matrix of coefficient $W_{DL,s}$. We only need to take l up to $\frac{J(DL)}{2}$, due to antipodal symmetry inherited from the definition of Radon Space —expression (3.1). Hence, we can compute L^1-norm on F_l using the formula (3.18). Thus, the feature vector

LNR^{DL} will be a vector of length $\frac{J(DL)}{2} - 1$ where each component is the volume of WRf on F_l^{DL}. Particularizing formula (3.18) we obtain:

$$LNR_{WRf}^{DL}(l) = \sum_{s=0}^{DL-1} \sum_{n=l}^{l+3} \sum_{m=0}^{K(s)} |WRf_{DL,s}[n,m]|^2 .$$

3.6 Discussion

In this chapter we have introduced a multiresolution descriptor based on the ridgelets transform able to properly represent shapes where "lines" are the relevant features. In this context, we have extracted the information related to the position of lines by applying a mathematical transform over the whole image —used as *primitive*— that avoids the use of vectorization methods. Hence, we can overcome some of the vectorization drawbacks with the presence of noise and vectorial distortions.

The construction of ridgelets descriptors has been done as follows: First of all, we have grouped ridgelets coefficients according to the scale parameters, defining each of these groups of coefficients as a *descriptor* —following the definition of descriptor introduced in Chapter **??**. Secondly, we have grouped ridgelets descriptors according to *decomposition levels*, which is a concept introduced in order to describe shapes having a different number of ridgelets descriptors in a coherent manner. Besides, with the help of the decomposition level, we can sort ridgelets descriptors using only a single subindex. Finally, we have proposed a similarity measure defined at each scale of ridgelets descriptors that has been extended to all scales belonging to the same decomposition level. Thereby, we can compare different shapes at different decomposition levels. Afterwards, thanks to the design of a combination rule based on a voting scheme, we can obtain a single similarity measure which permits us to compare shapes having different number of decomposition levels. The definition of this rule has been inspired in evidences obtained from the analysis of some preliminary results (cf. section 3.4.3) .

Unfortunately, this definition of ridgelets descriptors has two intrinsic drawbacks: the number of features, which is relatively huge regarding other shape descriptors, and the definition of a non heuristic combination rule based on the performance of ridgelets descriptor (at each scale) in recognition tasks. In this sense, we have reduced the size of the descriptor by introducing a new local norm descriptor based on ridgelets descriptor. Concerning the combination rule to be applied to ridgelets descriptor, in the next chapter we have tackled this problem by developing a general theoretical framework for combining classifiers.

Chapter 4

Classifier Fusion

Ridgelets descriptor can be considered as a set of shape descriptors. In this sense, the following question is raised: How can we combine ridgelets descriptors in order to reduce as much as possible classification errors? In this chapter, we will tackle this problem from the perspective of classifier fusion. In this context, we will review the state of the art of classifier fusion and we will propose a linear combination rule, which minimizes the classification error under some constraints in two-class classifiers.

4.1 Introduction

In the previous chapter, we have introduced a multiresolution descriptor based on the ridgelets transform. Although it can be considered as a single descriptor, it offers a shape representation divided into groups of coefficients according to the scale parameters of the ridgelets basis, $WR_{j,k}$ (cf. (3.12)). In order to combine all these groups into a single representation, we have proposed a ridgelets combination rule, CR, (cf. algorithm 3.2) in a supervised framework [Duda et al., 2000, Jain et al., 2000], taking into account the performance of ridgelets descriptors in a set of labeled data, namely the *Oracle*. However, we can not guarantee that this CR rule performs well for other classes of shapes. It was a first approximation but there can be many other aggregation operators outperforming this CR method and therefore, we have decided to explore other combination rules in the framework of classifier fusion.

As we have explained in the introduction chapter of this dissertation, in pattern recognition many applications have to face the problem of describing a large number of complex shapes for recognition or retrieval in large databases. In some cases, besides the large number of shapes, we can find other challenges for shape description, such as the similarity among some of the shapes, shape variability (due to transformations, noise, occlusions, etc.) or the large number of images in the database. In all these cases, one of the key issues is the design of highly discriminant shape

descriptors —as we have done in chapter 3 introducing the ridgelets descriptor— in combination with the use of powerful classification and retrieval methods. However, in chapter 2 we have reviewed existing shape descriptors, seeing that we cannot find a general descriptor able to properly represent all kinds of shapes. Consequently, a lot of research has been done in finding and proposing classifiers that attenuate descriptor lacks and improve recognition rates. Some examples are mixture models of normal distributions [Duda et al., 2000] or Bernoulli distributions [Juan and Vidal, 2002] using the well-known *Expectation-Maximization* (EM) algorithm, neural networks [Tumer and Ghosh, 1996a,b], boosting classifiers [Freund and Schapire, 1996, Schapire and Singer, 1999, Skurichina and Duin, 2002] and support vector machine [Burges, 1998], who have proved to be suitable in many applications. Specially those that are specifically designed to manage problems with small number of classes in closed environments. However, Kittler [2000] justifies the use of classifier fusion methods because high discriminant classifiers are not guaranteed to be superior to other carefully designed classifiers. Besides, for general purpose problems where the number of classes begins to be high and the shapes to be recognized can be counted by thousands, these expert classifiers begins to fail. Therefore, we need to find out classifier fusion strategies permitting us to introduce new classifiers in a flexible way when new classes are included, so that we can increase the discrimination capabilities and hence, we can reduce the misclassification rates.

In this chapter, we have attacked the problem of combining information from different sources from the perspective of *classifier fusion* methods.

4.2 Classifier Fusion approaches

The strategies used for combining classifiers depend on the type of classifiers. Not all aggregation operators can be applied to any type of classifier. In this sense, Xu et al. [1992] divide classifiers into three levels and for each one define a type of classifier fusion problem:

Abstract level is composed of those classifiers whose output is an unique label j corresponding to the class of shape. In this case, the problem is: given L labels, how can we use the classifier output to build a global classifier combining all the labels to return a single label (Type 1 problem).

Rank level classifiers provide a list of ranked labels being the top element the first choice. Thus, the problem is how to re-rank the labels in a new rank level classifier (Type 2 problem).

Measurement level is formed by classifiers that associate each label with a measurement of the degree of confidence that the element belongs to the class. Then, the problem is how to combine these measurements to obtain a global measure of confidence for each class (Type 3 problem).

These three levels of classifiers cover the different scopes of application of classifier fusion. Each one has motivated different combination rules in order to tackle the combination problem. For instance, the Borda count method, which is the sum of the number of classes ranked below a class by each classifier, is defined for Type 2 problems, whereas the majority vote rule and a combination rule based on the Dempster-Shafer theory are used for Type 1 problems. Furthermore, several aggregation operators, such as the *sum*, the *product* or the *max* of the measurement of each classifiers, as well as those reviewed by Stejic et al. [2005] can be defined for Type 3 problems. Moreover, we can say that Type 1 problems are more general than Type 2 and Type 3 problems because information provided by classifier of the *abstract* level can also be obtained from *rank* and *measurement* classifiers. Therefore, combination strategies for Type 1 problems can also be applied to Type 2 and Type 3 problems.

We have constrained to linear combination rules, among all types of strategies for combination of classifiers. On the one hand, experimental results reported in other works show that the *mean* rule usually performs better than other rules [Alkoot and Kittler, 1999, Kuncheva, 2002, Tax et al., 2000]. In this context, a discussion on when it is better to average or to multiply the classifiers is given in [Tax et al., 2000] under strict probabilistic conditions. On the other hand, high accurate classifiers such as boosting, neural networks or support vector machine are based on additive models. Therefore, we will review two theoretical frameworks that have been used to explain linear combination rules: *Bayesian approach* and *logistic regression*. Bayesian approaches have been used to explain some usual combination rules like *max*, *average*, *product*, *median* and *voting*. On the contrary, logistic regression has been used to combine classifiers in [Ho et al., 1994] but also to explain boosting classifiers [Friedman et al., 1998].

4.2.1 Bayesian approach

Bayesian approaches consist of considering that classifiers return an estimation of the following probability: $C(S) = P(S|\omega_j)$. We can understand such value like the probability that a shape S can be "generated" by the class ω_j. This approach has already been used in [Xu et al., 1992] but Kittler et al. are who derive from Bayes' formula a set of expressions which justify the most common combination rules such as *sum*, *product*, *max*, *min*, *median* and *majority* rules [Kittler et al., 1998]. These rules outperform individual classifiers, being sum and max rules those with the best results.

The mathematical framework based on Bayes' formula offers a comprehensive explanation, from conditional probabilities estimation, $C(S) = P(S|\omega_j)$. However, the classification problem yields the inverse question, if we have a shape S which is the probability that S belongs to class ω_j, $P(\omega_j|S)$? We can answer this question using the Bayes' formula:

Combination rule	Bayesian approach	r.v. approach
product	$\frac{1}{L}\prod_l P(X_l\|w_j)$	$\prod_l C_l$
mean	$\frac{1}{L}\sum_l P(X_l\|w_j)$	$\frac{1}{L}\sum_l C_l$
max	$\max_l P(X_l\|w_j)$	$\max_l C_l$
min	$\min_l P(X_l\|w_j)$	$\min_l C_l$
median	$\mathrm{med}_l P(X_l\|w_j)$	$\mathrm{med}_l C_l$

Table 4.1: Examples of combination rules

$$P(\omega_j|S) = \frac{P(S|\omega_j)P(\omega_j)}{P(S)}$$

where the right side of Bayes' formula is known or can be estimated. On a classifier fusion scheme, we denote by $X = (X_1, \ldots, X_L)$ the vector formed by L descriptors representing the shape S. If we assume that the *events*: $\{X_l|\omega_j\}$ are independent, then, applying the Bayes' formula we obtain:

$$\begin{aligned} P(\omega_j|X_1, \ldots, X_L) &= \frac{P(\omega_j)P(X_1, \ldots, X_L|\omega_j)}{P(X_1, \ldots, X_L)} \\ &= \frac{P(\omega_j)\prod P(X_l|\omega_j)}{P(X_1, \ldots, X_L)} \end{aligned} \tag{4.1}$$

If we consider that all classes ω_j have the same probability, we can directly derive from this last expression the *product* combination rule. Besides, doing more assumptions on the classifiers behaviour, *sum*, *max*, *min* and *median* rules are obtained [Kittler et al., 1998] —see table 4.1.

4.2.2 Additive models: logistic regression approach

Almost all mathematical objects can be expressed like "weighted sums" (linear combination). For instance, Taylor series, Fourier series, the integral operator are examples of infinite sums that can be truncated according to error bounds. Statistics is not an exception and we can find several methods which are based on additive models to approximate data. Maybe, one of the most well-known additive methods in pattern recognition community is boosting since [Friedman et al., 1998] proved that boosting algorithms could be defined using a logistic regression approach. But Logistic regression methods have also been used in classifier fusion [Ho et al., 1994].

If we denote by $C_l(S)$ the output of a given classifier for a shape S, logistic regression methods try to fit the probability $P(\omega_j|S)$ by an additive model: $\sum_l a_l C_l(S)$. However, in general we can not ensure that values obtained with the additive model are in the range $[0, 1]$ as it corresponds to a probability. So, a classical solution is to apply the logit transform:

Name	Notation	Meaning
Shape	S	the shape to recognize
Label	Y	the class of the shape
Descriptor	$X = FEM(S)$	the descriptor computed from the shape
Prediction	$Z = C(S)$	the classifier output
Validation	$U = YZ$	the validity of the prediction

Table 4.2: Definition of random variables for binary classifiers.

$$\begin{aligned} \log \frac{P(\omega_j|S)}{1 - P(\omega_j|S)} &= \sum_l a_l C_l(S) = H_j(S) \\ P(\omega_j|S) &= \frac{e^{H_j(S)}}{1 + e^{H_j(S)}} \end{aligned} \tag{4.2}$$

Ho et al. [1994] use a logistic approach to generalize Borda count method which is, at the same time, a generalization of the majority vote. The Borda count for a class is the sum of the number of classes ranked below it by each classifier (Type 2 problems). On the one hand, it does not require training but on the other hand, this method does not take into account each classifier capabilities. All classifiers are treated equally although some of them may be more accurate than others. Logistic regression tries to overcome this lack, giving more weight to those classifiers than perform better. Conversely, Friedman et al. [1998] show that the classifier obtained after applying boosting algorithms can be interpreted as an additive logistic regression.

4.3 The problem of classifier fusion: definitions

The general problem of defining combination rules for classifier fusion can be expressed as a minimization problem of the probability error:

Problem 1 (Classifier Fusion). *Given J classes, $\{\omega_1, \ldots, \omega_J\}$ and L classifiers C_l, we want to find the optimal aggregation operator, $f_0 \in \mathcal{F}$ which minimizes as much as possible the probability of misclassification, for all classes ω_j :*

$$f_0 = \arg\min_{f \in \mathcal{F}} P\left\{f(C_1, \ldots, C_L)(S) \neq \omega_j | \omega_j\right\} \tag{4.3}$$

where S denotes a shape belonging to the class ω_j.

Thus, in order to face this Problem 1, we propose a probabilistic framework based on five r.v —summarized in table 4.2— and defined as follows:

Shape variable, S: We consider that the shape to recognize is given by the random variable S. All other the random variables depend on this variable.

Label variable, Y**,** corresponds to the set of class labels: $j \in \{1, \ldots, J\}$ for a J-class problem. When we have a two-class problem, or when we decompose the problem of classifying J classes into J binary problems, we will also define the r.v. Y_j for each class denoting whether the shape S belongs ($Y_j = 1$) or not ($Y_j = -1$) to the class. Let us remark that the values of Y depend on the shape S, which means that Y and S are dependent r.v. and can be denoted by $Y_{|S}$ or by $f(y|s)$ if f is the pdf. of Y.

Descriptor variable, X**:** It is given by the feature extraction method (FEM) used to extract the descriptors applied to the shape r.v., $X = FEM(S)$. It also depends on S, so, we will denote it $X_{|S}$ or $f(x|s)$ if f is the pdf of X.

Prediction variable, Z**,** is the r.v of classifier. $Z = C(S)$ and hence, this variable depends on S. However, as classifiers are applied to descriptors, this r.v. also depends on the descriptor r.v. X. Therefore, we will prefer to denote the prediction r.v. as a function of the descriptor X by writing $Z = C(X)$ —and $Z_{|X}$ or $f(z|x)$ if f is the pdf of Z— instead of $Z = C(S)$.

Validation variable, U tells whether the prediction is correct or not. In binary classifiers, it is defined as the product between the *label* variable and the *prediction* variable, $U = YZ$. Observe that if u is positive, then the shape s has been correctly classified according to prediction $z = C(x)$. Otherwise, the prediction is wrong.

In this probabilistic framework, the *validation* r.v. play a central role and it is closed to the concept of *margin* in machine learning algorithms [Allwein et al., 2000, Breiman, 1996, James, 2003] —roughly speaking, the margin is a number that is positive if and only if the example is correctly classified [Allwein et al., 2000]. However, classifier fusion, as raised in Problem 1, is quite general and difficult to handle. Thus, we have simplified it by considering two types of restrictions. First of all, we have taken binary classifiers. Secondly, we have considered linear operators —as we have argued in section 4.2. In this way, we have developed our theoretical framework for binary classifiers, C_j and then, we have extended it to multiclass classifiers (a.k.a. J-class classifiers) by taking the class corresponding to the classifier with the maximum response:

$$j_0 = \arg\max_j C_j$$

The two main reasons that have motivated working with binary classifiers are simplicity and flexibility. Simplicity because binary classifiers are easier to formalize than multiclass classifiers. Flexibility because each time we want introduce a new class to our system it will be enough to train the new classifier for the new class, without needing to train again those classifiers fitted for other classes.

Besides, we want to emphasize that this probabilistic framework is general to any type of classifiers, whatever be its level, although we have reduced the complexity of

the Problem 1 by working on binary classifiers, mainly belonging to the measurement level. In this sense, for classifiers in the measurement level, we can easily identify the *prediction* r.v. and the classifier. Furthermore, we can easily define binary classifiers from multiclass classifiers in the *abstract* and *rank* levels. For *abstract* classifiers, we simply define:

$$Z_j = \begin{cases} 1 & \text{if } C(S) = j \\ -1 & \text{otherwise} \end{cases}$$

and for *rank* classifiers:

$$Z_j = \begin{cases} C(S) & \text{if the first output is } j \\ -C(S) & \text{otherwise} \end{cases}$$

Furthermore, we can observe that for *rank* classifiers, Z_j, the $\text{sign}(Z_j)$ is an *abstract* classifier, which is coherent with the discussion about Type 2 and Type 1 problems done in section 4.2.

For the combination of L classifiers, we can denote the L *prediction* and the L *validation* r.v. as random vectors: $Z = (Z_1, \ldots, Z_L)$ and $U = (U_1, \ldots, U_L)$. Moreover, as we have constrained our approach to linear combination of classifiers, we can express the linear combination of any type of classifier using the standard dot product defined in $\mathbb{R}^L$: $Z_\alpha = \langle Z, \alpha \rangle$ where α is a vector of weights. Thus, given a set of L binary classifiers, Z_l, the problem of looking for an optimal operator to apply to the L classifiers turns into the problem of finding an optimal vector of *positive* weights α minimizing the probability error of the linear combination. Therefore, according to the definition of the label r.v. Y, the classification error occurs when U_α is negative:

$$\langle U, \alpha \rangle < 0 \Leftrightarrow \begin{cases} \langle Z, \alpha \rangle < 0 & \text{if } Y = 1 \\ \langle Z, \alpha \rangle > 0 & \text{if } Y = -1 \end{cases}$$

Problem 2 (Linear Combination of Classifiers). *With the precedent definitions of r.v., the problem 1 is expressed as the optimization of the following objective function:*

$$\alpha = \arg\min_\alpha P(U_\alpha < 0|S) = \arg\min_\alpha P(\langle U, \alpha \rangle < 0|S) \tag{4.4}$$

with constraints:

$$\begin{cases} \alpha_l > 0 & \textit{for all the weights.} \\ \|\alpha\|_{L^1} & (\sum_l \alpha_l = 1). \end{cases} \tag{4.5}$$

In the next section we will show how to find the optimal weights for classifiers verifying two particular sets of constraints.

4.4 Optimal Linear Combination Rules: IN and DN

In this section we have developed two linear combination rules, namely IN and DN, which find the vector of weights α which minimizes the expression (4.4) under two sets of hypothesis about the classifiers. On the one hand, the IN method assumes that the *validation* r.v. are independent and normal. On the other hand, the DN method assumes that the *validation* r.v., U are dependent and besides, that the *validation* r.v of the combination, U_α is normal for all α. Let us observe that if the random vector U follows a multivariate normal distribution, then U_α is a normal distribution for all α. Therefore, in IN and DN methods we are imposing that the pdf of U_α is a normal r.v. for all α.

Concerning the hypothesis of normal distributions, the differences between our approach and Bayesian approaches are not significant in practice. Bayesian approaches consider that classifiers return a value p in $[0, 1)$ affected by a normal, or uniform, noise [Alkoot and Kittler, 1999, Kuncheva, 2002]. For instance, if we assume that the noise of a classifier is a normal r.v., it is not difficult to see that we can express the classifier output in terms of the sum of two independent r.v.: On the one hand, a Dirac distribution centered in p and, on the other hand, a normal r.v. centered in 0 and variance σ^2. The result of adding these two r.v. is a new normal r.v. centered in p and variance σ^2. Hence, we can include the noise in the definition of the classifier and simply say that the *prediction* r.v. of the classifier follows a normal r.v. centered in p and variance σ^2. Besides, if we consider that we are working with 1-class classifiers, the label r.v only reach the value 1, *i.e.* $Y = 1$ and hence, the prediction and the validation r.v are the same r.v.

On the contrary, the hypothesis of classifier independence is assumed in many works by simplicity in the theoretical development but also because the obtained experimental results perform satisfactory. However, Kuncheva et al. [2000] illustrate how dependence can help to increase classifier performance. Which justify the analysis of independence and dependence hypotheses. Actually, these approaches assume either the independence or dependence of events: $\{S|\omega_j\}$, whereas in our proposal, we talk about dependence, or independence, of r.v.

The IN and DN methods are obtained after the "normalization" of the *validation* r.v U_α, $\bar{U}_\alpha$, in such a way that it is centered in 0 with variance 1. Thus, if we denote by μ the mean vector of U and Σ the covariance matrix of U, then using expectation and variance properties, we can compute the mean, μ_α and the variance σ_α^2 of the combination of *validation* r.v., U_α:

$$\mu_\alpha = \langle \alpha, \mu \rangle \qquad \sigma_\alpha^2 = \alpha^t \Sigma \alpha = \|\alpha\|_\Sigma^2 \tag{4.6}$$

Therefore, the probability of expression (4.4) is equivalent to:

$$P(\langle U, \alpha \rangle < 0 | X) = \int_{-\infty}^{0} g_\alpha(u|x)\mathrm{d}u = \int_{-\infty}^{\frac{-\mu_\alpha}{\sigma_\alpha}} g_\alpha^0(v|x)\mathrm{d}v \tag{4.7}$$

Finally, imposing that g_α is the density of a normal distribution, g_α^0 does not depend on α and we can find the optimal weights α by maximizing the function ϕ defined by:

$$\phi(\alpha) = \frac{\mu_\alpha}{\sigma_\alpha} = \frac{\langle \alpha, \mu \rangle}{\sqrt{\alpha^t \Sigma \alpha}} \tag{4.8}$$

Then, imposing either dependence or independence of the *validation* r.v. has permitted us to maximize expression (4.8) in two different ways. In the IN method we have found an explicit solution for the Problem 2 whereas in the DN method we must solve a constrained optimization problem.

4.4.1 IN method

In the following discussion we will prove that the IN method —sketched in the Algorithm 4.1— reaches the optimal solution for independent and normal classifiers within the study of three different cases:

1. First, we have considered that all the classifiers are normal r.v. Thus, Proposition 1 shows that the optimal weight is: $\alpha_{\mathcal{N}} = \frac{\mu_l}{\sigma_l^2}$.

2. If the variance is 0, the last expression is ∞. Then, in this case we have considered that classifiers are "almost" perfect and we have considered these classifiers are Dirac r.v. Thereby, Proposition 2 proves that the optimal weights are given by: $\alpha_l^D = \mu_l$

3. Finally, we mix normal and Dirac r.v. and we prove in Proposition 3 that the optimal weights are given by a linear combination of the optimal weights computed from normal and Dirac distributions: $\lambda_{\mathcal{N}} \alpha^{\mathcal{N}} + \lambda_D \alpha^D$.

Proposition 1. *Given L classifiers whose* validation *r.v. U_l are independent and normal: $\mathcal{N}(\mu_l, \sigma_l^2)$, the weight vector α minimizing the misclassification rate is:*

$$\alpha = \Sigma^{-1}\mu \tag{4.9}$$

Proof. As we have explained above, we must maximize:

Algorithm: *IN*

Input: *Oracle*: $\{(X_n, Y_n)|X_n$ is a vector of L descriptors$\}$
Output: H, combination of classifier
begin:
Train L classifiers for the L descriptors, h_l;
for $l = 1 \ldots L$,
Get hypothesis $z_{l,n} = h_l(X_{l,n})$;
Obtain the *validation* values: $u_{l,n} = y_n z_{l,n}$;
Compute mean and variance of u_l: μ_l, $\sigma_l 2$;
If $\sigma_l^2 = 0$,
$\alpha_l^D = \mu_l$;
else,
$\alpha_l^{\mathcal{N}} = \frac{\mu_l}{\sigma_l^2}$;
endif;
endfor;
Set: $A = \sum_{l,t=\mathcal{N}} \alpha_l^t$ and $A = \sum_{l,t=D} \alpha_l^t$;
if $A > B$,
$\lambda_{\mathcal{N}} = \frac{A-B}{2A-B}$ and $\lambda_{\mathcal{N}} = \frac{A}{2A-B}$;
else
$\lambda_{\mathcal{N}} = 0$ and $\lambda_{\mathcal{N}} = 1$;
end;
update:
$\alpha_l^{\mathcal{N}} = \lambda_{\mathcal{N}} \alpha_l^{\mathcal{N}}$;
$\alpha_l^D = \lambda_D \alpha_l^D$;
and normalize α such that: $\sum_l \alpha_l = 1$;
return $H = \sum_l \alpha_l h_l$;
end:

Algorithm 4.1: IN method

$$\phi_{\mathcal{N}}(\alpha) = \frac{\mu_\alpha}{\sigma_\alpha} = \frac{\langle \alpha, \mu \rangle}{\sqrt{\alpha^t \Sigma \alpha}} \tag{4.10}$$

Our proof is based on geometric arguments. We have expressed the function ϕ as the dot product of the mean vector μ and α, normalized by the standard deviation σ_α. First, we will apply a linear function, given by a matrix A, to the vector α setting $\tilde{\phi}(\tilde{\alpha}) = \phi_{\mathcal{N}}(A\tilde{\alpha})$. Then, we will prove that $\tilde{\phi}$ has a unique maximal point (modulus a positive scalar factor λ). Hence, ϕ reaches a unique maximal point too, because linear functions preserve the monotony of functions. Finally, we will find the optimal vector α.

We have defined $\tilde{\phi}$ by setting $A = \Sigma^{-1/2}$, which is correctly defined because Σ is the covariance (and diagonal) matrix of the L independent normal r.v. Then, $\tilde{\phi}$ is :

$$\tilde{\phi}(\tilde{\alpha}) = \frac{\langle A\tilde{\alpha}, \mu\rangle}{\sqrt{\tilde{\alpha}^t(A^t\Sigma A)\tilde{\alpha}}} = \left\langle \frac{\tilde{\alpha}}{\|\tilde{\alpha}\|}, A^t\mu \right\rangle$$

In such way we have expressed $\tilde{\phi}$ as the dot product of $A^t\mu$ and the unknown vector $\tilde{\alpha}$. But we known that the dot product reaches its maximum value when vectors are aligned and thus:

$$\tilde{\alpha} = \lambda A^t\mu$$

Plugging it into $\tilde{\phi}$ and manipulating, a bit, we obtain:

$$\begin{aligned}\tilde{\phi}(\lambda A^t\mu) &= \left\langle \frac{\lambda A^t\mu}{\|\lambda A^t\mu\|}, A^t\mu \right\rangle = \frac{\langle AA^t\mu, \mu\rangle}{\sqrt{(\mu^t A)(A^t\mu)}} \qquad (AA^t = \Sigma^{-1}) \\ &= \frac{\left\langle \Sigma^{-1}\mu, \mu\right\rangle}{\sqrt{\mu^t\Sigma^{-1}\mu}} = \frac{\left\langle \Sigma^{-1}\mu, \mu\right\rangle}{\|\Sigma^{-1}\mu\|_\Sigma} = \phi(\Sigma^{-1}\mu)\end{aligned}$$

Therefore:

$$\alpha = \Sigma^{-1}\mu \tag{4.11}$$

which finalizes the proof. □

In particular, the weight of each classifier is given by:

$$\alpha_l = \frac{\mu_l}{\sigma_l^2} \tag{4.12}$$

Observe that for classifiers with big variance and small mean, the misclassification rates are significant and thus, the obtained weights are smaller than for classifiers with small variance. In such way if the variance of a *validation* r.v. is almost zero, then the weight is almost infinity. In these cases, the model fails and we have considered that it is more suitable to approximate the *validation* r.v. by a Dirac distribution than approximate it by a normal distribution. The optimal weights for Dirac distributions are given by the following proposition:

Proposition 2. *Given L classifiers such that the validation r.v., U_l, are Dirac and independent r.v., δ_{μ_l}, centered in μ_l, the optimal weight vector α is μ.*

Proof. The sum of L Dirac distributions is again a Dirac distribution. Hence, the linear combination of L Dirac distributions is another Dirac distribution centered in $\mu_\alpha = \langle \alpha, \mu \rangle$.

For these distributions, the probability error given by the integral defined in expression (4.7) is zero. However, we can subtract μ_α to U_α:

$$P(\langle U, \alpha \rangle < 0 | X) = \int_{-\infty}^{0} g_\alpha(u|x)\mathrm{d}u = \int_{-\infty}^{-\mu_\alpha} g_\alpha^0(v + \mu_\alpha|x)\mathrm{d}v$$

obtaining that the ϕ function —defined in (4.8)— is in this case:

$$\phi_D(\alpha) = \langle \alpha, \mu \rangle \,. \tag{4.13}$$

Then, following the same geometric arguments than for normal r.v., we find that the optimal weights are given by:

$$\alpha = \mu \tag{4.14}$$

□

In real situations, we can find classifiers with good recognition rates, but not enough to be used alone. These classifiers can be approximated by Dirac distributions and must be combined with other classifiers that can be better approximated by a normal distribution. For simplicity in the notation, let us suppose that $Z_1, \ldots, Z_N$ are normal classifiers whereas $Z_{N+1}, \ldots, Z_{N+M}$ are Dirac ones. Then, the first N components in the mean vector will correspond to the normal r.v. whereas the last M components will correspond to the Dirac r.v., $\mu = (\mu_\mathcal{N}, \mu_D)$. Besides, the covariance matrix Σ (which is diagonal) decomposes as follows:

$$\Sigma = \left(\begin{array}{c|c} \Sigma_\mathcal{N} & 0 \\ \hline 0 & 0 \end{array} \right)$$

Therefore, we can denote by $\alpha_\mathcal{N}$, and α_D, the optimal weights for the normal and Dirac distributions, respectively, $\alpha = (\alpha_\mathcal{N}, \alpha_D)$. In this way, the function ϕ —defined in (4.8)— is written as:

$$\begin{aligned} \phi(\alpha) = \frac{\langle \alpha, \mu \rangle}{\|\alpha\|_\Sigma} &= \frac{\langle \alpha_\mathcal{N}, \mu_\mathcal{N} \rangle}{\|\alpha_\mathcal{N}\|_{\Sigma_\mathcal{N}}} + \frac{\langle \alpha_D, \mu_D \rangle}{\|\alpha_\mathcal{N}\|_{\Sigma_\mathcal{N}}} \\ &= \phi_\mathcal{N}(\alpha_\mathcal{N}) + \frac{\phi_D(\alpha_D)}{\|\alpha_\mathcal{N}\|_{\Sigma_\mathcal{N}}} \end{aligned} \tag{4.15}$$

We realize that ϕ is maximized when the two summands —in the right size of Eq. 4.15— are maximized (because they are allways positive). Therefore, the vector of optimal weights, when we mix normal and Dirac distribution, is a linear combination of the optimal weights for normal distribution and the optimal weights for Dirac distributions given by the Prop. 1 and the Prop. 2:

$$\alpha = \lambda_{\mathcal{N}} \Sigma_{\mathcal{N}}^{-1} \mu_{\mathcal{N}} + \lambda_D \mu_D$$

Proposition 3. *With the precedent notation, given N independent and normal classifiers and M independent and Dirac, the optimal weights α are:*

$$\alpha = \lambda_{\mathcal{N}} \Sigma_{\mathcal{N}}^{-1} \mu_{\mathcal{N}} + \lambda_D \mu_D$$

where $\lambda = (\lambda_{\mathcal{N}}, \lambda_D)$ is :

$$\begin{cases} \lambda = (\frac{A-B}{2A-B}, \frac{A}{2A-B}) & \text{if } A > B \\ \lambda = (0,1) & \text{if } A \leq B \end{cases} \tag{4.16}$$

being $A = \sum_{n=1}^{N} \frac{\mu_n}{\sigma_n^2}$ and $B = \sum_{m=N+1}^{N+M} \mu_m$.

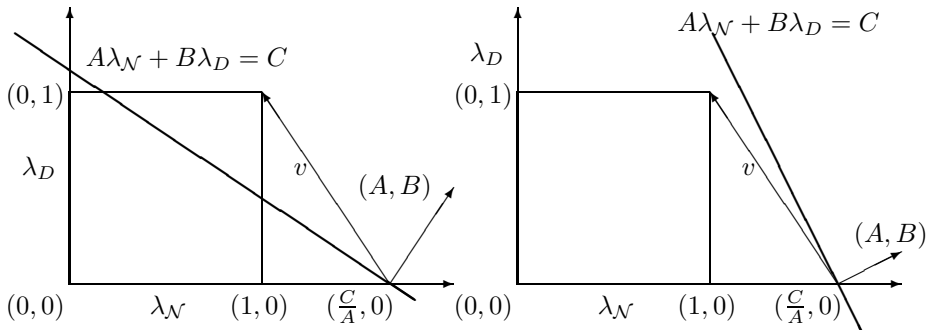
Proof. Observe that the optimal weights are obtained maximizing the the right side of Eq. (4.15) wrt $\lambda_{\mathcal{N}}$ and λ_D:

$$\arg \max_{\lambda=(\lambda_{\mathcal{N}},\lambda_D)} \phi_{\mathcal{N}}(\Sigma_{\mathcal{N}}^{-1}\mu_{\mathcal{N}}) + \frac{\lambda_D}{\lambda_{\mathcal{N}}} \frac{\phi_D(\mu_D)}{\left\|\Sigma_{\mathcal{N}}^{-1}\mu_{\mathcal{N}}\right\|_{\Sigma_{\mathcal{N}}}} \tag{4.17}$$

Moreover, we can observe, that for Dirac and normal classifiers we have always the trivial solution $\lambda = (\lambda_N, \lambda_D) = (0,1)$ which reaches the maximum value of function ϕ, (∞), cf. Exp. (4.15). This solution is easy to understand from a theoretical viewpoint. As we have "perfect" classifiers there is no need to use other types of classifiers like normal classifiers. However, in practice this trivial solution does not offer good performance, because in fact, Dirac classifiers are not "perfect". Therefore, we have to ensure that normal classifiers are considered in the combination rule by imposing a second constraint linking $\lambda_{\mathcal{N}}$ and λ_D.

As we have seen for normal classifiers and Dirac classifiers, the optimal weights are invariant to scalar factor. Then, we can consider, without loos of generality, the following constraint:

$$\sum_l \alpha_l = \lambda_{\mathcal{N}} A + \lambda_D B = C \tag{4.18}$$



(a) verifying Ineq. (4.20) (b) not verifying Ineq. (4.20)

Figure 4.1: Straight lines, $A\lambda_{\mathcal{N}} + B\lambda_D = C$, crossing the set of feasible $\lambda_{\mathcal{N},D}$: $0 \leq \lambda_{\mathcal{N}}, \lambda_D \leq 1$.

where we have denoted by $A = \sum_n \frac{\mu_n}{\sigma_n^2}$ and by $B = \sum_m \mu_{N+m}$. Besides, *a priori*, C can be any real constant value because if λ is an optimal solution maximizing ϕ, then $C\lambda$, $\forall C > 0$, is also an optimal solution. In fact, the expression (4.18) is the implicitly equation of a straight line perpendicular to vector (A, B), crossing the line given by $\lambda_D = 0$ in the point $(\frac{C}{A}, 0)$ —see figure 4.1. Therefore, depending on the choice of parameter C we will cross these segments in different points.

The remainder of this prove is divided into two parts:

- First of all, we will prove that for any combination of normal and Dirac classifiers, *i.e.* for any $A, B > 0$, $C = A$ in order to find an optimal solution λ in the square defined by the constraints: $0 \leq \lambda_D, \lambda_{\mathcal{N}} \leq 1$.
- Finally, we will prove that the optimal λ is given by the expression (4.16).

In order to see that $C = A$, we express λ_D in terms of $\lambda_{\mathcal{N}}$ from Eq. 4.18 and then, we impose the condition $\lambda_D \geq 0$. Thus, we find the following inequality:

$$C \geq A \tag{4.19}$$

To finalize this part ofthe proof, we have to show that $C \leq A$. In this sense, in order to impose that the optimal λ fall into the square given by the inequalities $0 \leq \lambda_D, \lambda_{\mathcal{N}} \leq 1$, we should impose that the straight line given by the implicit equation: $A\lambda_{\mathcal{N}} + B\lambda_D = C$ cuts the unitary square: $[0,1] \times [0,1]$. One way to impose this condition is computing the inner product between the vector (A, B) and the vector $v = (1 - \frac{C}{A}, 1)$, which is the vector joining the point $(\frac{C}{A}, 0)$ and the vertex of square $(1, 1)$ —see figure 4.1. Thus, if the inner product between these two vectors is negative, then the straight line does not cross the unitary square —see figure 4.1.b.

$$\left\langle (A,B), \left(1-\frac{C}{A},1\right)\right\rangle \geq 0 \Leftrightarrow A-C+B \geq 0 \Leftrightarrow C \leq A+B \tag{4.20}$$

In this way, if we consider both inequalities (4.19) and (4.20), we realize that for any $A, B \geq 0$:

$$\begin{aligned} C &\geq A \\ C &\leq A+B \end{aligned}$$

which are always verified if $C = A$.

Therefore, if we express $\lambda_{\mathcal{N}}$ in terms of λ_D in Eq. (4.18) —setting $C = A$— and then we plug it into the expression (4.17), we obtain the following univariate optimization problem:

$$\arg\max_{\lambda_D} \phi(\Sigma_{\mathcal{N}}^{-1}\mu_{\mathcal{N}}) + \frac{\lambda_D}{1-\frac{B}{A}\lambda_D}\frac{\phi_D(\mu_D)}{\left\|\Sigma_{\mathcal{N}}^{-1}\mu_{\mathcal{N}}\right\|_{\Sigma_{\mathcal{N}}}} \tag{4.21}$$

Thus, we have reduced the original problem —given in Eq. (4.15)— to another problem —given in Eq. (4.21)— that is expressed as the search of a maximum point in an univariate function. This problem can be solved using the standard calculus techniques. In this way, in order to find the optimal λ given by the expression (4.16), we have to differentiate the objective function given in (4.21) wrt λ_D and then we have verified that it is always positive (except in the point $\frac{A}{B}$, which is not defined). Hence, the maximal point is achieved in some point in the border of interval $[0, \min\{1, \frac{A}{B}\}]$. Then, the optimal λ is computed by setting $\lambda_D = \min\{1, \frac{A}{B}\}$ and $\lambda_{\mathcal{N}} = 1 - \frac{B}{A}\min\{1, \frac{A}{B}\}$, which proves that the optimal solution lies in one of the two segments:

$$\begin{aligned} \lambda \in \{0\} \times [0,1]_{\mathcal{N}} = 0 \\ \lambda \in (0,1] \times \{1\} \end{aligned} \tag{4.22}$$

and normalizing λ so that $\lambda_{\mathcal{N}} + \lambda_D = 1$, we obtain the solution:

$$\begin{cases} \lambda = (\frac{A-B}{2A-B}, \frac{A}{2A-B}) & \text{if } A > B \\ \lambda = (0,1) & \text{if } A \leq B \end{cases}$$

which finalizes the proof. □

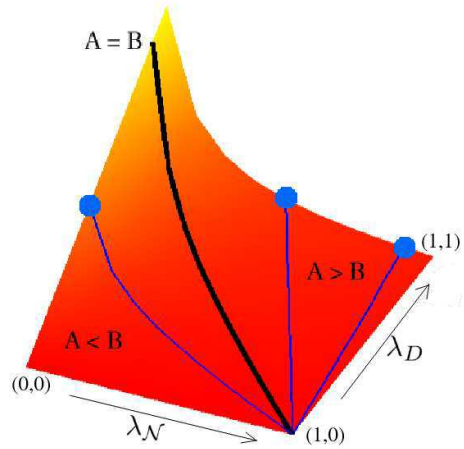


Figure 4.2: Surface defined by function (4.17). The blue lines correspond to the univariate function to maximize defined in (4.21) and the blue dots are the maximal values

Let us analyze the obtained result with the help of the figure 4.2 where we have plotted a simplification of the Eq. (4.17): $f(\lambda_{\mathcal{N}}, \lambda_D) = 1 + \frac{\lambda_D}{\lambda_{\mathcal{N}}}$. A and B are the L^1-norm of vectors μ_N and μ_D, respectively. We have also depicted several curves depending on the values of A and B. If $A \leq B$ the function (4.21) is like the function $\frac{1}{1-\lambda_D}$. It means that the "contribution" of Dirac distributions are equal or bigger than normal ones. Thereby, the maximal value is reached in the set $\{0\} \times (0, 1]$ and hence, the maximal point is $\lambda_D = 1$ and $\lambda_{\mathcal{N}} = 0$. This solution tell us that we can omit normal distributions and only use those that are Dirac. Conversely, if $A > B$ the function (4.21) is like the function λ_D and the optimal point is found in the set $(0, 1] \times \{1\}$. Besides, if $B << A$, then $\lambda_{1,2} \approx \frac{1}{2}$, which means that Dirac-like classifiers have a significant weight when they are with other classifiers with lower performance.

4.4.2 DN method

In the DN method, we have assumed dependence for *validation* r.v., U_l and besides, we have supposed that the *validation* r.v. of the combination, U_α, is a normal distribution. Therefore, the density of $\bar{U}_\alpha$ (expression (4.7)) does not depend on α and the optimal weights are obtained after maximizing (cf. Problem 2):

$$\phi(\alpha) = \frac{\mu_\alpha}{\sigma_\alpha} = \left\langle \frac{\alpha}{\sqrt{\alpha^t \Sigma \alpha}}, \mu \right\rangle \tag{4.23}$$

with the constraints:

$$\begin{cases} \alpha_l > 0 & \forall l \\ \sum_l \alpha_l = 1 \end{cases} \tag{4.24}$$

Algorithm: *DN*

Input: $\{(X_n, Y_n) | X_n$ is a vector of L descriptors$\}$
Output: H, combination of classifier
begin:
Train L classifiers for the L descriptors, h_l;
for $l = 1 \ldots L$,
Get hypothesis $z_{l,n} = h_l(X_{l,n})$;
Obtain the *validation* values: $u_{l,n} = y_n z_{l,n}$;
endfor;
Obtain weights: $\alpha = DependentWeight(U)$;
return $H = \sum_l \alpha_l h_l$;
end:

Algorithm 4.2: DN method

This constrained optimization problem is solved in the function $DependentWeight$ —see the Algortihm 4.2— once all the *validation* r.v. are estimated. Unlike for independent classifiers, for dependent classifiers the positive condition on the weights is not assured if we directly maximize function (4.23). Therefore, in the following discussion we will justify the need of the constraints given in (4.24).

We can not directly apply the arguments used for normal and Dirac, independent classifiers because the covariance matrix Σ is not diagonal. Then, there is a matrix R such that the covariance matrix $\Sigma = RDR^t$ ($|\det(R)| = 1$ and D a diagonal semi-positive defined). In this way, we can decorrelationate the *validation* random vector, U, and reduce this problem to another similar where the covariance matrix is diagonal and the *validation* r.v. mix normal and Dirac distributions.

Suppose we have $L_{\mathcal{N}}$ normal and L_D Dirac classifiers ($L_{\mathcal{N}} + L_D = L$) and let us denote by $\beta = \beta_{\mathcal{N}} + \beta_D$ the weights after decorrelating the *validation* random vector U. Afterwards, we define the projection maps : $P_{\mathcal{N}} : \mathbb{R}^L \to R^{L_{\mathcal{N}}}$ and $P_D : \mathbb{R}^L \to R^{L_D}$, given by the matrices $R_{\mathcal{N}}$ and R_D, respectively, which essentially consist of applying the matrix R^t to the vectors in $\mathbb{R}^L$. Then, the diagonal matrix D is:

$$D = \left(\begin{array}{c|c} D_{\mathcal{N}} & 0 \\ \hline 0 & 0 \end{array} \right)$$

Writing $\alpha_{\mathcal{N}} = R_{\mathcal{N}}^t D^{1/2} \beta_{\mathcal{N}}$, $\alpha_D = R_D^t \beta_D$, so that $\alpha = \alpha_{\mathcal{N}} + \alpha_D$:

$$\left\langle \frac{\alpha}{\sqrt{\alpha^t \Sigma \alpha}}, \mu \right\rangle = \left\langle \frac{R_{\mathcal{N}} D^{-1/2} \beta_{\mathcal{N}}}{\|\beta_{\mathcal{N}}\|}, \mu \right\rangle + \frac{1}{\|\bar{\beta}_{\mathcal{N}}\|} \langle R_D \beta_D, \mu \rangle$$
$$= \left\langle \frac{\bar{\beta}_{\mathcal{N}}}{\|\bar{\beta}_{\mathcal{N}}\|}, D^{-1/2} R_{\mathcal{N}}^t \mu \right\rangle + \frac{1}{\|\bar{\beta}_{\mathcal{N}}\|} \langle \beta_D, R_D^t \mu \rangle$$

and the maximal vector is obtained:

$$\arg \max_{(\bar{\beta}_{\mathcal{N}}, \beta_D)} \left\langle \frac{\bar{\beta}_{\mathcal{N}}}{\|\bar{\beta}_{\mathcal{N}}\|}, D^{-1/2} R_{\mathcal{N}}^t \mu \right\rangle + \frac{1}{\|\bar{\beta}_{\mathcal{N}}\|} \langle \beta_D, R_D^t \mu \rangle \tag{4.25}$$

with the following constraints:

$$D^{-1/2} R_{\mathcal{N}}^t \bar{\beta}_{\mathcal{N}} + R_D^t \beta_D > 0 \qquad \text{constraint: } \alpha_l \qquad > 0 \tag{4.26}$$
$$\left\| D^{-1/2} R_{\mathcal{N}}^t \bar{\beta}_{\mathcal{N}} + R_D^t \beta_D \right\|_{\mathrm{L1}} = 1 \qquad \text{constraint: } \sum_l \alpha_l \qquad = 1 \tag{4.27}$$

If we directly maximize function (4.25), then we will find an optimal solution, β, similar to that obtained in normal and Dirac, independent r.v. given in Proposition 3. However, we can not assure that that the weights α obtained from this solution β are positive because the matrix R will rotate the vector β and hence, some of the α components may be negative. So, we must impose the constraints given in (4.26) if we want maximize the objective function given in (4.25) or, alternatively, if we want maximize the objective function (4.23), we have to impose the restrictions imposed in (4.24)

4.5 Discussion

In this chapter, we have reviewed some theoretical frameworks —Bayesian framework and logistic regression— which justify common combination rules used for the aggregation of classifiers. These theoretical frameworks have motivated the introduction of a new theoretical framework which has permitted the definition of two linear combination rules depending on the distribution of classifiers. More specifically, we have proved that the IN method minimizes the classification error. In this case, the IN method assumes that the L validation r.v. —provided by the L classifiers— are independent and normally distributed. On the contrary, the DN method is able to minimize the classification error when the L validation r.v. are dependent and their linear combination is also normally distributed.

This theoretical approach to the problem of combining classifiers has been evaluated in the next chapter, where we have also validated the ridgelets descriptor introduced in chapter **??**. The experimental evaluation is concluded with the adaption of the IN and DN methods to the multiresolution properties of ridgelets descriptors.

Chapter 5

Experimental Evaluation

In this chapter we will evaluate ridgelets descriptors within the context of symbol recognition and classifier fusion methods proposed in the previous chapter. Two different experimental scenarios have been developed according to the proposals for ridgelets descriptors and classifier fusion. First of all, we have compared ridgelets descriptors to other descriptors that do not need previous vectorization of images and can be applied to segmented shapes. Secondly, we have compared our proposal of classifier fusion to other classifier fusion approaches. We have finalized this chapter by applying the fusion methods to ridgelets descriptors.

5.1 Introduction

So far we have proposed multiresolution descriptors based on the ridgelets transform (cf. chapter 3) and two classifier fusion methods (cf. chapter 4). In this chapter we will evaluate these theoretical approaches carrying out three groups of experiments. First of all, we will compare ridgelets descriptors with regard to other well-known shape descriptors. Secondly, we will evaluate classifier fusion rules comparing their performance with regard to other fusion rules training several classifiers for several shape descriptors and applying them to well-known shape databases. Finally, as the motivation of classifier fusion was to find a method to combine all scales of ridgelets descriptors, we will conclude this chapter by applying the classifier fusion methods to the ridgelets descriptor.

We have used the database introduced in the symbol recognition contest during the GREC'03 Workshop [Valveny and Dosch, 2004] to evaluate ridgelets descriptors. The GREC'03 database is formed by shapes from fifty classes of symbols from engineering, architectural and electronic domains —see figure 5.1— and it is organized in several tests according to: the number of classes, the similarity transforms applied to the symbols, the symbol degradation and the vectorial distortion. Thus, we have carried

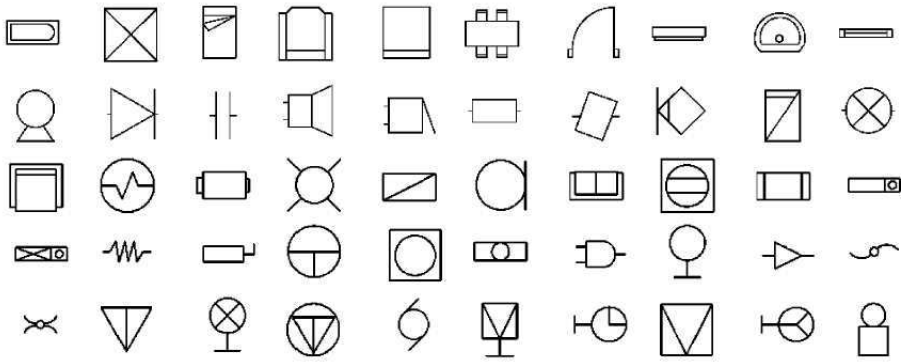


Figure 5.1: Models of GREC'03 database

out the tests of the GREC'03 contest comparing ridgelets results regarding the results achieved using the ART descriptor [Kim et al., 1999, Manjunath et al., 2002], as it is a general purpose and standard 2D descriptor included in the MPEG-7. The second set of experiments is related to classifier fusion methods. We have compared our method with other reference fusion methods such as *mean*, *max* and *median*. The evaluation has been drawn in two directions. Firstly, we have simulated an arbitrary number of classifiers that have permitted us to state the validity of the theoretical approach. Secondly, we have used well-known databases to compare the performance of our proposal regarding other classifier fusion methods. For these experiments we have used again the GREC'03 database, as it is a suitable database for the goals of this dissertation, and furthermore, a reduced version of the MNIST database —see figure 5.2— as is is a standard and general database widely used in pattern recognition. In this set of experiments, we have used several types of descriptors: ART, $\mathcal{R}$-signature and LNR.

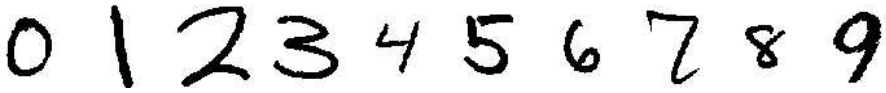


Figure 5.2: Numerals of the MNIST database

The last set of experiments consists of applying the classifier fusion methods to the combination of ridgelets descriptors. Thus, we have repeated the tests of the GREC'03 symbol recognition contest using the ridgelets descriptors combined using the classifier fusion methods.

Before describing these experiments we will briefly summarize the descriptors used in the different proposed experiments.

5.2 Descriptors

We have seen in chapter 2 that a huge number of shape descriptors have been proposed in the last years. Unfortunately, any of them can be considered as an universal shape descriptor than perform well for any kind of shapes. Depending on the shape properties some of them will perform better than others. Hence, to evaluate our ridgelets-based descriptor we have compared it to other existing shape descriptors: Angular Radial Transform (ART) [Kim et al., 1999, Manjunath et al., 2002], Fourier $\mathcal{R}$-Signature (F-R) [Tabbone and Wendling, 2002, Tabbone et al., 2006] and the Local norm ridgelets descriptors (LNR), introduced at the end of chapter 3. We have chosen these descriptors because they have similar properties to ridgelets descriptors and we can apply them in similar problems. In other words, and using the vocabulary introduced in chapter 2, we can say that all these descriptors are 2D and polar descriptors. Besides F-R, ridgelets and LNR are based on Radon transform and F-R and LNR are based on local norm descriptors.

Angular Radial Transform: We have selected this descriptor for several reasons. On the one hand, we can apply it to the same shape images as ridgelets descriptors, *i.e.* segmented and not vectorized images. On the other hand, because the ART descriptor is included in the MPEG-7 standard. This fact permits us to situate ridgelets descriptors in a more general shape description framework than that of graphics recognition.

The ART is an orthogonal transform defined on an unitary disk that consists of a complete basis decomposition using sinusoidal functions in *polar* coordinates. Each basis function $V_{n,m}$ is expressed by the multiplication of an angular function $A_m(\theta)$ and a radial one $R_n(t)$:

$$A_m(\theta) = \frac{1}{2\pi} e^{im\theta}$$
$$R_n(t) = \begin{cases} 1 & n = 0 \\ 2\cos(n\pi t) & n \neq 0 \end{cases}$$

The ART descriptor will be obtained by projecting images to the ART basis obtaining a coefficient for each function basis. Thus, it is an *information preserving* descriptor:

$$ART(f)_{n,m} = \langle f, V_{n,m} \rangle$$

Kim et al. only use 35 coefficients –setting $n = 2$ and $m = 11$– and we have considered the same parameters. Regarding similarity invariance, we achieved rotation invariance considering the modulus of coefficients, because ART coefficients are complex values. Besides, scale invariance is obtained by dividing all coefficients by the first coefficient.

In the *shape* comparison stage we have used as similarity measure the L^1 distance:

$$ART(f) - ART(g) = \sum_{n,m} ||ART(f)| - |ART(g)||$$

$\mathcal{R}$-Signature: We have selected this descriptor because it is constructed from the Radon transform of the image in a similar way to the ridgelets descriptor. In the literature, we can find several evolutions of this descriptor. First, Tabbone and Wendling [2002] introduced this descriptor by computing the squared norm of the Radon coefficients for each angular parameter of the Radon transform:

$$\mathcal{R}_f(\theta_m) = \sum_n |Rf[\rho_n, \theta_m]|^2 \tag{5.1}$$

Then, Tabbone et al. [2006] modified the previous definition of $\mathcal{R}$-signature to improve the performance of this transform. They compute the map of distances of the shape achieving a representation of the shape at different levels of distances. Then for each level they compute the $\mathcal{R}$-Signature obtaining a kind of multiresolution description. Then, to compare different shapes using this descriptor, Tabbone et al. [2006] define a similarity transform based on the modulus of the Fourier transform of the $\mathcal{R}$-signature, to achieve invariance to shape rotation.

We have used $\mathcal{R}$-signature to evaluate the performance of fusion methods. Thus, we have adopted the definition given by the expression (5.1) but using the similarity measure based on Fourier transform as it is not necessary to achieve the best recognition rates.

Local norm ridgelets: We have introduced this type of descriptor in section 3.5 to reduce the size of ridgelets descriptors. We have defined local norm descriptors based on the ridgelets representation by defining sets of descriptors, F_l:

$$F_l^{DL} = \left\{ W_{DL,s}(t-n, \theta-m) \in V_{DL} \left| \begin{matrix} 0 \leq n-l < 4, \\ 0 \leq m < K(s) \end{matrix} \right| \right\}$$

Then, the local norm descriptor is defined:

$$LNR_{WRf}^{DL}(l) = \sum_{s=0}^{DL-1} \sum_{n=l}^{l+3} \sum_{m=0}^{K(s)} |WRf_{DL,s}[n,m]|^2$$

And the similarity measure used is the Euclidean distance between the descriptors.

5.3 Ridgelets descriptors

We have used the tests from the GREC'03 contest in graphics recognition to evaluate the goodness of the ridgelets descriptors introduced in chapter 3 comparing them with the ART descriptor. The GREC'03 symbol database is composed of 512×512 binary images. Due to the *FSS* implementation of the Radon transform, the decomposition process must start with a 1024×2048 array (cf. section 3.2.1) and according to the ridgelets descriptors computation (cf. Algorithm 3.1), it yields 56 sub-matrices of ridgelets coefficients, arranged into six decomposition levels. From all these matrices of coefficients we have only used the first 20, corresponding to the first four levels of decomposition. Considering higher levels would have considerably increased the computation time, and could have introduced noise in the representation. These experiments have been carried out before developing the linear fusion methods for combining multiresolution descriptors proposed in chapter 4 and hence, we have combined the similarity values between the ridgelets descriptors using the CR method —see Algorithm 3.2.

We have grouped these tests into three groups of experiments:

1. **Robustness to resolution.** We have considered ideal representation of the symbol model, *i.e.* without applying any kind of symbol rotation and symbol degradation and we have re-scaled the symbol at different resolutions. The goal is to evaluate the stability of the multiresolution representation at different sizes of the input image.

2. **Invariance to similarity transforms.** Although, strictly speaking, ridgelets descriptors are not invariant to symbol rotation we have evaluated whether the ensemble composed of ridgelets descriptors and the similarity measure defined in (3.17) achieves invariance when shapes are *rotated*, *scaled* and *rotated* and *scaled*.

3. **Robustness to degradation and vectorial distortion.** The goal of this set of experiments is to evaluate the robustness of the shape models constructed in section 3.4.1 to handle symbol degradation and vectorial distortion.

5.3.1 Robustness to resolution

We have implemented some of the algorithms using Matlab, while others have been implemented in C, in order to reduce the computation time. However, the ridgelets transform of a 512×512 image is quite time expensive. As explained in section 3.4, depending on the original size of image we will obtain a different number of decomposition levels. Hence, a 64×64 has 3 decomposition levels whereas a 512×512 has 6 decomposition levels. We have only used the first four levels, so that we can wonder whether we can take advantage of the multiresolution properties of ridgelets to reduce the initial size of the image, and thus, reduce the computation time too, without losing discrimination power. For example, for an initial size of 512×512

we will compute 6 decomposition levels, but we will only use 4. Alternatively, if we re-scale the same image to a size of 128×128, we will only compute 4 decomposition levels, the same number as we will use for comparing. With this approach we can reduce the computation time, but at the same time, we work at lower resolution and we could lose discrimination power.

In order to answer this question, we have taken those tests containing only ideal images of each symbol. In the GREC'03 database, there are three such tests, each one with an increasing number of symbols: 5, 20 and 50 symbols. We have scaled the images contained in those tests to sizes 64×64, 128×128, 256×256 and 512×512. Then, we have applied the ridgelets transform and the classification scheme to each of the re-scaled sets of images. We have to note that for a 64×64 image, we only have three decomposition levels, whereas for bigger images, we can work with four levels, as explained before.

Results are very satisfying. We have obtained a 100% recognition rate with all tests, all image sizes and all decomposition levels except for one case, corresponding to the last decomposition level with an image size of 64×64 using the test with 50 symbols. From these results we can draw three main conclusions: first, the ridgelets transform has good properties for shape discrimination at all scales. It is to be noted that there are very similar symbols in the database. Second, ridgelets performance on shape discrimination is not degraded when the number of symbols increases. Third, the size of the image seems to have no influence on the results, except maybe in the case of 64×64 images. Therefore, in the forthcoming experiments, we will only use those tests working with the biggest set of symbols, *i.e*, the set containing 50 symbols. This way, we will test the ridgelets performance with the the hardest set of symbols in the database in terms of number of symbols and shape similarity among them. With regard to the initial size of the image, it seems that there can be a certain degradation of the performance for 64×64 images. Thus, we will use the next experiment to extract a final conclusion about this issue.

5.3.2 Invariance to similarity transforms

The next experiment aims to test the robustness to global transformations of the image, such as rotation and scaling. There are three tests in the symbol database including this kind of transformations: one test containing only rotated images of the symbols, another one containing only scaled images, and another one containing images which have been rotated and scaled at the same time, each test containing 250 images of all the symbols. We have applied our approach to these tests twice, with images resized to 64×64 and 128×128, in order to determine if recognition rates are degraded with 64×64 images.

Results are shown in table 5.1. Regarding image size, we can see that results are clearly lower for 64×64 images than for 128×128 images, specially at decomposition level 2. Even though recognition rates tend to decrease with the decomposition level, results at level 3 for 128×128 images are better than results at level 2 for 64×64 images. Therefore, we will use 128×128 images in all experiments, as a compromise

	0	1	2		
(a) Results with 64×64 *images*					
rotation	100,00%	93,20%	69,60%		
scale	99,60%	99,20%	99,60%		
rotation & scale	98,40%	91,20%	72,00%		
	0	1	2	3	**Combination**
(b) Results with 128×128 *images*					
rotation	100,00%	92,40%	93,60%	93,60%	**100,00%**
scale	100,00%	100,00%	100,00%	100,00%	**100,00%**
rotation & scale	98,40%	95,20%	92,80%	82,00%	**98,80%**

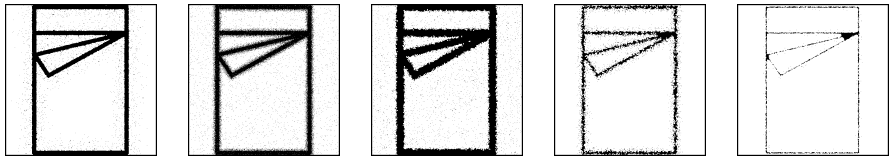
Table 5.1: Recognition rates at each decomposition level (columns) for each test. The last column in (b) corresponds to the recognition rate when combining the results using the CR algorithm

between computation time and reliability.

Table 5.1 shows that the ridgelets transform is invariant to scale but not to rotation, specially at higher decomposition levels. More specifically, non invariance to rotation is due to the fact that wavelets are not invariant to arbitrary shifts, as expressed by the inequality in expression (3.16). Rotation means a shift in the angular parameter of the ridgelets representation and therefore, a shift in the angular position of the singularities into the parameter space. However, the ridgelets response to this shift is slightly different depending on the wavelets basis and the scale parameters.

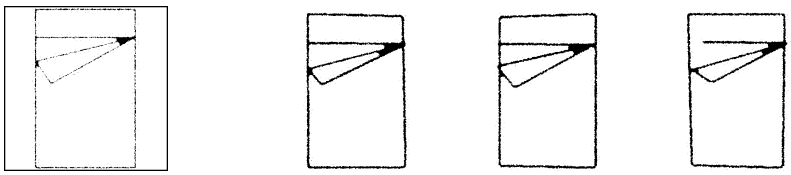
On the other hand, analyzing values from table 5.1 we can remark that recognition rates tend to decrease as decomposition level increases. This tendency is understandable after considering some properties of the ridgelets decomposition. First of all, level 0 is composed by a 32×32 matrix of ridgelets coefficients corresponding to the scaling function, not to wavelets. Therefore, in this matrix we concentrate almost all information, having a low resolution Radon Transform of a symbol, similar to the Hough Transform used by Fränti et al. [2000]. Secondly, the low percentage of hits at the last level is due to noise. For 128×128 images, level 3 corresponds to the last level of decomposition, which increases the presence of noise into symbol description. Moreover, as we normalize each matrix to 1, using L^2 norm, the noise is amplified and, consequently, values of the local maxima in the matrix decrease.

So far, we have shown results for each decomposition level separately. We have applied the voting scheme introduced in section 3.4.3 to the results from tests with rotation and scaling, and the final result is summarized in the last column of table 5.1. In all cases, the combination of levels reaches better performance than any of the four levels alone, showing strong stability in the ridgelets representation along all decomposition levels. We can also see that different decomposition levels can capture relevant features in different ways and thus, all levels are necessary and can provide relevant information for classification. Thus, we can state that the ridgelets transform is able to manage well this kind of transformations. Only when images are



(a) model 2 (b) model 3 (c) model 5 (d) model 7 (e) model 9

Figure 5.3: Some examples of degraded images. (a) Local noise. (b) Local & global noise. (c) Thickening. (d) Broken lines. (e) Thinning



(a) No distortion (model 9) (b) Level 1 (c) Level 2 (d) Level 3

Figure 5.4: Some examples of distorted images. (a) Without distortion. (b), (c) and (d) correspond to increasing distortion degrees.

both rotated and scaled the performance is slightly degraded.

5.3.3 Robustness to degradation and vectorial distortion

Finally, in the GREC'03 database there are some tests containing images with several kinds of binary degradations and vectorial distortions. These images have been generated using models of image degradation and deformation trying to simulate degradations found in real images. There are nine different models of binary degradation enumerated from *model 1* to *model 9* —examples of images generated using the most representative models are shown in figure 5.3— and three degrees of vectorial distortion —examples for each degree can be found in figure 5.4. The nine models of degradation try to simulate those binary degradations introduced by copying, printing or scanning documents [Kanungo et al., 1994]. They are based on a statistical model to add binary noise to images. On the other hand, vectorial distortions try to model shape variability introduced by hand-drawing. It is based on a statistical model to generate variations of the lines in the symbol. The generation of vectorial distortion has been limited, in the symbol database, only to 15 symbols exclusively composed of straight lines, and each test contains 75 images. Nevertheless, we have used the models of all symbols when classifying these images.

Strictly speaking, the ridgelets representation of the model of a symbol depends on two parameters: the two elements in the diagonal of matrix Σ, used to smooth the representation of the symbol —cf. section 3.4.1. This dependency is specially important when working with degraded and distorted images, as Σ have a direct relation to the amount of variability allowed in the representation of a symbol. We

	No distortion		Level 1		Level 2		Level 3	
	ridgelets	ART	ridgelets	ART	ridgelets	ART	ridgelets	ART
model 1	100,00	100,00	98,67	100,00	100,00	96,00	94,67	99,00
model 2	100,00	100,00	97,33	99,00	98,67	95,00	94,67	97,00
model 3	100,00	86,00	98,67	88,00	98,67	89,00	90,67	73,00
model 4	99,60	85,00	98,67	88,00	97,33	89,00	93,33	85,00
model 5	100,00	88,00	97,33	89,00	96,00	91,00	96,00	81,00
model 6	100,00	100,00	97,33	100,00	97,33	96,00	92,00	96,00
model 7	100,00	100,00	100,00	100,00	97,33	96,00	94,67	96,00
model 8	98,40	89,00	100,00	100,00	98,67	99,00	98,67	99,00
model 9	89,20	84,00	100,00	99,00	100,00	99,00	96,00	100,00

Table 5.2: Recognition rates for degraded and distorted images using ridgelets and ART descriptors. Results are shown for 9 degradation models (models 1-9 in rows) and 3 distortion levels (level 1-3 in columns).

have considered different values for Σ in order to determine the ideal ones. We have not reached any significant conclusion. We can see that low values of Σ are not able to manage variability and recognition results are also low. But if we increase too much the values of Σ, the recognition rates are also low because we allow too much variability and we increase confusions among symbols. Thus, we have fixed the best experimental values for Σ and we have carried out the tests with only binary degradations (involving 50 symbols), and the tests with binary degradations and each of the three degrees of distortion (involving only 15 symbols).

Table 5.2 shows the final recognition results for all these tests. We can see how the ridgelets representation is able to perform well in all models of degradation, except in model 9, where lines are thinner and more degraded. In general, with increasing degrees of vectorial distortion, recognition rates decrease, showing that our representation is more robust to binary degradations than to shape variability. These results mean that, even with binary noise, the ridgelets transform is able to accurately capture line singularities. However, as vectorial distortion implies changing the location of these singularities, the performance decreases.

We find a surprising result corresponding to *model 9* of degradation. In such case, the recognition rate is higher for distorted images than for non-distorted images. We can explain these results after visual inspection of images —figure 5.4. We can see that lines in non-distorted images are thinner than lines in distorted images. This fact is due to the way images have been generated in the database. Then, the ridgelets representation for non-distorted images of *model 9* fails to detect some of the line in the symbol and therefore, recognition rates are lower. For a final comparison, we have computed the global recognition rate for both descriptors, taking all images in the tests with degradation and distortion. These global rates are 93.64% for the ART descriptor and 97.49% for the ridgelets descriptor.

5.4 Combining Classifiers

In chapter 4 we have proposed two optimal linear rules for classifier fusion depending on the properties of classifiers, namely IN and DN. IN corresponds to classifiers whose validation r.v. is *independent* and *normal* whereas DN corresponds to *dependent* and *normal* validation r.v. In this context, we have organized the evaluation of IN and DN in two ensembles of experiments depending on the type of data used. First of all, we have generated synthetic data to simulate the classifier outputs. In this way, we can ensure that classifiers follow the desired distributions. Secondly, we have used two shape databases, one database of graphic symbols (GREC'03 database) and one database of hand-written numerals (MNIST database), to test the performance in real situations. For each type of data we have carried out the following experiments:

Synthetic data: We have defined five type of distributions depending on whether the validation r.v. of classifiers follows or not the conditions for IN and DN methods. Afterwards, we have generated an arbitrary number of outputs to carry out two experiments:

1. **Evaluation of the theoretical approach.** The goal of this experiment is linking the theoretical development and the empirical evidence. In this sense, we have verified that the empirical misclassification error fits the estimated error using the error formula given in (4.7). Besides, we have verified that IN and DN method leads to the same optimal weights when the validation r.v. are independent and normal.
2. **Comparison with other fusion methods.** In this experiment we compare IN and DN methods regarding max, $mean$ and $median$ rules.

Shape databases: We have applied the classifier fusion methods to the combination of shape descriptors using three binary classifiers: Discrete Adaboost (DAB), Linear Classifier ($LinClass$) and a normal classifier ($CNormal$), which are applied to the distribution of distances between the descriptors of the query shape and the descriptors of the model shape. Two experiments have been designed:

1. **Binary classifiers.** As for synthetic data, in this experiment we compare IN and DN method regarding max, $mean$ and $median$ rules but using binary classifiers
2. **Multiclass classifiers.** Similarly to the case of binary classifiers but introducing the $\arg\max$ function to define multi-class classifiers.

5.4.1 Using synthetic data

It is not easy to verify specific conditions on the output of real classifiers. Thus, many works have focused on generating synthetic data to simulate the output of classifiers in order to evaluate classifier fusion methods [Zouri, 2004]. In this section we will define five kind of classifiers according to different distributions. Four of these are:

$T-IN$, $T-I\bar{N}$, $T-DN$ and $T-D\bar{N}$ for respectively *independent* & normal, *independent* & non normal, *dependent* & normal and *dependent* & non normal. In this way, we can test the performance of the method depending on whether the assumptions are fulfilled or not. The fifth family of classifiers is $T-IID$, which satisfies the independence and normality but they are also identically distributed. The obtained results from this classifier have permitted us the comparison with other approaches that also make the assumption that classifiers are identically distributed [Alkoot and Kittler, 1999, Kuncheva, 2002].

Generating classifier outputs

We will explain how to generate L distributions corresponding to the output of L classifiers. Thereby, we have used three well-known properties of normal and multivariate normal distributions, X, to simulate the desired classifiers:

1. Uncorrelation of joint distribution implies independence of r.v.
2. $\langle X, \alpha \rangle$ is a univariate normal distribution for all α. This property will permit us to generate data satisfying the condition of normal distribution needed in the DN method.
3. For any affine transform, T such that $Z = TX$, Z is a multivariate normal distribution.

To generate a distribution from one of these five families of classifiers we need to generate random positives values for μ and σ. we have chosen interval $r_\mu = [.1, 1]$ and $r_\sigma = [.2, 1.2]$ —for μ and σ, respectively— similar to those defined in Alkoot and Kittler [1999]. Besides, to generate dependent distributions, we also need a random matrix, R, for rotation in $\mathbb{R}^L$. The rotation matrix R will permit us to correlationate the output of classifiers. To find a random rotation, we randomly generate a matrix A and we define $B = A + A^t$. B is symmetric. Then, the rotation matrix R verifies $B = RSR^t$ and $\det R = \pm 1$ (S is the matrix of eigenvalues of B that we will not use). For each group of distribution we will proceed as follows:

$T-IID$: We generate L independent normal distributions, U_0, for the positive examples and L independent normal distributions, U_1 for the negatives examples . Then, we obtain a random value $\mu \in r_\mu$ and σ from r_σ and we scale these $2L$ classifiers by σ and we shift them by μ. Therefore:

$$Z_{|Y=1,l} = \sigma U_{0,l} + \mu$$
$$Z_{|Y=-1,l} = -\sigma U_{1,l} - \mu$$

$T-IN$: As in $T-IID$ but obtaining L pairs (μ, σ). One for each classifier:

$$Z_{|Y=1,l} = \sigma_l U_{0,l} + \mu_l$$
$$Z_{|Y=-1,l} = -\sigma_l U_{1,l} - \mu_l$$

$T - I\bar{N}$: In real situations it is usual to approximate unknown distributions by a mixture of normal distributions. For distributions from $T - I\bar{N}$, we will generate a pair $(\mu_{pos}, \sigma_{pos})$ for the elements belonging to the class and a new pair $(\mu_{neg}, \sigma_{neg})$ for those that do not belong. In such way, the *validation* r.v. will not be normal. Hence:

$$Z_{|Y=1,l} = \sigma_{pos,l} U_{0,l} + \mu_{pos,l}$$
$$Z_{|Y=-1,l} = -\sigma_{neg,l} U_{1,l} - \mu_{neg,l}$$

$T - DN$: We generate a random mean vector μ and a random diagonal matrix D with the variance values. We obtain the rotation matrix as we have explained above and we compute the new distribution for a multivariate normal distribution:

$$Z_{|Y=1} = RDU_0 + \mu$$
$$Z_{|Y=-1} = -RDU_1 - \mu$$

$T - D\bar{N}$: As in $T - I\bar{N}$, we generate a mixture of multivariate normal distributions. In this case, we have generated two multivariate normal distribution for the negatives examples in order to ensure that the validation r.v. is not normal. Besides, we generate each multivariate normal distribution as in $T - DN$. Therefore, each multivariate normal distribution is:

$$Z_{|Y=1} = R_{pos} D_{pos} U_0 + \mu_{pos}$$
$$Z_{|Y=-1,1} = -R_{neg,1} D_{neg,1} U_1 - \mu_{neg,1}$$
$$Z_{|Y=-1,2} = -R_{neg,2} D_{neg,2} U_2 - \mu_{neg,2}$$

Thus, for each classifier we are able to generate a random number of outputs. We have tested our approach by combining 3, 9 and 20 classifiers and then, we have randomly generated, 2000 outputs, for each of them. Afterwards, to avoid the possibility that the result could be influenced by the particular distributions of 20 classifiers, we have repeated the process of generating 20 classifiers up to 500 times. In other words, for the five families of classifiers we have generated 500 tuples of 20 classifiers.

Evaluation of the theoretical approach

This experiment aims to verify the theoretical framework using data following different types of distribution. Thus, we have generated different types of distributions —as

explained in the precedent section— in order to compare the empirical error and the theoretical error depending on whether the distributions of data follow, or not, the conditions imposed by the IN and DN methods. Moreover, we also aim to compare the differences between the obtained weights using both methods.

When the *validation* r.v. satisfies the condition of normal distribution, the theoretical results permit us to estimate the misclassification rate using IN and DN methods (cf. expression (4.7)). Besides, when classifiers are independent, DN and IN must lead to the same optimal weights (because the solution is unique).

In this experiment, we have validated these theoretical results combining $N = 2000$ outputs of 20 classifiers grouped in sets of 3, 9 and 20 classifiers for the five types of distributions. For each group of classifiers we have obtained the optimal weights α with both methods and thus, we have computed both the empirical misclassification rate (EMR) and the theoretical misclassification rate (TMR):

$$EMR = \frac{1}{N}\sum_{n} 1_{\{\langle u_n,\alpha\rangle<0\}}(u_n)$$

$$TMR = P(\langle U,\alpha\rangle < 0|X) = \int_{-\infty}^{\frac{-\mu_\alpha}{\sigma_\alpha}} g_\alpha^0(v|x)\mathrm{d}v$$

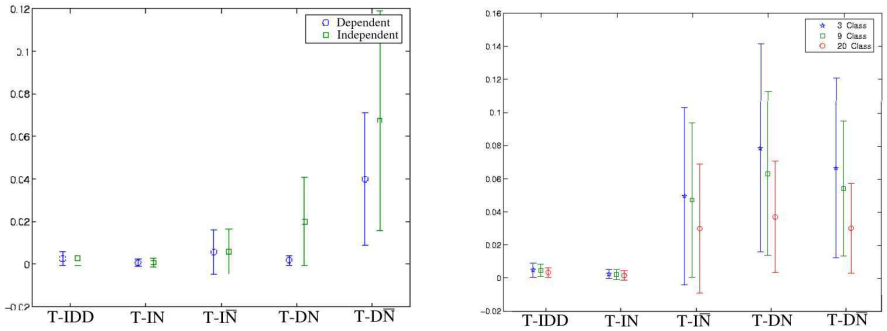
where TMR is given by the expression (4.7) and we have denoted by u_n the n-th validation vector of length 3,9 or 20

Then, we have computed the difference between both errors: $|EMR - TMR|$. On the left hand side of figure 5.5 we have plotted the mean and the standard deviation of the differences between the theoretical error and the empirical error obtained for the five families of classifiers using the IN and DN method —in blue and green in the figure 5.5. The mean and standard deviation have been computed over the five hundred distributions of each classifier combined in groups of 3, 9 and 20 classifiers, *i.e.* we have computed the EMR and TMR values of one thousand and five hundred combinations of distributions.

Besides, for the same set of classifiers and grouping them into groups of 3, 9 and 20 classifiers as well, we have computed the differences between the obtained weights using the IN and the DN method. This experiment aimed to verify that both methods reach the same solution when classifiers are independent and normal. On the right-hand side of figure 5.5 we can see the mean and the standard deviation of these differences for the five families of descriptors and grouping them into groups of 3, 9 and 20 classifiers (in blue, green and red, respectively).

Results are quite satisfying and we can extract the following two main conclusions:

- When the classifiers verify the hypothesis for the IN or DN method. There are not differences between the empirical error (EMR) and the theoretical error (TMR). Conversely, when classifiers do not verify the required conditions for the IN and DN methods the differences between the EMR and the TMR are



(a) Differences between EMR and TMR

(b) Differences between the obtained weights using IN and DN method

Figure 5.5: Validation of the classifier fusion method. Mean and Standard deviation: (a) difference between theoretical error and the empirical error (b) difference between the obtained weights using IN and DN method.

significant. Moreover, the "optimal" weights returned by these methods are quite different depending on the method used.

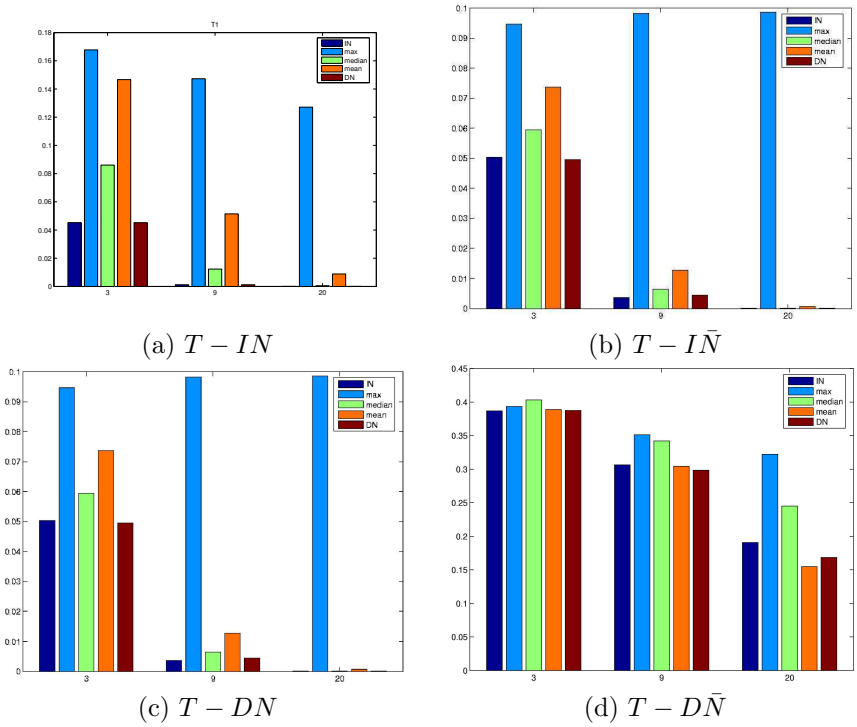
- When classifiers are *independent* and normal, the IN and DN find the same optimal weights —see first and second column in figure 5.5.b.

Therefore, the probabilistic framework introduced in Chapter 4 is suitable to face the problem of classifier fusion.

Comparing with other classifier fusion methods

The goal of this experiment is to compare IN and DN regarding other combination rules: *max*, *mean* and *median*, depending on the distribution of classifiers. Besides, this experiment links to other experimental results reported in Alkoot and Kittler [1999], Kuncheva [2002] and Tax et al. [2000] where they show that the *mean* rule usually performs better than the other rules. In Alkoot and Kittler and Kuncheva experiments, they work with 1-class classifiers that estimate a posterior probability $p \in [.5, 1]$ which is affected by a normal or uniform noise. These classifiers are equivalent to binary classifiers whose the *validation* r.v. is centered in $p - .5$.

Besides, the *mean* rule is a linear rule so that the IN and the DN methods should perform better than the *mean* rule and, consequently, than the other reference methods. To test the validity of this hypothesis we have followed the same scheme than in the previous experiment. *i.e.* For the five type of classifiers we have generated 500 tuples of 20 classifiers grouping them into groups of 3, 9 and 20 classifiers. Then we have computed the empirical classification error obtained by the different combination rules. The results obtained for classifiers belonging to $T-IID$ group are quite similar to that obtained for classifiers in the $T-IN$ group. Hence, in figure 5.6 we have only



(a) $T - IN$ (b) $T - I\bar{N}$

(c) $T - DN$ (d) $T - D\bar{N}$

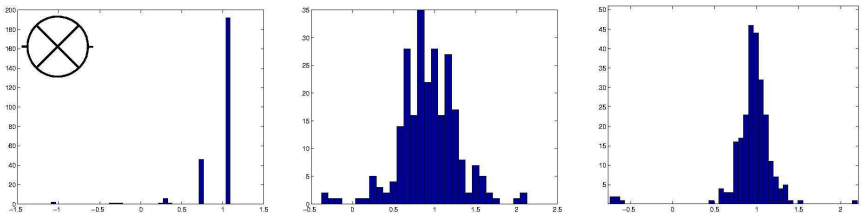
Figure 5.6: Misclassification rates using combination rules: IN, max, median mean and DN

depicted the results from $T - IN$, $T - I\bar{N}$, $T - DN$ and $T - D\bar{N}$. Results confirm our hypothesis: Linear combinations rules, such as *mean*, IN and DN outperform other combination rules like *max* and *median* rules. Besides, IN and DN outperform the *mean* rule in the five type of distributions except for the combination of 20 classifiers of the $T - D\bar{N}$ type. Besides, we can observe that in all methods (with the exception of *max* rule) the classification error is reduced when we increase the number of classifiers for the five types of classifiers even though the conditions for the IN and DN methods are not satisfied.

Later, *mean*, *max* and *median* rules have been used as reference methods in the evaluation using shape databases. Thereby, the obtained results in this part have been used as a baseline when we have compared fusion methods using shape databases. For instance, in this experiment we can observe that the *max* combination rule offers a very poor performance in the five type of classifiers. Similarly, the *max* rule has obtained a poor performance in experiments using shape databases.

5.4.2 Using Shape Databases

The experiments with synthetic data have permitted us to state the validity of the theoretical framework introduced to derive the IN and DN methods. However, in practice we can not always verify the dependence, or independence, and the distribution law followed by classifiers. In figure 5.7 we have an example of the distributions of the three binary classifiers used in our experimental evaluation for the symbol *Electrical J*. The "normality" of the DAB classifier is unclear since it returns five discrete values. The same effect can be observed in the distribution of the $LinClass$ classifier. However, concerning the distribution of the $CNormal$ classifier we can see that it is normal (by construction).



(a) DAB (b) LinClass (c) CNormal

Figure 5.7: Example of distributions of symbol *Electrical J* using DN method and using three different classifiers

As we have explained before, we have used two shape databases (GREC and MNIST) and four descriptors (ridgelets, ART, $\mathcal{R}$-signature, and *LNR*) to test the classifier fusion methods. Thereby, we have created three sets of experiments combining databases and descriptors: *GREC ridgelets*, *GREC descriptors* and *MNIST descriptors*. *GREC ridgelets* is formed by the ridgelets descriptors introduced in chapter 3 applied to the GREC database. Conversely, *GREC descriptors* and *MNIST descriptors* are composed of the set of descriptors explained in section 5.2 applied to shapes from the GREC and MNIST database, respectively. More specifically, *GREC descriptors* test is composed by six descriptors: ART, $F-R$ and four *LNR* descriptors (one for each decomposition level). Finally, the *MNIST descriptor* is formed by four descriptors: ART, $F-R$ and two *LNR* descriptors. In this case, we have two descriptors from the *LNR* because shape images are smaller and hence we have obtained less decomposition levels —we have summarized the three sets of experiments in table 5.3.

Name	GREC ridgelets	GREC descriptors	MNIST descriptors
Database	GREC	GREC	MNIST
Descriptors	WR	LNR, $F-R$, ART	LNR, $F-R$, ART

Table 5.3: Sets of experiment using two shape databases and two sets of descriptors

Concerning the learning process, we have divided the GREC database into two halves: one half for training and the other half for testing. In this database, there are

few positive examples for each class, so we have included all of them in the training set whereas the negatives examples have been randomly chosen up to 200 examples per class. Conversely, the MNIST database already contains a training set formed by 60.000 hand-written examples of the ten numerals and a test database composed of 10.000 examples. For this database we have randomly chosen 200 examples of numerals 0,1,2,3,6,8. We have not considered the ten numerals because the database is composed of hand-written numerals and descriptors are invariant to rotation. More specifically, we have discarded the numerals 4 and 9 to avoid misclassification with the numeral 6. The numeral 7 is discarded to avoid misclassification with 1 and finally, 5 as it is symmetric to 2.

The model descriptor has been obtained computing the mean of descriptors of positive examples for each class and for all the three tests. However, for the *GREC ridgelets* test we have not considered rotated symbols because ridgelets descriptors are not invariant to rotations (cf. Chapter 3). Then, we have used the same similarity measure —the Euclidean distance— to compare the query shape and the descriptors of the model shape.

We must emphasize that the main goal of these experiments is to compare combination rules regardless the poor recognition rates achieved by some of the considered descriptors either because we have used in all cases the Euclidean distance or either because these descriptors can not be the most suitable for the type of images in the database (as it is the case for the MNIST database).

Construction of classifiers

The three classifiers are trained on the distribution of distances between shape descriptors, X_n, and the descriptor used as shape model, M_j, namely $D_n = d(X_n, M_j)$. Besides, we have adopted some conventions in assigning letters to subindex. Thereby, n denote a particular shape; j is used for denoting classes and l the classifiers. Finally, the three classifiers are based on finding suitable thresholds, d_T, and dividing the space of distances into two sub-spaces: One sub-space corresponds to the shapes belonging to the class j, $T_{pos} = \{D_n < d_T\}$, and the second sub-spaces corresponds to the shapes that do not belong to the class j, $T_{neg} = \{D_n > d_T\}$.

We have chosen three types of classifiers: DAB, $LinClass$ and $CNormal$. The DAB classifier is the implementation of the Discrete Adaboost [Freund and Schapire, 1996], which has proved to perform well. On the contrary, the $LinClass$ is a linear classifier which is the simplest classifier that we can define and, besides, implicitly assumes that the distribution of distances is normal. Finally, the $CNormal$ classifier is a linear classifier which modifies the distribution of distances in order to make it normal. Hence, with these conventions on the notation, we have constructed the three classifiers as follows:

DAB classifier is a Discrete Adaboost classifier. We have used a slightly modified version of the Adaboost algorithm proposed by Schapire and Singer [1999] without considering the sign of the output —Algorithm 5.1.

Algorithm: *Generalized Adaboost*

Input: $\{(D_n, y_n)\}$
Output: H, hard classifier
begin:
Initialize $w_{0,n} = 1/N$;
for $m = 1$ to M,
Train weak learner h_m;
Get weak hypothesis $h_m : \mathbb{R} \rightarrow \mathbb{R}$;
Choose c_m;
Update:
$w_{m,n} = \frac{1}{Z_m} w_{m,n} exp(-c_m y_n h_m(D_n))$
where Z_m is a normalization factor.
endfor
return $H(D) = \sum_m c_m h_m(D)$;
end:

Algorithm 5.1: Generalized version of Adaboost Schapire and Singer [1999] .

Schapire and Singer compute the weight c_m when they train the weak classifier, which simply consists of computing a threshold value, d_T, in the distance distribution minimizing the weighted classification error. Then, the prediction of the weak classifier is given by:

$$h(D_n) = \frac{1}{2} \log \frac{W_+^a}{W_-^a} \quad \text{if } D_n \in T_a \tag{5.2}$$

where $W_+^a = \sum_{n:D_n \in T_a} w_n$ and $W_-^a = \sum_{n:D_n \in T_a} w_n$, $a = pos, neg$, and w_n are the weights assigned to each distance, D_n.

LinClass classifier is a linear classifier defined as:

$$h(D_n) = \begin{cases} r_{pos}(D_n) & \text{if } D_n \in T_{pos} \\ -r_{neg}(D_n) & \text{otherwise} \end{cases} \tag{5.3}$$

where the distance threshold, d_T is obtained from the intersection of two straight lines, r_{pos} and r_{neg}. We have defined both lines by implicitly assuming that the margin distribution of $Z_{|Y=1}$ and $Z_{|Y=-1}$ are normal distributed. Thus, r_{pos} is the tangent to the pdf of $Z_{|Y=1}$ at point $t_{pos} = \mu_{pos} + \sigma_{pos}$ and r_{neg} is the tangent to the pdf of $Z_{|Y=-1}$ at points $t_{neg} = \mu_{neg} - \sigma_{neg}$.

CNormal classifier is a linear classifier defined on a mixture of two normal distribution of data. In *LinClass* classifier we have assumed that D is normal

distributed. However, in the $CNormal$ classifier we modify the D distribution so that it becomes a mixture of two normal distributions, $D^{\mathcal{N}}$ (one distribution for the positive examples and the other distribution for the negatives examples). The three key points of the $CNormal$ classifier —sketched in Algorithm 5.2— are: the selection of the threshold distance d_T; the choice of parameters for the mixture of the two normal distributions, and the construction of the cumulative function of the distribution D, F_W. d_T is obtained after applying once the weak classifier used in the DAB classifier. In this case, the weights assigned to D_n are 1/N. Regarding the mixture parameters, the positive examples are centered in -1 and the variance is $\sigma^{\mathcal{N}}_{pos} = \frac{\sigma^2_{pos}}{\mu_{pos}}$ whereas the negative examples are centered in 1 and the variance is $\frac{\sigma^2_{neg}}{\mu_{neg}}$. Then, we compute the cumulative function, G of the mixture of the two normal distributions.

Algorithm: *CNormal*

Input: $\{(D_n, y_n)\}$
Output: H, hard classifier
begin:
Obtain d_T from the weak learner h (setting the weights to $1/N$);
Estimate the variance of the mixture of the two normal distributions;
$\sigma^{\mathcal{N}}_{pos} = \frac{\sigma^2_{pos}}{\mu_{pos}}$;
$\sigma^{\mathcal{N}}_{neg} = \frac{\sigma^2_{neg}}{\mu_{neg}}$;
Compute the cumulative function of the mixture of normal: G;
Compute the cumulative function of the distance distribution: F_W;
$D^{\mathcal{N}} = (G \circ F_W^{-1})(D)$;
return $h = -D^{\mathcal{N}}$;
end:

Algorithm 5.2: $CNormal$ Classifier

It remains to explain how to compute the cumulative function F_W. In this sense, we assign a weight, W_{pos}, to the positives examples and a weight W_{neg} such that $F_W(d_T) = \frac{1}{2}$. These weights are given by:

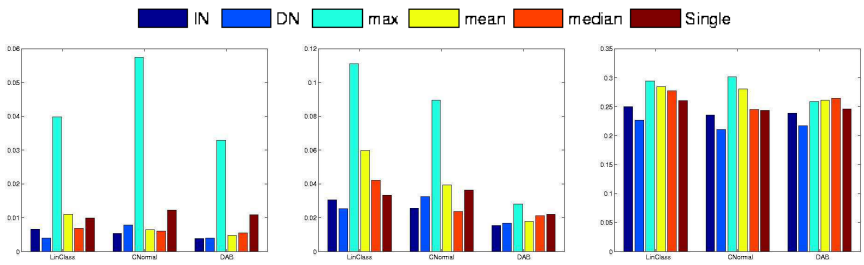
$$W_{pos} = \frac{\#T_{pos}}{N} \qquad\qquad W_{neg} = \frac{\#T_{neg}}{N}$$

We have computed $D^{\mathcal{N}}$ by linearly interpolating functions G and F_W^{-1}.

Binary Classifiers

In this experiment the question to answer is whether the fusion of binary classifiers outperforms the classifier fusion methods used as reference when they are applied

to real data. Besides, we also want to verify if the classification rate using the best descriptor, denoted as *single*, is outperformed by the IN and the DN methods. Hence, we have trained *DAB*, *Linclass* and *CNormal* classifiers and derived the optimal weights for both methods (DN and IN). Then, we have applied these classifiers to the training tests of the three sets of evaluation: *GREC ridgelets*, *GREC descriptors* and *MNIST descriptors* as depicted in figure 5.8



(a) GREC ridgelets (b) GREC descriptors (c) MNIST descriptors

Figure 5.8: Mean of misclassification rates for binary classification for "GREC ridgelets","GREC descriptors" and "MNIST descriptors" using IN, DN, max, mean median and single —from left to right.

Results are quite positive. IN and DN rules outperform the other combination rules even though the hypothesis of independence and normal distribution have not been verified. Except for the "GREC descriptors" test using $CNormal$ classifier. In this case, the *median* rule outperforms the IN and DN methods. If we focus on the performance of IN and DN methods, we can not conclude that one of them generally outperforms the other. Depending on the shape database, classifier and descriptor IN is better than DN and reciprocally.

Concerning the *max* rule, we can observe that the general bad performance of *max* rule pinpointed in the experiments using synthetic data is stressed in binary classifiers. Hence, the problem of the *max* method might be in how we have adapted the *max* rule to our case: $max(Z_l) = Z_{l_0}$, where $l_0 = \arg\max\{\arg\max_l Z_l, -\arg\min_l Z_l\}$. This definition is motivated due to the fact that we must treat both positive and negative outputs of the binary classifiers. Thereby, the classification error of the combination rule may be provoked by one classifier having low recognition rate and big variance. This classifier can reach big values with the incorrect sign (positive when the shape does not belong to the class and reciprocally). Hence, we may think about other definitions of the *max* rule.

Therefore, we can summarize this experiment by saying that IN and DN methods actually outperform other fusion methods in binary classifiers even if the required conditions are not always satisfied. However, we can not draw any conclusion about which of the two methods is generally more suitable to compute the weights of combination. The next experiment (in a multi-class context) will enlighten us in this direction.

Multiclass Classifiers

Experiments using binary classifiers have permitted us to compare IN and DN rules regarding other reference rules without considering the effect of the arg max operator in a multi-class problem. Hence, in this section, we have examined the performance of IN and DN in a multi-class problem. Then, in this experiment we have used DAB and $CNormal$ classifiers, discarding $LinClass$ classifiers because its recognition rates are lower and we have considered that it will not contribute to the discussion with additional relevant information.

On the other hand, we have slightly modified the IN method when it is applied to the DAB classifier. In the original IN method we have approximated classifiers by a Dirac distribution when their variance is 0. However, preliminary results in the multi-class experiment were surprising for some classes of shapes. For instance in the GREC database, we have found that there were many symbols miss-classified as belonging to the class *Architectural A*. Hence, we have examined the estimated weights for the ridgelets descriptors. In figure 5.9 we have depicted the weights corresponding to the twenty descriptors of *GREC ridgelets* test. Thus, the classifier with a small variance (the 13th) reaches a weight higher that .9 whereas the rest of classifiers were weighted with values near 0. It means that the result of combining the ridgelets descriptor was essentially the same as the result of the descriptor having the smallest variance.

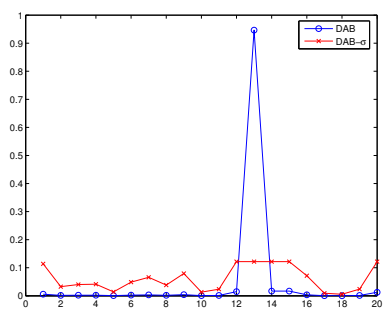


Figure 5.9: Differences between DAB and $DAB-\sigma$ classifiers. Weights corresponding to class *Architectural A* in *GREC ridgelets* test for classifiers DAB and $DAB-\sigma$.

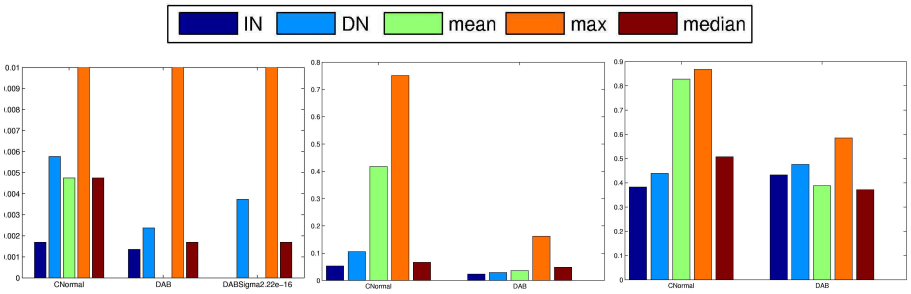
Consequently, we though about approximating classifiers with low variance by a Dirac distribution. In this sense, we looked for an error threshold so that classifiers with an error lower than this threshold would be approximated by a Dirac distribution whereas for higher errors we would approximate the classifiers by a normal distribution. First of all, we have tried an error $\epsilon = 1.10^{-16}$ and then we have computed the variance, σ_0 of the normal distribution (centered in 1), U, satisfying:

$$P(U < 0) = \epsilon$$

being the variance $\sigma_0 \approx 0.0151$. Hence, in the modified version of the classifier,

$DAB{-}\sigma$, we have approximated by a Dirac distribution all classifiers with a variance lower than σ_0 whereas classifiers with higher variances have been approximated by a normal classifier —see figure 5.9 for the new weights obtained using this criteria, where differences between weights are lower than in the DAB classifier .

In figure 5.10 we can see the misclassification rates for $CNormal$, DAB and $DAB-\sigma$ classifiers applied to the *GREC ridgelets* and the misclassification rates for the *GREC descriptors* and *GREC MNIST* test using only $CNormal$ and DAB classifier. We have not applied the $DAB-\sigma$ classifier to these two tests because we have not observed that there were classifiers with a variance lower that 0.0151 (and different of 0) that could modify the misclassification rates.



(a) GREC ridgelets (b) GREC descriptors (c) MNIST-descriptors

Figure 5.10: Mean of misclassification rates for multiclass classification for "GREC ridgelets","GREC descriptors" and "MNIST-descriptors" using IN, DN, max, mean and median rules –from left to right columns.

A first analysis of these results shows that recognition rates decrease for the three set of descriptors, classifiers and fusion strategies regarding recognition rates of binary classifiers. This result can be explained by the fact that even if the classifier of the true class returns a positive value, this value can be lower than the value provided by other classifiers. In particular, the *max* rule achieves again a poor recognition rate in all cases (always lower than 50%). Therefore, we confirm our suspicion and we should look for another definition of the *max* rule suitable for binary classifiers. Besides, unlike binary classifiers, we have noted that the performance of classifiers depend on the test and we will comment the obtained results for each one:

GREC descriptors: We start our experiment discussion with this set of descriptors because the obtained results are the more coherent with the results obtained with binary classifiers. If we compare plots in figures 5.8 and 5.10 we will not realise significant differences in the classifier fusion methods performance. Roughly speaking, IN and DN methods outperform *max*, *mean* and *median* classifiers. Besides, IN is better than DN.

GREC ridgelets: The recognition rates for all classifier fusion methods (except for the *max* rule) is higher than 99,9%. Besides, we can note that the *mean* rule has reached a 0% of misclassification rate using DAB classifier, achieving better

recognition rates than IN and DN method. However, the DAB–σ classifier has also reached a 0% of classification error for both, IN and *mean* rule. However, the DN method has obtained poorer recognition rates for $DAB-\sigma$ than for DAB classifier.

MNIST descriptors: Although the evaluation of the misclassification rates is not the main goal of these experiments, we must say some words about it in this discussion of results. The selected descriptors for the MNIST database provide quite similar description of almost all classes, *i.e.* most of the classes overlap and consequently, they lead to low performance classifiers. In this context, we must note that for $CNormal$ classifier, recognition rates using *mean* and *max* are near to the random choice (whereas for binary classifiers misclassification rates were under 30%). Conversely, IN and DN methods have misclassification rates near to 40% for the $CNormal$ classifier (whereas for binary classifiers they achieve a rate near to 25%). However, in spite of these low recognition rates, these results are still coherent with the results using binary classifiers in the same way that the obtained in the *GREC descriptor* test, *i.e.* the IN and DN methods are better than the other combination rule and besides, the DN method is outperformed by the IN method.

IN	0	1	2	3	6	8
0	**906**	2	0	0	72	0
1	0	**1135**	0	0	0	0
2	925	104	**0**	3	0	0
3	0	1	696	**0**	0	313
6	1	3	116	225	**613**	0
8	10	1	151	0	10	**802**

mean	0	1	2	3	6	8
0	**946**	1	0	0	33	0
1	0	**1135**	0	0	0	0
2	991	22	**0**	19	0	0
3	4	1	683	**0**	0	322
6	0	2	80	57	**819**	0
8	23	1	115	0	8	**827**

Table 5.4: Confusion matrices for IN and *mean* rules.

We will pay attention to the obtained results using DAB classifier. For this classifier the *mean* rule outperforms IN and DN methods, when in binary classifiers the optimal linear rules were the best. If we analyse the confusion matrices for the IN and DN methods —see table 5.4, the confusion matrices— we can distinguish three groups of numerals for IN and *mean* methods. Numeral '1' has a recognition rate over 95%, numerals '2' and '3' have a 0% of recognition, finally numerals '0', '6' and '8' have a recognition rate higher than 80% (but under 95%). Hence, the difference in the recognition rates are achieved for those numerals achieving low recognition rates (numerals '0', '6' and '8'). As a conclusion, we can say that when we use binary classifiers achieving low recognition rates, being not normal distributed and then merged using the $\arg\max$ operator the *mean* rule can perform better than IN or DN rules. Furthermore, we must remark that IN method has achieved a 61,73% recognition rate using the $CNormal$ classifier whereas the *mean* rule has achieved a 61,21% of recognition rate using the DAB classifier. It can mean that the $CNormal$ classifier fits better the distribution of descriptors than the DAB classifier. However, we must carry out more experiments in this direction before drawing more con-

clusions. The only conclusion we are able to state is that it seems that the set formed by ART, F-R and LNR descriptors are suitable for detecting the numeral ' 1' of the MNIST database, whereas for the other numerals other descriptors (and classifiers) should be chosen.

To finalize this experiment, we can conclude saying that in multi-class classifiers the IN method generally performs better than *mean*, *max* and *median* combination rules.

5.5 Combining ridgelets descriptors

The work in classifier fusion was motivated by the problem of combining the ridgelets descriptors. Hence, the last set of experiments consists of applying the classifier fusion methods to the combination of ridgelets descriptors. There are three reasons that justify this experiment, although we have already used ridgelets descriptors in the evaluation of fusion methods —cf. section 5.4:

1. In *GREC ridgelets* test we have not considered rotated shapes because ridgelets descriptors are not invariant to rotation.
2. In *GREC ridgelets* we have used the Euclidean distance instead of the similarity measure defined in (3.15).
3. In *GREC ridgelets*, we have taken the 20 descriptors obtained from the ridgelets transform without considering the division of ridgelets descriptors in decomposition levels —as explained in section 3.4.

In this section, we want to use the similarity measure introduced before in section 3.4.2 and besides, we want to take advantage of the representation of ridgelets descriptors into decomposition levels but using IN and DN methods instead of the heuristic CR method introduced in section 3.4.3. In this way, we will be able to compare the obtained results in section 5.3 to the results obtained in this section using the IN and the DN methods. Nevertheless, we need to slightly modify both linear methods in order to adapt them to the representation in decomposition levels.

Thus, we had to slightly modify the IN and DN methods designing pyramidal versions of them, PIN and PDN respectively, to compare shapes having different number of decomposition levels in ridgelets descriptors —see Algorithms 5.3 and 5.4. Both algorithms return a set of classifiers, namely $C_1, \ldots, C_{DL}$, one for each decomposition level. Afterwards, we have repeated the test of the GREC'03 symbol recognition contest using the ridgelets descriptors combined, but using the PIN and PDN methods in order to compare them, regarding the CR method and the ART descriptor —cf. section 5.3.

For the IN method, the combination rule is defined in a straightforward way. As the classifiers have been considered to be independent, we can compute all the weights

Algorithm: *PIN*

Input: *Oracle*: $\{(D_n, Y_n)|D_n$ is a vector of L descriptors$\}$
DL, maximum decomposition level
Output: $C_1, C_2, \ldots, C_{DL}$, Pyramidal classifier

begin:
Obtain α from the IN method;
for $s = 1, \ldots, DL$,
$S = (s+1)(s+2)$;
Set: $A = \sum_{l \leq S, t=\mathcal{N}} \alpha_l^t$ and $A = \sum_{l \leq S, t=D} \alpha_l^t$;
if $A > B$,
$\lambda_{\mathcal{N}} = \frac{A-B}{2A-B}$ and $\lambda_{\mathcal{N}} = \frac{A}{2A-B}$;
else
$\lambda_{\mathcal{N}} = 0$ and $\lambda_{\mathcal{N}} = 1$;
end;
update:
$\alpha_l^{\mathcal{N}} = \lambda_{\mathcal{N}} \alpha_l^{\mathcal{N}}$;
$\alpha_l^{D} = \lambda_{D} \alpha_l^{D}$;
and normalize α such that: $\sum_l \alpha_l = 1$;
$C_s = \sum_{l<S} \alpha_l h_l$;
endfor
end:

Algorithm 5.3: Pyramidal IN method

together at the beginning of the PIN algorithm. The difficulty lies in computing the weights $\lambda_{\mathcal{N}}$ and λ_D in case we mix both normal and Dirac classifiers. Thus, for each decomposition level from 1 to DL, we have to consider the S first classifiers and aftwewards, apply to them the mix rule defined in the Proposition 3 to obtain the suitable $\lambda_{\mathcal{N}}$ and λ_D —cf. section 4.4. More precisely, the classifier C_1 takes into account the ridgelets descriptors from the decomposition level 0 and 1 (normalized so that the sum of weights is 1). The classifier C_2 is defined as the classifier C_1 but considering descriptors up to the decomposition level 2. Finally, the classifier C_{DL} takes into account all the descriptors up to the decomposition level DL. On the contrary, for DN method the combination rule is defined as an hybrid fusion rule because classifiers are considered to be dependent. Thus, we have to apply the DN method for each classifier C_s. The minimal ridgelets representation consist of the two first decomposition levels (composed of the six first descriptors). Hence, we have applied the DN method to obtain the optimal weights for the classifier C_1. Then, to define the classifier C_2 we have considered the classifiers trained for the descriptors from the decomposition level 2 and we have combined with the classifier C_1 obtained in the previous iteration. The classifiers C_3 is defined as C_2 in a similarly way and we have repeated the process for all decomposition levels:

Algorithm: *PDN*

Input: $\{(X_n, Y_n)|X_n$ is a vector of L descriptors$\}$
Output: H, combination of classifier
begin:
Train L classifiers for the L descriptors, h_l;
for $l = 1 \ldots L$,
Get hypothesis $z_{l,n} = h_l(X_{l,n})$;
Obtain the *validation* values: $u_{l,n} = y_n z_{l,n}$;
endfor;
Set $C_0 = 0$;
for $s = 1, \ldots, DL$,
$S_0 = s(s+1)$;
$S = (s+1)(s+2)$;
Obtain the *validation* values: $\tilde{u}_n = y_n C_{s-1}$;
Set $U_s = (U_{S_0}, \ldots, U_S)$ and $\tilde{U} = (\tilde{u}_n, U_s)$;
Obtain weights: $\alpha = DependentWeight(\tilde{U})$;
$C_s = \alpha_0 C_{s-1} + \sum_{l=S_0}^{S} \alpha_l h_l$;
endfor
end:

Algorithm 5.4: Pyramidal DN method

$$C_{s+1} = \alpha_0 C_s + \sum_{l=S_0}^{S} \alpha_l h_l$$

As we have computed the ridgelets descriptors from 128×128 images, we have only four decomposition levels and therefore, the PIN and PDN methods will return three classifiers: C_1, C_2 and C_3.

5.5.1 Invariance to similarity transforms

For this experiment we have applied the tests: *rotation*, *scale* and *rotation*and *scale* defined for the *set 3* in the GREC'03 contest (composed of 50 different symbols). We can see the results in table 5.5, where we have written in bold letter the results obtained using the CR method:

A quick analysis of the obtained results permits us to verify that the ridgelets descriptors are invariant to the change of scale for all classifiers and pyramidal combination rules. Only the C_1 classifier does not reach the 100% of recognition rate. However, when we introduce rotated symbols, the performance of the pyramidal methods decreases. Only the PIN method applied to the DAB classifier reaches the 100% of recognition rate (as the CR method) for all the decomposition levels. The other

	C_1	C_2	C_3	C_1	C_2	C_3
	Scale: **100,00%**			*Rotation:* **100,00%**		
PIN-DAB	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
PDN-DAB	99,00%	100,00%	100,00%	94,00%	94,00%	99,00%
PIN-CNormal	99,00%	100,00%	100,00%	92,00%	92,00%	99,00%
PDN-CNormal	100,00%	100,00%	100,00%	96,00%	94,00%	96,00%
	C_1	C_2	C_3			
	Rotation & Scale: **98,80%**					
PIN-DAB	100,00%	100,00%	100,00%			
PDN-DAB	96,00%	96,00%	94,00%			
PIN-CNormal	92,00%	95,00%	94,00%			
PDN-CNormal	94,00%	93,00%	90,00%			

Table 5.5: Invariance to similarities. The rates in bold letter correspond to the recognition rates using the CR method

methods have a recognition rate lower than 96% for classifiers C_1 and C_2. Finally, we have tested the pyramidal classifiers in rotated and scaled images. In this case, the $CNormal$ classifier and the DAB classifiers combined using the PDN method are outperformed by the CR method, that has achieved 98,80% of recognition rate. However, the DAB classifier combined using the PIN method has reached the 100% of recognition rate even for the C_1 classifier.

Therefore, we can draw three conclusions: First, in general the PIN and PDN methods do not outperform the CR method when we apply them for any classifier. Second, the DAB classifier combined using the PIN method is actually invariant to similarity transforms even using the two first decomposition levels (classifier C_1). Finally, the PIN method outperforms the PDN when all classifiers are considered.

5.5.2 Robustness to degradation and vectorial distortion

Then, we have tested the robustness to degradation and vectorial distortion of the PIN method using the DAB and $CNormal$ classifiers. We have discarded the PDN method because of the low recognition rates achieved in the previous experiment with rotated symbols. We recall that, in the GREC'03 database there are some tests containing images with several kinds of binary degradations and vectorial distortions. These images have been generated using models of image degradation and deformation trying to simulate degradations found in real images. There are nine different models of binary degradation enumerated from *model 1* to *model 9* and three degrees of vectorial distortion. The nine models of degradations try to simulate those binary degradations introduced by copying, printing or scanning documents [Kanungo et al., 1994]. They are based on a statistical model to add binary noise to images. On the other hand, vectorial distortions try to model shape variability introduced by hand-drawing. It is based on a statistical model to generate variations of the lines in the symbol [Valveny, 1999]. The generation of vectorial distortion has been limited, in

the symbol database, only to 15 symbols exclusively composed of straight lines, and each test contains 75 images. Nevertheless, we have used the models of all symbols when classifying these images. In table 5.6 we can see the results of applying the PIN method using DAB and $CNormal$ classifiers to *degraded* and *deform-degrad-level3* tests, and comparing them to the results using the ART descriptor and the CR method introduced at the beginning of this chapter.

			C_1		C_2		C_3	
	CR	ART	DAB	CNormal	DAB	CNormal	DAB	CNormal
(a) degraded								
model 1	100,00	100,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 2	100,00	100,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 3	100,00	86,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 4	99,60	85,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 5	100,00	88,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 6	100,00	100,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 7	100,00	100,00	100,00%	99,00%	100,00%	98,00 %	100,00%	97,00%
model 8	98,40	89,00	98,00%	98,00 %	100,00%	100,00%	100,00%	100,00%
model 9	89,20	84,00	95,00%	93,00 %	99,00%	97,00 %	100,00%	100,00%
			C_1		C_2		C_3	
	CR	ART	DAB	CNormal	DAB	CNormal	DAB	CNormal
(b) deform-degrad-level3								
model 1	98,67	100,00	99,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 2	97,33	99,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 3	98,67	88,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 4	98,67	88,00	99,00%	100,00%	99,00%	100,00%	100,00%	100,00%
model 5	97,33	89,00	99,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 6	97,33	100,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 7	100,00	100,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 8	100,00	100,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
model 9	100,00	99,00	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%

Table 5.6: Test with degradation and vectorial distortions tests using the PIN method with DAB and $CNormal$ classifiers

Concerning degraded symbols, the ART descriptor achieves a poor recognition rate for the degradation models: *model 3, model 4, model 5, model 8* and *model 9*, with a recognition rate lower than 90%. Besides, the CR method had a poor performance in symbols with the degradation model 9. However, the PIN method achieves recognition rates higher than 90% with the C_1 classifier, which only uses the six first descriptors with both DAB and $CNormal$ classifier. Besides, the DAB classifier achieves the 100% of recognition rate for the nine models of degradation for classifier C_3, using all the decomposition levels.

Besides, for degraded and vectorial distortions of symbols we have achieved a similar performance. The CR rule and ART descriptor degrade performance in most of tests whereas the PIN method achieves a 100% of recognition rate for the nine

models of degradation in both classifiers, DAB and $CNormal$ —except for some few tests using the DAB classifier for C_1 classifier.

5.5.3 Discussion

The previous experiment validates the approach of applying fusion methods to combine the description obtained using multiresolution descriptors. If we observe the obtained weights for all classes, we can easily realize that not all the resolutions have the same relevance for all classes. It means that not all resolutions properly describe all symbols in the same way. Thereby, depending on the class considered some resolutions will be better than others and this fact will be reflected in the estimated weights. On the contrary, in the definition of the CR rule, we have discarded a set of resolutions according empirical evidences —as explained in section 3.4.3— and besides, we have assigned the same weight to all resolutions belonging to the same decomposition level.

In this last discussion, we have tackled a last question: Which is the relationship between the optimal weights using PIN and PDN method and the ridgelets descriptors selected in the CR method? we have plotted two surfaces representing the weights of resolutions using CR and PIN method —in figure 5.11. For the PIN method, we have taken into account all the optimal weights computed for the 50 symbols of the GREC'03 database and we have computed the mean of these weights —see right side of figure 5.11. Similarly, for the CR method we have assigned a weight 0 for the descriptors we have not considered. Then, we have assigned a weight to the rest of descriptors of the same decomposition level inversely proportional to the number of descriptor —see the definition of the similarity measure in Eq. (3.4.2)— *i.e.* at level 0, we have one descriptor and hence, we have assigned a weight of 1. In the second level, we have two descriptors and the assigned weight has been $\frac{1}{2}$. For the third level, the assigned weight is $\frac{1}{3}$ and finally, for the fourth decomposition level the weight is $\frac{1}{4}$.

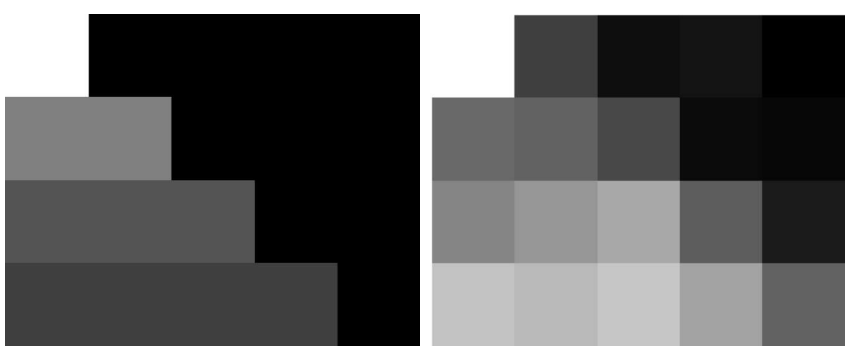


Figure 5.11: weights of CR and IN methods

Each square of these surfaces represent a descriptor —following the same order than in section 3.4— the dark gray and black colors represent the lowest weights

values whereas the light gray and white colors represent the highest values. We can observe that the lowest weights using the PIN method correspond to the descriptors with poorer recognition rates that have not been considered in the CR method (we recall that these descriptors correspond to the "vertical" descriptors in the decomposition level). Besides, the first descriptor has achieved the highest weight in both, CR and PIN, method. On the contrary, descriptors from the "horizontal" part of higher decomposition levels have achieved higher weights than descriptors from decomposition levels 1 and 2. If we take into account the results depicted in table 5.6, both facts may mean that the discriminant information is found in the higher resolutions.

Chapter 6

Conclusions

In this chapter we have summarized the contributions of this dissertation to the field of shape description and classifier fusion. Besides, we have presented the main conclusions of the research done in ridgelets descriptors and linear combination rules. We have finalized by pointing some possible lines of continuity to improve the performance of the proposed methods.

6.1 Summary of the Contributions

In this dissertation we have tackled different issues related to the shape recognition process. In the chapter 1 we have distinguished three different conceptual stages involved in this process: *shape description*, *shape comparison* and *shape classification*. In this context, the forthcomings chapters have been focused on different aspects of these stages, leading to the following contributions:

- In chapter 2, we have reviewed several shape descriptors proposing the definitions of **descriptor**, **primitive** and **feature extraction method** in order to homogenize and simplify the terminology used to describe shape descriptors. Besides, we have introduced the notion of **primary** feature extraction method in order to decompose in a flexible manner a descriptor in more elemental entities.

- In chapter 3, we have introduced a **ridgelets** descriptor to be used for representing symbols, and more in general, to represent shapes where the "straight line" is a relevant feature. This descriptor is based on a multiresolution transform which has permitted us to group ridgelets coefficients into several decomposition levels according to the original shape resolution. Besides, in the comparison stage we have defined a rotation invariant similarity measure which has been used, later, in a heuristic combination rule proposed in the classification stage.

- In chapter 4, we have proposed two **linear combination rules**, IN and DN. These classifier fusion methods have been proposed in a probabilistic framework based on the definition of suitable r.v. that has permitted us to prove in a rigorous way that IN and DN **minimize the classification error** of classifiers verifying a set of constraints.

Then, in the chapter 5 we have used well-known shape databases as a reference for comparing the ridgelets descriptor and IN and DN methods regarding similar approaches (the ART descriptor and the *mean*, *max* and *median* rules, respectively). Besides, concerning the proposed combination rules we have carried out some experiments with real and synthetic data. The experiments using synthetic data have been useful to evaluate the IN and DN performance depending on whether the data satisfies, or not, the constraints theoretically stated. The experiments using real data have been suitable to state the performance of these methods in a more realistic framework.

In the next section we have summarized the main conclusions concerning ridgelets descriptors and IN and DN methods.

6.2 Discussion and Conclusions

When we have reviewed the bibliography in shape descriptors, we have realized that there was not an unified terminology to denote the different elements taking part in the shape description process. In this context, we have formalized and unified the concept of feature vectors, signatures and other terms used to refer to the results of feature extraction methods under the name of *descriptor* and generalizing the notion of *primitive*, in such way a that it can be applied to any descriptor. Finally, we have introduced the notion of *primary* feature extraction method in order to decompose in a flexible manner a descriptor in more elemental entities.

In this way, we have linked some properties of descriptors to one of these three elements taking part in the process of shape description. We have characterized descriptors according to the geometry of their primitives —distinguishing between 1D and 2D descriptors. Besides, we have connected structural and multiresolution properties to the descriptor itself and, finally, the invariance of descriptors to affine transforms and the information preserving property has been linked to the feature extraction methods. Thereby, we have proposed a descriptor that, thanks to this categorization of shape descriptors, we can describe as 2D, multiresolution and polar descriptor based on the ridgelets transform.

6.2.1 Ridgelets descriptors

Ridgletets descriptors are based on a multiresolution transform that can be applied to the analysis of general images. It is built on a solid mathematical background which permit us to define a multiresolution and homogeneous representation. Ridgelets

transform is specifically defined to localize —using polar coordinates— linear singularities or, in general, tangents to C^1 curves. This characteristic allows us to apply our method to any kind of regular contours. However, it will work better applied to images in which the linear component is a relevant feature.

We have carried out a set of experiments to validate the symbol representation proposed in section 3.4. If we compare these results with those obtained using the ART descriptor and those obtained in the contest in Graphic Recognition held during the Workshop GREC'03 [Valveny and Dosch, 2004], then we can state that ridgelets coefficients are an excellent set of descriptors for degraded graphics; robust to vectorial distortion and scaling of images. Moreover, the similarity measure defined in expression (3.15) is actually invariant to rotation.

However, some drawbacks appear in the use of this type of descriptor. On the one hand, an image of 128×128 generate 32,768 ridgelets coefficients which is a huge size for a descriptor since the usual size turns around 16, 30 or 64 features. It is not only a problem of stocking descriptors or a problem in the required time for computing similarities between descriptors. Besides, it is also a problem for indexing and retrieval purposes since high dimensional descriptors drop away the performance of indexing methods [Berrani, 2004, Ciaccia et al., 1997, Manolopoulos et al., 2003]. On the other hand, the estimation of the coordinates origin used to computed the change to polar coordinates is another drawback. A bad estimation of the center of coordinates can lead to different representations of the shape. In fact, this is a inherent drawback of *polar* descriptors in general and hence, it affects to ridgelets descriptor as well (cf. section 2.3.1).

Therefore, we have partially overcome the drawback of the size of descriptor by introducing the *Local norm* descriptors based on ridgelets transform and combining them with other shape descriptors as the *ART* or the $\mathcal{R}$-Signature. Besides, the multiresolution representation and, more specifically the decomposition level, has helped us to reduce the size of descriptors and attenuates the variability in determining the center of the image.

6.2.2 Classifier Fusion

We have proposed a rigorous probabilistic approach consisting of the definition of several r.v. such as the *validation* r.v. or the *prediction* r.v. Indeed, we have identified the classifier as the *prediction* r.v. Then, it has sense to talk about the independence, or the dependence, of classifiers, as well as the classifier distribution since we are actually talking about the *prediction* r.v. In this way, we have found two optimal combination rules for normal classifiers (being either independent or dependent):

IN: The IN method assumes that classifiers are independent and normal. The experimental results —cf. section 5.4— illustrates that this combination rule outperforms other similar combination rules. Besides, the optimal weights are very easy to compute as they only depend on the mean and variance of classifiers. However, it would be interesting to find out optimal solutions for other distri-

butions such as a mixture of normal or Bernoulli distributions since classifiers can be better modeled by these kind of r.v.

DN: The DN method assumes that classifiers are dependent and besides, that the linear combination of classifiers is also a normal distribution. Experimental results —cf. section 5.4— show that this method does not take any advantage of the dependence of classifiers and it has a poorer performance than the IN. Besides, in multi-class problems other fusion methods like *mean* perform better than the DN method. Therefore, the condition of normal distribution for the linear combination is too restrictive and we should find weaker conditions for the joint distribution of classifiers.

Finally, the last experiment has validated the approach consisting of the combination of ridgelets descriptors using the linear combination rules. Besides, the obtained description composed of shape descriptors, but also of the weights computed from the considered classifier depends on each class. It means that not all ridgelets resolutions properly describe all symbols in the same way. Depending on the class considered some resolutions are better for graphics description than others. This fact is reflected in the estimated weight: the best resolution achieves the highest weights. In other words, we can have a set of descriptors and classifiers and only use those that properly describe each class of shape discarding the others for description purposes.

6.3 Open issues

In the previous section we have seen that each of the aspects treated in this dissertation has left several open issues. Concerning shape descriptors, we have seen that we can not effectively reduce the size of ridgelets descriptors without dropping the recognition rates. Concerning classifier fusion, we have experimentally verified that IN method outperforms other related methods but without being able to determine a set of conditions on classifiers permit to assure that linear methods outperform other classifier fusion methods. Besides, DN method has not proved to outperform IN method when classifiers are dependent.

In this section, we propose plausible lines of continuity of the work introduced in this dissertation from the viewpoint of the three stages defined in the shape recognition process:

Shape description: Multiresolution transforms are suitable transforms to organize the information according to the decomposition levels. However, ridgelets transform, as it has been defined, is not suitable to be applied to whole documents. On the one hand, there is the problem of the size of descriptor. On the other hand, there is the problem associated to the change of Cartesian to polar coordinates. Given a specific application, the size of descriptor can be reduced using existing methods for reducing the size of descriptors such Principal Component Analysis or Linear Discriminant Analysis methods, to name some examples.

Concerning the use of polar coordinates, when we move away from the origin of coordinates the representation in polar coordinates become poorer. Therefore, other multiresolution transforms have to be investigated. For instance, *curvelets* and *contoulets* transforms are possible transforms to be taken into account [Candès and Demanet, 2002, Candès and Donoho, 2003a,b].

Shape Comparison: An open issue since the beginnings of the shape recognition problem is shape comparison. This is a well-known problem in the structural pattern recognition where Conte et al. [2004] review structural approaches for shape recognition from the point of view of the matching algorithms used. However, this is a problem shared by all shape descriptors. More specifically, the problem is how to deal with huge amount of information. In this sense, Bartolini et al. [2005] propose a descriptor based on the Fourier transform that will be stored in a M-Tree structure [Ciaccia and Patella, 2002]. The M-Tree which is an evolution of the R-Tree [Manolopoulos et al., 2003] for metric space organizes descriptors in a tree structure according to the metric defined in the space of descriptors. The architecture of this structure follows similar rules than the B-Trees, which look for balanced trees in order to reduce the mean of the number of comparisons. However, this structure only takes into account one type of descriptor. Thus, for a shape described using several descriptors, we must construct an indexing structure for each descriptor. Therefore, we have to look for indexing structures, compatible with a multiresolution description of documents and taking into account several types of descriptors.

Shape Classification: We have based our proposal for classifier fusion on training univariate classifiers based on the distribution of distance between the query shape and the model shape. The probabilistic framework introduced for classifier fusion, in chapter 4, can be easily extended to multivariate classifiers in a straightforward way. Each feature of the descriptor can be considered as a r.v. and hence, the problem of the combination of L dependent classifiers can be seen as the problem of determining the border of two classes minimizing the misclassification rates in a multivariate random vector. Therefore, we can investigate the relationship between this probabilistic approach an other theoretical frameworks used for developing classifiers as support vector machines or boosting classifiers.

6.4 Final Conclusion

In this dissertation we have touched different aspects of the shape recognition process. From a wide viewpoint, we have studied multiresolution descriptors, grouping them into several decomposition levels that have been combined using optimal linear classifier fusion methods. These theoretical approaches, have been applied in particular, to the combination of ridgelets descriptors using the IN method for the recognition of graphics symbols. The recognition rates achieved in the GREC'03 database are near to the 100% in almost all tests.

Bibliography

S. Adam, J. M. Ogier, C. Cariou, R. Mullot, J. Gardes, and Y. Lecourtier. Using the fourier mellin transform for multi-oriented and multi-scaled patterns recognition: application to automatic analysis of technical documents. *Traitement du signal*, 18: 17–33, 2001.

F. M. Alkoot and J. Kittler. Experimental evaluation of expert fusion strategies. *Pattern Recognition Letters*, 20:1361–1369, 1999.

E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1: 113–141, 2000.

H. Asada and M. Brady. The curvature primal sketch. In *MIT AI Memo*, number 758. MIT Press, 1984.

P. Auscher, G. Weis, and M. V. Wickerhauser. Local sine and cosine bases of coifman and meyer and the construction of smooth wavelets. *Wavelets: A tutorial in theory and Applications*, pages 237–256, 1991.

A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, and J. Waldén. Fast slant stack: A notion of radon transform for data in a cartesian grid which is rapidly computible, algebraically exact, geometrically faithful and invertible. Technical report, Standford University, 2001.

D. H. Bailey and P. N. Swarztrauber. The fractional fourier transform and aplicattions. *SIAM Review*, 33(3):389–404, September 1991.

D. H. Ballard. Generalising hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, February 1981.

L. Bartolini, P. Ciaccia, and M. Patella. Warp: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *IEEE Transactions on PAMI*, 27(1):142–147, January 2005.

S.-A. Berrani. *Recherche approximative de plus proches voisins avec contrôle probabiliste de la précision; application à la recherche d'images par le contenu.* PhD thesis, Université de Rennes I, 2004.

S. Berreti, A. del Bimbo, and P. Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Transactions on Multimedia*, 2(4):225–239, December 2000.

H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, 1967.

L. Breiman. Bias, variance and arcing classifiers. Technical Report 460, Departement of Statistics, University of California, 1996.

H. Bunke. Attributed programmed graph grammars and their application to schematic diagram interpretation. *IEEE Transactions on PAMI*, 4:574–582, 1982.

H. Bunke. Hybrid pattern recognition methods. In H. Bunke and Sanfeliu A., editors, *Syntactic and Structural Pattern Recognition. Theory and applications*, pages 307–347. World Scientific publishing Company, 1990.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Dataming and knowledge discovery*, 2(2):1–43, 1998.

E. J. Candès. *Ridgelets: Theory and Applications*. Phd, Standford University, September 1998.

E. J. Candès and L. Demanet. Curvelets and fourier operators. *Comptes-Rendus de l'Academie des Sciences*, pages 395–398, 2002.

E. J. Candès and D. Donoho. Continuous curvelet transform: I. resolution of the wavefront set. Technical report, Statistics Department, Standford University, 2003a.

E. J. Candès and D. Donoho. Continuous curvelet transform: Ii. discretization and frames. Technical report, Statistics Department, Standford University, 2003b.

E. J. Candès and D. L. Donoho. Ridgelets: a key to higher-dimensional intermittency? *Phil. Trans. R. Soc. Lond. A*, 357:2495–2509, 1999.

J. F. Canny. A computational approach to edge detection. *IEEE Transactions on PAMI*, 8:679–698, 1986.

Y. Chen, N. A. Langrana, and A. K. Das. Perfecting vectorized mechanical drawings. *Computer Vision and Image Understanding*, 63(2):273–286, March 1996.

C.-W. Chong, P. Raveendran, and R. Mukundan. A comparative analysis of algorithms for fast computation of zernike moments. *Pattern Recognition*, 36:731–742, 2003.

P. Ciaccia and M. Patella. Searching in metric spaces with user-defined and approximate distances. *ACM transactions on Database Systems*, 27:398–437, 2002.

P. Ciaccia, M Patella, and P. Zezula. M-tree: An efficient acces method for similarity search in metric spaces. In *Proceedings of the 23rd VLDB Conference*, 1997.

D Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal on Pattern Recognition and Artificial Intelligence*, 18(3):265–298, 2004.

L. da F. R. Costa and R. M. Cesar Jr. *Shape analysis and classification.* CRC Press, 2001.

M. Das, M. J. Paulik, and N. K. Loh. A bivariate autoregressive modeling technique for analysis and classification of planar shapes. *IEEE Transactions on PAMI*, 12 (1):97–103, January 1990.

S. R. Deans. Hough transform from the radon transform. *IEEE Transactions on PAMI*, 3(2):185–188, March 1982.

D. L. Donoho. Orthonormal ridgelets and linear singularities. *SIAM Journal*, 31(5): 1062–1099, 2000.

R. 0. Duda, P. E. Hart, and D. G. Stork, editors. *Pattern Classification.* Wiley Interscience, 2000.

M. El Rube, I. Ahmed, and Kamel M. Wavelet approximation-based affine invariant shape representation fucntions. *IEEE Transactions on PAMI*, 28(2):323–327, February 2006.

H. Fahmy and Blonstein D. A graph grammar programming style for recognition of music notation. *Machine Vision and Applications*, 6:83–99, 1993.

A. G. Flesia, H. Hel-Or, A. Averbuch, E. J. Candés, R. R. Coifman, and D. L. Donoho. Digital implementation of ridgelet packets. In *Beyond Wavelets*, volume 10, chapter 2, pages 31–60. AP, 2001.

D. Forsyth, J. L. Mundy, A. Zisseman, C. Coelho, A. Heller, and C. Rothwell. Invariant descriptors for 3d object recognition and pose. *IEEE Transactions on PAMI*, 13(10):971–991, 1991.

P. Fränti, A. Mednonogov, V. Kyrki, and H. Kalviainen. Content-based matching of line-drawing images using the hough transform. *IJDAR*, 3:117–124, 2000.

Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Thirteenth International conference on machine learning*, 1996.

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Dept. of Statistics, Standford University, 1998.

F. Groen, A. Sanderson, and F. Schalag. Symbol recognition in electrical diagrams using probabilistic graph matching. *Pattern Recognition Letters*, 3:343–350, 1985.

A. Habacha. Structural recognition of disturbed symbols using discrete relaxation. In *1st. International Conference of Document Analysis and Recognition*, pages 170–178, Saint Malo, France, 1991.

T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifiers systems. *IEEE Transactions on PAMI*, 16(1):66–75, 1994.

M.-K. Hu. Visual pattern recognition by moments invariants. *IRE Transactions on Information Theory*, 8:179–187, 1962.

A. K Jain, R. P. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on PAMI*, 22(1):4–37, January 2000.

G. M. James. Variance and bias for general loss functions. *Machine Learning*, 2: 115–135, 2003.

X. Jiang, A. Munger, and H Bunke. Synthesis of representative graphical symbols by computing generalized median graph. In A. Chabra and Dori D., editors, *Graphics Recognition: Recent Advances*, volume 1941 of *LNCS*, pages 183–192. Springer-Verlag, 2000.

A. Juan and E. Vidal. On the use of bernoulli mixture models for text classification. *Pattern Recognition*, 35:2705–2710, 2002.

T. Kanungo, Haralick R., Baird H., W. Stuetzle, and Madigan D. Document degradation models: Parameter estimation and model validation. In *International Workshop on Machine Vision Applications*, December 1994.

R. Kashyap and R Chellappa. Stochastic models for closed boundary analysis: Representation and reconstruction. *IEEE Transaction on Information Theory*, 27(5): 627–637, September 1981.

M. I. Khalil and M. M. Bayoumi. Invariant 2d object recognition using the wavelet modulus maxima. *Pattern Recognition Letters*, 21:863–872, 2000.

M. I. Khalil and M. M. Bayoumi. A dyadic wavelet affine invariant function for 2d shape recognition. *IEEE Transactions on PAMI*, 23(10):863–872, October 2001.

M. I. Khalil and M. M. Bayoumi. Affine invariants for object recognition using wavelet transform. *Pattern Recognition Letters*, 23:57–72, 2002.

W.-Y. Kim, Y.-S. Kim, and Kim Y. S. A new region-based shape descriptor. Technical report, Hanyang University and Konan Technology, December 1999.

J. Kittler. A framework for classifier fusion: Is it still needed? In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 45–56. Springer-Verlang, 2000.

J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on PAMI*, 20(3):226–239, 1998.

L. I. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on PAMI*, 24:281–286, 2002.

L. I. Kuncheva, C. J. Whitaker, C. A. Shipp, and R. P. W. Duin. Is independence good for combining classifiers? In *15th International Conference on Pattern Recognition, 2000*, volume 2, pages 168–171, 2000.

V. F. Leavers. Use of the radon transform as a method of extracting information about shape in two dimensions. *Image Vision Computing*, 10(2):99–107, march 1992.

S. Lee. Recognizing hand-written electrical circuit symbols with attributed graph matching. In H. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Analysis*, pages 340–358. Springer Verlag, Berlin, 1992.

P. G. Lemarie and Y. Meyer. Ondelettes et bases hilbertiennes. *Revista Matemática Iberoamericana*, 2(1,2):1–18, 1986.

R. P. Lippmann. An introduction to computing with neural nets. *SIGARCH Computer Architecture News*, 16(1):7–25, March 1988.

J. Lladós. *Combining Graph Matching and Hough Transform for Hand-Drawn Graphical Document Analysis. Application to Architectural Drawings.* Phd, Universitat Autònoma de Barcelona, September 1997.

S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on PAMI*, 11(7):674–693, July 1989.

S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 1999.

B. Manjunath, P. Salembier, and T. Sikora, editors. *Introduction to MPEG-7. Multimedia Content Description Interface.* John Wiley and Sons, 2002.

B. S. Manjunath, P. Wu, S. Newsam, and H. D. Shin. A texture descriptor for browsing and similarity retrieval. *Signal Processing: Image Communication*, 16 (1-2):33–43, 2000.

Y Manolopoulos, A Nanopoulos, A. N. Papadopoulos, and Y. Theodoridis. R-tree have grown everywhere. *ACM Computing Surveys*, 5:1–07, 2003.

B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee. Shape measures for content based image retrieval: a comparison. *Information Processing and Management*, 33(3): 319–337, 1997.

B. Messner and H. Bunke. Automatic learning and recognition of graphical symbols in engineering drawings. In R. Kasturi and K. Tombre, editors, *Graphics Recognition: Methods and Applications*, volume 1072 of *LNCS*, pages 123–134. Springer, Berlin, 1996.

J. L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision.* The MIT Press, Massachusetts Institute of Technology, Cambrige, Massachusetts 02142, 1992.

V. Nagasamy and N. A. Langrana. Engineering drawing processing and vectorisation system. *Computer Vision, Graphics and Image Processing*, 49:379–397, 1990.

T. Pavlidis. Survey: A review of algorithms for shape analysis. *Computer Graphics and Image Processing*, 7(7):243–258, 1978.

O. Ramos Terrades and E. Valveny. Indexing technical symbols using ridgelets transform. In J. Lladós and Kwon Y.B., editors, *Graphics Recognition: Recent Advances and Perspectives.*, volume 3088 of *LNCS*, pages 177–187. Springer-Verlang, 2004.

A. Rosenfeld and A. C. Kak. *Digital Picture Processing.* Academic Press, San Diego, California, 1982.

Y. Rui and T. S. Huang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10: 39–62, 1999.

G. Sánchez. *Un modelo sintáctico para la representación y reconocimiento de símbolos texturados en documentos gráficos.* Phd, Universitat Autònoma de Barcelona, November 2001.

R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, march 1999.

I Sekita, T Kurita, and N. Otsu. Complex autoregressive model for shape recognition. *IEEE Transactions on PAMI*, 14(4):489–496, April 1992.

K. Siddiqi, A. Shokoufandeh, Dickinson S. J., and S. W. Zucker. Shock graphs and shape matching. *International Journal on Computer Vision*, 35(1):13–22, 1999.

M. Skurichina and R. P. W. Duin. Bagging, boosting and the random subspace method for linear classifiers. *PAA*, 5(2), 2002.

S. Skyun. A simple algorithm for computing the smallest enclosing circle. *Information Processing Letters*, 37:121–125, 1991.

A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on PAMI*, 22(12): 1349–1380, December 2000.

Z. Stejic, Y. Takama, and K. Hirota. Mathematical aggregation operators in image retrieval: effect on retrieval performance and role in relevance feedback. *Signal Processing*, 85(2):297–324, 2005.

S. Tabbone. *Quelques contribution à la reconaissance de formes dans des documents graphiques.* Hdr, Université Nancy 2,, December 2005.

S. Tabbone and L. Wendling. Technical symbols recognition using the two-dimensional radon transform. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 200–203, August 2002. Montreal, Canada.

S. Tabbone, L. Wendling, and J. P. Salmon. A new shape descriptor defined on the radon transform. *Computer Vision and Image Understanding*, 102(1):42–51, April 2006.

S. Tabbone, L. Wendling, and K. Tombre. Indexing of technical line drawings based on f-signatures. In *International Conference of Document Analysis and Recognition*, September 2001.

D. M. J. Tax, M. van Breukelen, R. P. W. Duin, and J. Kittler. Combining multiple classifiers by averaging or by multiplying. *Pattern Recognition*, 33:1475–1485, 2000.

C.-H. Teh and R. T. Chin. On image analysis by methods of moments. *IEEE Transactions on PAMI*, 10(4):496–513, July 1988.

Q. M. Tieng and W. W. Boles. An application of wavelet-based affine-invariant representation. *Pattern Recognition Letters*, 16:1287–1296, 1995.

Q. M. Tieng and W. W. Boles. Wavelet-based affine invariant representation: A tool for recognizing planar objects in 3d space. *IEEE Transactions on PAMI*, 19(8): 846–857, August 1997.

K. Tombre, C. Ah-Soon, P. Dosch, G. Masini, and S. Tabonne. Stable and robust vectorization: How to make the right choices. In A.K. Chhabra and D. Dori, editors, *Graphics Recognition: Recent Advances*, pages 3–18. Springer-Verlag, Berlin, 2000. Vol. 1941 of Lecture Notes in Computer Science.

I. D. Trier, A. K. Jain, and T. Taxt. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29(4):641–662, 1996.

K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):314–348, 1996a.

K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3-4):385–403, 1996b.

E. Valveny. *Reconocimiento de Símbolos Gráficos a Mano Alzada Mediante Modelos Deformables en un Entorno Bayesiano*. Phd, Universitat Autònoma de Barcelona, September 1999.

E. Valveny and P. Dosch. Symbol recognition contest: A synthesis. In Josep Lladós and Y.B. Kwon, editors, *Graphics Recognition: Recent Advances and Perspectives*, volume 3088 of *Lecture Notes in Computer Science*, chapter Symbol Recognition Contest: A Synthesis, pages 368–386. Springer-Verlang, 2004.

L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transaction on Systems, Man and Cybernetics*, 22(3):418–435, 1992.

P.-T. Yap and R. Paramesran. An efficient method for the computation of legendre moments. *IEEE Transactions on PAMI*, 27(12), December 2005.

C. T. Zahn and R. Z. Roskies. Fourier descriptors for plane closed curves. *Transactions on Computers, IEEE*, c-21(3):269–281, March 1972.

D. Zhang. *Image Retrieval Based on Shape*. Phd, Monash University, March 2002.

D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004.

H. K. Zouri. *Contribution à l'évaluation des méthodes de combinaison parallèle de classifieurs par simulation*. Phd, Université de Rouen, December 2004.

Publications

This dissertation have leaded to the following communications:

Publications in peer-reviewed journals and book's chapters

- O. Ramos Terrades and E. Valveny *A New Use of The Ridgelets Transform for Describing Linear Singularities in Images* Pattern Recognition Letters, Vol 27(6) pages 587 – 596, 2006.
- O. Ramos Terrades and E. Valveny *Indexing Technical Symbols Using Ridgelets Transform.* In J. Lladós, Y.B. Kwon, editors, *Graphics Recognition: Recent Advances and Perspectives* Volume 3088 of Lecture Notes in Computer Science, pages 177 – 187, 2004.
- O. Ramos Terrades and E. Valveny *Line Detection Usign Ridgelets Transform for Graphics Symbol Representation.* In F.J. Perales, A. J.C. Campilho, N. Pérez de la Blanca and A. Sanfeliu, editors, *1rst Iberian Congres on Pattern Recognition and Image Analysis* Volume 2652 of Lecture Notes in Computer Science, pages 829 – 837 , 2004.

Technical Report

- O. Ramos Terrades *Descripció i classificació de símbols tècnics usant la transformada de crestetes*, Computer Vision Center, Universitat Autonoma de Barcelona, CVC-TR74, September, 2003 (in catalan)

Contributions to Conferences and Workshops

- O. Ramos Terrades, S. Tabbone and E. Valveny *Combination of shape descriptors using an adaption of boosting* International Conference on Pattern Recognition, Hong-Kong, 2006.
- O. Ramos Terrades and E. Valveny *Local Norm Features Based on Ridgelets Transform* 8th International Conference on Document Analysis and Recognition, Seoul, 2005.
- O. Ramos Terrades, S. Tabbone , L. Wendling and E. Valveny *Symbol Recognition based on Multiresolution Analysis of the Radon Transform* 1rst International Workshop on Multidisciplinary Image, Video and Audio Retrieval and Mining, Sherbrooke, Canada, 2004.

- O. Ramos Terrades and E. Valveny *Radon Transform for Linear Symbol Representation* 7th International Conference on Document Analysis and Recognition, Edimburg, 2003.
- O. Ramos Terrades and E. Valveny *Indexing Technical Symbols Using Ridgelets Transform* 5th Workshop on Graphics Recognition, Barcelona, 2003.
- O. Ramos Terrades and E. Valveny *Line Detection Using Using Ridgelets Transform for Graphics Symbol Representation* 1rst Iberian Congres on Pattern Recognition and Image Analysis, Mallorca, 2003.

Pre-prints in journals

- O. Ramos Terrades, E. Valveny and S. Tabbone *Optimal Classifier Fusion in a non-Bayesian framework*
- O. Ramos Terrades, S. Tabbone and E. Valveny *A new way for describing descriptors: A survey.*
- O. Ramos Terrades, E. Valveny and S. Tabbone *Combination of ridgelets descriptors using optimal linear combination rules*