Graphics Recognition using Spatial Relations and Shape Analysis

Santosh K.C.

Nancy-Université INPL

Département de formation doctorale en informatique

École doctorale IAEM Lorraine

Graphics Recognition using Spatial Relations and Shape Analysis

THÈSE

présentée et soutenue publiquement le 28 Novembre 2011

pour l'obtention du

Doctorat de l'Institut National Polytechnique de Lorraine

(spécialité informatique)

par

Santosh K.C.

Composition du jury

Rapporteurs	Thierry PAQUET	Université de Rouen
	Jean-Yves RAMEL	Université de Tours
Examinateurs	Ernest VALVENY	Universitat Autònoma Barcelona
	Pierre-Etienne MOREAU	Nancy Université – INPL
	Laurent WENDLING	Université Paris Descartes (Paris V)
	Bart LAMIROY	Nancy Université – INPL

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

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Mis en page avec la classe thloria.

Acknowledgements

It is difficult to overstate my gratitude to my Ph.D. supervisors: Laurent WENDLING and Bart LAMIROY for their enthusiasm, inspiration, and huge effort to explain things clearly and simply throughout my research work and thesis writing. Without them, it is not possible. I also thank other members of my thesis committee: Thierry PAQUET, whose helpful suggestions increased readability and reduced possible ambiguity, and Jean-Yves RAMEL.

I am also indebted to many members in QGAR team for providing a stimulating and fun environment in which to learn and grow. I wish to thank my entire family – more specifically my mother. I also wish to thank Anju, for her care throughout the research period.

Lastly and most importantly, I gratefully acknowledge the funding sources that made my Ph.D. work possible: INRIA–CORDI research grant for 3 years and Nancy 2 Université for plus extra 2 months.

I dedicate this thesis to
my mother and late father, and of course to Anju.

***Santosh**.*

Summary

List of Figures xi

List of Tables xv

Part I Thesis Summary in French 3

Résumé de la Thèse en Français

1 Présentation . . . 6
1.1 Reconnaissance d'images Graphiques . . . 6
1.2 Cadre - Contributions . . . 9
2 Contributions à la Thèse . . . 13
2.1 Programmation Dynamique et Caractéristiques de Radon pour la Reconnaissance de Graphiques . . . 13
2.2 La Reconnaissance de Symboles à l'aide de Relations Géométriques . 20
2.3 Intégration des Signatures de Forme dans la Représentation ARG pour la Reconnaissance de Symboles . . . 27
2.4 Sac de Caractéristiques Spatiales pour la Reconnaissance de Symboles 31
2.5 Apprentsissage de Symboles par programmation logique inductive basée sur des descriptions formelles . . . 35

Part II Complete Thesis in English 41

Chapter 1
Introduction

1.1 Organisation of the Chapter . . . 43
1.2 Graphics Recognition . . . 44
1.2.1 Motivation . . . 44
1.2.2 Scientific Context . . . 46

1.2.3 Framework – Contributions of the Thesis 47
1.3 What is Next? 49

Chapter 2
State-of-the-Art Approaches

2.1 Organisation of the Chapter 51
2.2 Graphics Recognition 52
2.2.1 Statistical Approaches 52
2.2.2 Structural Approaches 54
2.2.3 Hybrid Approaches 63
2.3 Application Domains 64
2.4 Conclusions 65
2.5 What is Next? 66

Chapter 3
DTW–Radon: A New Shape Descriptor for Graphics Recognition

3.1 Organisation of the Chapter 68
3.2 Introduction 68
3.2.1 Related Work 69
3.2.2 Outline of the proposed Method 69
3.3 Method 70
3.3.1 Radon Features 70
3.3.2 Matching 75
3.3.3 Recognition and Retrieval 77
3.4 Experiments 78
3.4.1 Benchmarking Methods 78
3.4.2 Graphics Recognition and Retrieval 78
3.4.3 Case Studies 84
3.5 Conclusions 88
3.6 What is Next? 89

Chapter 4
Symbol Recognition using Spatial Relations

4.1 Organisation of the Chapter 92
4.2 Introduction 92
4.2.1 Related Work 92
4.2.2 Outline of the Proposed Recognition Method 94
4.3 Spatial Relations 94

4.3.1 Unique Reference Point Set . . . 95
4.3.2 Directional Relations . . . 95
4.4 Symbol Description . . . 103
4.4.1 Visual Vocabulary . . . 104
4.4.2 Pairwise Spatial Relations . . . 104
4.4.3 Attributed Relational Graph . . . 106
4.5 Symbol Recognition . . . 107
4.5.1 Matching . . . 107
4.5.2 Ranking . . . 109
4.6 Experiments . . . 109
4.6.1 Dataset and Evaluation Metric . . . 110
4.6.2 Matching Scope . . . 110
4.6.3 Experimental Results . . . 110
4.6.4 Experimental Results Analysis . . . 115
4.6.5 Advantages and Limitations . . . 117
4.7 Conclusions . . . 118
4.8 What is Next? . . . 118

Chapter 5
Integrating Shape Signatures in ARG Representation for Symbol Recognition

5.1 Organisation of the Chapter . . . 120
5.2 Introduction . . . 120
5.2.1 Related Work . . . 120
5.2.2 Outline of the Proposed Method . . . 121
5.3 Vertex Signature through Shape Descriptions in ARG . . . 122
5.3.1 Symbol Description . . . 122
5.3.2 Symbol Recognition . . . 123
5.4 Vocabulary Clustering through Shape Analysis in ARG . . . 124
5.4.1 Symbol Description . . . 124
5.4.2 Symbol Recognition . . . 134
5.5 Experiments . . . 137
5.5.1 Experimental Results . . . 138
5.5.2 Experimental Results Analysis . . . 140
5.5.3 Advantages and Limitations . . . 142
5.6 Conclusions . . . 142
5.7 What is Next? . . . 144

Chapter 6
Spatial-Bag-Of-Features for Symbol Recognition

6.1 Organisation of the Chapter . . . 146
6.2 Introduction . . . 146
6.2.1 Related Work . . . 147
6.2.2 Outline of the Proposed Method . . . 148
6.3 Spatial-Bag-Of-Features . . . 149
6.3.1 Vocabulary Categorisation . . . 149
6.3.2 Spatial Reasoning . . . 152
6.4 Symbol Recognition . . . 153
6.4.1 Relation Matching . . . 153
6.4.2 Symbol Recognition and Ranking . . . 154
6.5 Experiments . . . 154
6.5.1 Datasets and Evaluation Metric . . . 155
6.5.2 Experimental Results . . . 155
6.5.3 Experimental Results Analysis . . . 155
6.6 Advantages . . . 156
6.7 Conclusions . . . 160

Chapter 7
Summary

7.1 Conclusions . . . 161
7.2 Future Perspectives . . . 163

Part III Appendix **165**

Appendix A
Evaluation Metric

A.1 Background . . . 167
A.2 Evaluation Metric . . . 168
A.2.1 Recognition . . . 168
A.2.2 Retrieval . . . 168
A.3 Conclusions . . . 171

Appendix B
Affine Transformation Invariant Spatial Relation

B.1 Spatial Relations . . . 173
B.1.1 Radial line Model – Working Principle . . . 174
B.1.2 Affine Transformation Invariant properties . . . 174
B.2 Experiments . . . 175
B.2.1 Dataset . . . 175
B.2.2 Results . . . 177
B.2.3 Analysis . . . 177
B.3 Conclusions . . . 178

Appendix C
Symbol Learning via Inductive Logic Programming based on Formal Description

C.1 Introduction . . . 179
C.2 Inductive Logic Programming . . . 180
C.2.1 Fundamentals of the Inductive Logic Programming . . . 180
C.2.2 Global Behaviour – how does ILP work? . . . 181
C.2.3 State-of-the-Art . . . 182
C.3 Symbol Learning via Inductive Logic Programming based on Formal Description 184
C.3.1 Visual Vocabulary . . . 185
C.3.2 Symbol Description . . . 185
C.3.3 Symbol Recognition . . . 187
C.4 Conclusions . . . 191

Bibliography **193**

List of Figures

1.1 Document analysis or processing [Kasturi et al., 2002]. . . . 44
1.2 A sample of few electrical symbols [FRESH, 2007]. . . . 46
1.3 Selection of similar database symbols for a chosen query. . . . 47

2.1 A simple example, showing proximity graph representation using interest points using local descriptor. . . . 55
2.2 A simple example, illustrating graphical symbol representation using relational graph. . . . 55
2.3 Possible topological relations between two objects $\mathbb{A}$ and $\mathbb{B}$. . . . 57
2.4 Star calculus via bi-centre model using angle-based theory. . . . 58
2.5 MBR model showing horizontal and vertical projections. . . . 59
2.6 An example showing false overlapping. . . . 59
2.7 Graph representations: (a) only node features, (b) only relations and (c) integrating both. . . . 61
2.8 Inner transversal splitting example based on [Egenhofer and Shariff, 1998]. . . . 61

3.1 An architecture. . . . 69
3.2 Radon transform – a complete illustration: (a) pattern projection using the Radon transform theory and (b) radon features for all possible projections over $[0, \pi[$ and a complete radon transform i.e., a collection of all radon histograms or features. . . . 71
3.3 Images, their corresponding orientation estimation and the Radon features. Samples are taken from graphical symbol dataset [GREC, 2003]. . . . 72
3.4 Rotational images and their Radon features. Samples are taken from graphical symbol dataset [GREC, 2003, FRESH, 2007]. . . . 73
3.5 A single Radon feature over projection angle-range Θ. . . . 74
3.6 DTW algorithm – illustration. . . . 76
3.7 Computing DTW distance between two sequences having different lengths. . . . 76
3.8 Matching scores between reference, rotation, scaling and degradation sample images. A reference sample image is taken from GREC dataset [GREC, 2003]. It follows Figure 3.3. . . . 77
3.9 Matching scores between distortion as well as deformed images. A reference sample image is taken from FRESH dataset [FRESH, 2007]. . . . 78
3.10 A sample of few electrical symbols and their similar sample images. . . . 80
3.11 GREC2003 samples – graphical symbol. . . . 81
3.12 2 hand-drawn samples from 10 different known classes. . . . 83
3.13 2 numeral samples from 4 different known scripts for 10 $(0-9)$ classes. . . . 85
3.14 An example of binarisation sample from the ETL dataset. . . . 85
3.15 Difficulties in character recognition – a few examples. . . . 87
3.16 8 samples of 2 different known footwear print classes. . . . 88

4.1 An architecture for Symbol Description and Recognition using spatial relations. . 94
4.2 Asymmetry of spatial predicates in the case of MBR. 95
4.3 Reference point set $\mathbb{R}$ via topological relations from a pair. 96
4.4 Projection models for all possible types of $\mathbb{R}$. 97
4.5 Behaviour of computational time for boolean and metric or refined relation matrices. 98
4.6 Illustration of relational matrices using $\mathbb{R}$ 98
4.7 Radial line rotation $\circlearrowleft$ from a given reference point $\mathbb{R}_{p_c}$. 99
4.8 Relational histogram using radial line rotation. 100
4.9 Visibility angle values do not change even when objects shape changes 100
4.10 Different radial line resolutions effect precision. 101
4.11 Behaviour of computational time for boolean and metric histograms, using a few sets of resolutions: 1°, 3°,5° and 9°. 101
4.12 Histograms at 3° resolution for *disconnected* objects including scaling. 102
4.13 Histograms at 3° resolution for (a) false as well as (b) real *overlapping* objects. . 102
4.14 Histograms at 3° resolution for *overlapping* objects. 103
4.15 Illustration of vocabulary type. 104
4.16 Relational matrices and histograms for all possible pairs of vocabulary types. . . 105
4.17 Attributed Relational Graph for a *symbol 1* in Fig. 4.15 – an example. 106
4.18 Computing matching cost between two graphs G^A and G^B – an example [Aksoy, 2009]. 108
4.19 A graph transformation: $G^q \rightarrow G^d$. 109
4.20 An example illustrating the matching scope. 111
4.21 Average retrieval efficiency over requested list: 1 to 10 using projection model. . . 112
4.22 Average retrieval efficiency over requested list: 1 to 10 using radial line model. . . 112
4.23 Average retrieval efficiency over requested list: 1 to 10 using fundamental spatial relation models: 113
4.24 Average retrieval efficiency over requested list: 1 to 10 using global signal-based descriptors: Zernike moments, shape context, GFD, $\mathcal{R}-$signature and $\mathcal{D}-$Radon. 114
4.25 Average retrieval efficiency over requested list: 1 to 10 using pixel-based approaches: 114
4.26 Comparison. 115
4.27 Visual illustration of symbol retrieval and ranking for a few queries, showing the true (✓) and false (✗) retrieval. 116

5.1 An architecture for symbol description and recognition using shape features and spatial relations. 122
5.2 A complete ARG description for a symbol. 123
5.3 A complete ARG description for a symbol. 125
5.4 Symbols and their corresponding *thick* patterns. 126
5.5 Dendrogram example. 128
5.6 Sample points. 129
5.7 Cluster validation indices give two clusters. 133
5.8 A real example of how cluster validation indices behave for GFD. 134
5.9 Clustering *thick* patterns using GFD shape descriptor – at a glance. 135
5.10 Searching for the right clusters where query *thick* pattern belongs to. 136
5.11 *thick* pattern(s) in cluster(s) and its (their) corresponding symbol. 137
5.12 Average retrieval efficiency over requested list: 1 to 10 using global signal-based shape descriptors as vertex signature in our ARG framework. 138
5.13 Average retrieval efficiency over requested list: 1 to 10 using global signal-based shape descriptors as vertex signature in our ARG framework. 139

5.14 Comparison. . . . 140
5.15 Average retrieval efficiency over requested list: 1 to 10 using global signal-based shape descriptors for *thick* patterns clustering. . . . 141
5.16 Visual illustration of symbol ranking at the output for a few queries: $Q1$, $Q2$ and $Q3$, showing ✓ for true retrieval and false, otherwise. . . . 143

6.1 A symbol and its visual elementary parts i.e., primitives: *circle* and four *corners*. 146
6.2 Bag-Of-Features model – an example. . . . 147
6.3 Our spatial-BOFs model – a symbolic example. Each item represents a visual primitive and its colour represents a vocabulary type information. . . . 148
6.4 An architecture of the symbol recognition method based on spatial-bag-of-features. 149
6.5 Vocabulary categorisation in a pair. . . . 150
6.6 An example of vocabulary categorisation. . . . 151
6.7 A *disconnected* pair and directional relations using MBR model. . . . 153
6.8 Relation matching example in a particular category: *disconnected*. . . . 154
6.9 Average retrieval efficiency over requested list: 1 to 10 for FRESH dataset. . . . 156
6.10 Symbol retrieval using isolated symbols as queries: $\mathcal{Q}_1$ and $\mathcal{Q}_2$. . . . 157
6.11 A few examples of a collection of visual primitives (including their spatial organisation) of only type or more than one, as queries. . . . 157
6.12 Symbol retrieval example for FRESH dataset. . . . 158
6.13 Symbol retrieval example for GREC dataset. . . . 159

B.1 Computing spatial relations using radial–line rotation. . . . 174
B.2 Histograms at 3° resolution. Real-world example is taken from *symbol 1* in Figure 4.15. . . . 175
B.3 Sample images for a few classes. . . . 176

C.1 ILP scheme. . . . 182
C.2 Character description based on idea presented in [Amin, 2003]. . . . 183
C.3 Dependency tree for a sentence: 'Fraun is a German Optician' [Horváth et al., 2009]. 183
C.4 Name of the entities. . . . 184
C.5 First-order logical probability tree [Fierens, 2010]. . . . 184
C.6 An architecture for semantic related symbol recognition using inductive logic programming based on formal description. . . . 185
C.7 Visual primitives or vocabulary types: *thick*, *circle*, *corner* and *extremity* for corresponding symbols. . . . 186
C.8 First image set for ILP experimentation. . . . 188
C.9 Second image set for ILP experimentation. . . . 190
C.10 Third image set for ILP experimentation. . . . 190

List of Tables

3.1 Retrieval efficiency (in %) over 50 queries for FRESH dataset. 81
3.2 Average recognition rates (in %) for four different categories: ideal, rotation, scaling and rotation + scaling (from 50 models) of GREC dataset. 82
3.3 Average retrieval rate (in %) for CVC dataset. 83
3.4 Average running time. 84
3.5 Average recognition rate in % using $\mathbb{K}$–fold CV (where $\mathbb{K} = 5$). 86
3.6 Average recognition rate (in %) for full print as well as partial print. 88

4.1 Average running time. 117

5.1 Cophenetic correlation coefficient from all possible combinations of distance metric and clustering linkage method. 130
5.2 Cluster validation test for all indices. 132
5.3 Number of clusters after cluster validation tests from GFD. 133
5.4 Average running time. 142

6.1 Possible number of combinations that exist for all pairs of visual primitives. . . . 152
6.2 Recognition rate in % for a few categories of GREC2003 dataset. 156

B.1 Recognition rate in % for all categories. 177

C.1 Language of FOL – grammar [Russell and Norvig, 2010]. 182

Abstract

In the current state-of-the-art, symbol recognition usually means recognising isolated symbols. However, isolated symbol recognition methods are not always suitable for solving real-world problems. In case of composite documents that contain textual and graphical elements, one needs to be able to extract and formalise the links that exist between the images and the surrounding text, in order to exploit the information embedded in those documents.

Related to this context, we first introduce a method for graphics recognition based on dynamic programming matching of the Radon features. This method allows to exploit the Radon Transform property to include both boundary and internal structure of shapes without compressing the pattern representation into a single vector that may miss information. The method outperforms all major set of state-of-the-art of shape descriptors but remains mainly suited for isolated symbol recognition only. We therefore integrate it in a completely new approach for symbol recognition based on the spatio-structural description of a 'vocabulary' of extracted visual primitives. The method is based on spatial relations between pairs of labelled vocabulary types (some of which can be characterised with the previously mentioned descriptor), which are further used as a basis for building an attributed relational graph (ARG) to describe symbols. Thanks to our labelling of attribute types, we avoid the general NP-hard graph matching problem. We provide a comprehensive comparison with other spatial relation models as well as state-of-the-art approaches for graphics recognition and prove that our approach effectively combines structural and statistical descriptors together and outperforms them significantly.

In the final part of this thesis, we present a Bag-Of-Features (BOFs) approach using spatial relations where every possible pair of individual visual primitives is indexed by its topological configuration and the visual type of its components. This provides a way to retrieve isolated symbols as well as significant known parts of symbols by applying either an isolated symbol as a query or a collection of relations between the important visual primitives. Eventually, it opens perspectives towards natural language based symbol recognition process.

Keywords — Radon Features, Dynamic Programming, Shape Descriptors, Visual Vocabulary, Spatial Relations, Spatial-Bag-of-Features, Graphics Recognition.

Résumé

Dans l'état de l'art actuel, la reconnaissance de symboles signifie généralement la reconnaissance des symboles isolés. Cependant, ces méthodes de reconnaissance de symboles isolmés ne sont pas toujours adaptés pour résoudre les problèmes du monde réel. Dans le cas des documents composites qui contiennent des éléments textuels et graphiques, on doit être capable d'extraire et de formaliser les liens qui existent entre les images et le texte environnant, afin d'exploiter les informations incorporées dans ces documents.

Liés à ce contexte, nous avons d'abord introduit une méthode de reconnaissance graphique basée sur la programmation dynamique et la mise en correspondance de caractéristiques issues de la transormée de Radon. Cette méthode permet d'exploiter la propriété de cette transformée pour inclure à la fois le contour et la structure interne des formes sans utiliser de techniques de compression de la représentation du motif dans un seul vecteur et qui pourrait passer à côté d'informations importantes. La méthode surpasse en performances les descripteurs de forme de l'état de l'art, mais reste principalement adapté pour la reconnaissance de symboles isolés seulement. Nous l'avons donc intégrée dans une approche complètement nouvelle pour la reconnaissance de symboles basé sur la description spatio-structurelle d'un «vocabulaire» de primitives visuelles extraites. La méthode est basée sur les relations spatiales entre des paires de types étiquetés de ce vocabulaire (dont certains peuvent être caractérisés avec le descripteur mentionné précédemment), qui sont ensuite utilisées comme base pour construire un graphe relationnel attribué (ARG) qui décrit des symboles. Grâce à notre étiquetage des types d'attribut, nous évitons le problème classique NP-difficile d'appariement de graphes. Nous effectuons une comparaison exhaustive avec d'autres modèles de relations spatiales ainsi qu'avec l'état de l'art des approches pour la reconnaissance des graphismes afin de prouver que notre approche combine efficacement les descripteurs statistiques structurels et globaux et les surpasse de manière significative.

Dans la dernière partie de cette thèse, nous présentons une approche de type sac de caractéristiques utilisant les relations spatiales, où chaque paire possible primitives visuelles est indexée par sa configuration topologique et les types visuels de ses composants. Ceci fournit un moyen de récupérer les symboles isolés ainsi que d'importantes parties connues de symboles en appliquant soit un symbole isolée comme une requêten soit une collection de relations entre les primitives visuelles. Finalement, ceci ouvre des perspectives vers des processus de reconnaissance de symboles fondés sur le langage naturel.

Mots-clés — descripteur de Radon, programmation dynamique, descripteurs de forme, vocabulaire visuel, relations spatiales, sac de caractéristiques spatiales, reconnaissance graphique.

Part I

Thesis Summary in French

Résumé de la Thèse en Français

Avant-propos. Cette thèse traite de la reconnaissance de symboles graphiques. La reconnaissance de symboles ne signifie pas seulement reconnaître les symboles isolés, mais aussi ceux qu'on trouve dans la forme composite, ou relié avec d'autres éléments graphiques, en plus de textes. Par conséquent, la thèse vise à récupérer ces symboles qui sont similaires en termes d'apparence, soit dans leur totalité, soit en grande partie.

Les expériences soutenant notre travail utilisent des symboles électriques contenus dans des schémas de câblage. Ces diagrammes sont d'une série de données inédites appelée ensemble de données FRESH et déjà utilisée dans [Salmon, 2008]. Les approches développées dans cette thèse sont aussi validés sur des données telles que des symboles architecturaux, symboles dessinés à la main, ensembles de caractères manuscrits pour plusieurs scripts différents comme les Indiens, Japonais Katakana et romaine ainsi que d'un ensemble de données des chaussures impression preuves. ■

Mise en garde — Ce résumé en français a été obtenu avec des outils de traduction automatiques imparfaits, suivi d'une relecture. Quelques formulations malheureuses peuvent persister. Le lecteur intéressé mais néanmoins confus, est invité à se reporter à la version détaillée en anglais.

1 Présentation

Cette section fournit l'idée générale de la thèse. Nous commençons avec une description générale de la reconnaissance graphique sous l'angle de l'analyse de documents. Ensuite, nous soulignons les problèmes sous-jacents, les besoins et l'importance des enjeux motivant notre travail. Par la suite, l'objectif principal, la reconnaissance de symboles, est présenté et discuté explicitement en quelques points majeurs.

1.1 Reconnaissance d'images Graphiques

Motivation

Analyse ou traitement de documents est principalement lié aux textes et graphiques. Il s'agit de la séparation, localisation et de reconnaissance. Selon [Nagy, 2000], l'analyse de documents est liée à l'analyse d'images de documents (DIA). Depuis, les travaux de recherche globale ont été concernés par l'interprétation des images de documents. Les tâches de base sont la segmentation d'images, l'analyse de mise en page et la reconnaissance de symboles. De la même manière, [Kasturi et al., 2002] catégorise l'analyse d'images de documents dans deux domaines:

1. traitement de texte et
2. traitement graphique.

La Figure 1.1 montre une perspective de celui-ci avec un autre sous-catégorisation. Le traitement graphique se réfère à l'analyse des lignes, des angles et des courbes, d'un côté, tandis que de l'autre côté, il est préoccupé avec les régions et l'analyse des symboles.

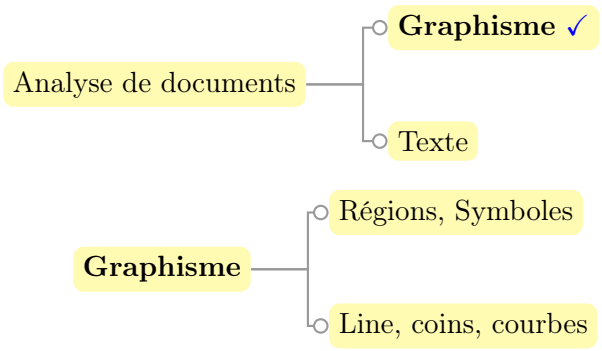


Figure 1: Analyse ou traitement de documents [Kasturi et al., 2002].

Une partie de l'analyse de l'image du document signifie de reconnaître les éléments graphiques et ou de localiser des parties visuelle importantes ou connues des images. Dans le contexte ci-dessus, la reconnaissance de graphismes (par exemple, la reconnaissance de symboles graphiques) a été une activité de recherche intensive en la communauté de la reconnaissance des formes et analyse de documents, depuis quelques décennies. La reconnaissance de symboles, notamment, a une longue histoire depuis les années 70. En 1998, [Tombre, 1998] a fait une déclaration: «Aucune de ces méthodes ne fonctionne». Depuis, le domaine a été activement étendu. La reconnaissance de symboles - la partie centrale de l'analyse de documents et la reconnaissance d'image graphique systèmes - joue un rôle important dans une variété d'applications telles que la reconnaissance automatique et l'interprétation des

- schémas de circuits [Okazaki et al., 1988, Feng et al., 2009],
- dessins d'ingénierie [Yang et al., 2007] et
  dessins architecturaux [Lladós et al., 2001, Valveny and Martí, 2003],
- notations musicales [Rebelo et al., 2010],
- cartes [Samet and Soffer, 1996a],
- expressions mathématiques [Chaudhuri and Garain, 2000], as well as
- caractères optiques [Yuen et al., 1998, Heutte et al., 2004].

Par conséquent, un symbole peut être défini comme une entité graphique avec une signification particulière dans le contexte d'un domaine spécifique. En tant que tel, et exception faite du prétraitement, l'ensemble du processus de reconnaissance de symboles est basée sur deux prédicats:

1. la fonction de correspondance entre une requête et des symboles ou des données
2. la comparaison de parties décomposées comme des lignes et des arcs ainsi que leurs relations entre eux.

L'analyse de formes pour la représentation des symboles est un exemple d'approche, liée au point 1. Un aperçu intéressant de la performance des descripteurs de forme les plus couramment utilisés pour la représentation des symboles est donnée dans [Valveny et al., 2007]. Ils sont communément appelés approches statistiques. D'autre part, le point 2. se réfère à des approches structurelles. Ils incluent principalement les problèmes de graphes, comme les graphes attribués relationnels (ARG), les graphes d'adjacence de régions (RAG) [Lladós et al., 2001] et des réseaux de contraintes [Ah-Soon and Tombre, 2001]. Dans les deux cas, les performances de reconnaissance s'expriment en fonction du taux de reconnaissance et de l'efficacité et la complexité du calcul. La première question est largement basée sur la façon dont les symboles sont représentés, à savoir la qualité de la représentation de symboles (soit juste un contour d'information ou de l'apparence globale est pris en compte) et la complexité de calcul basée sur des techniques d'appariement. En outre, en général, les techniques d'appariement sont souvent induites par la façon dont les symboles sont représentés. Signatures statistiques se trouvent être un meilleur choix, où des symboles linéaires ou isolés sont traités et aussi quand la complexité en temps est un problème. En revanche, les techniques structurelles sont capables de traiter tous les types de symboles (par exemple, isolés, composite), et avoir une représentation puissante relationnelle. Cependant, ils souffrent de la complexité des calculs intenses en raison de la NP-difficulté du problème de l'appariement de sous-graphe, résultant de la variation de la structure de graphe avec le niveau de bruit, l'occlusion, la distorsion etc.

Dans l'état de l'art courant, la reconnaissance de symboles signifie généralement la reconnaissance des symboles isolés. Ce n'est pas toujours adapté à la résolution de problèmes du monde réel, cependant. Considérons une situation de deux symboles isolés ainsi que les symboles composites qui sont connectés avec d'autres éléments graphiques. Dans un tel cas, les signatures statistiques en utilisant les descripteurs de forme, par exemple, ne peuvent pas bien performer, car ils prennent l'apparence globale en compte. Les approches structurelles ne donnent pas bons résultats en raison de la complexité des calculs intenses. Dans ce contexte, très récemment, [Tombre, 2010] pose la question de «ce qui distingue la reconnaissance des caractéristiques

des graphiques des problèmes généraux de reconnaissance des formes?» Car il a été indiqué que les méthodes varient d'une application à l'autre. Par conséquent, il existe toujours un intérêt intense pour le domaine de reconnaissance graphique. Considérant tous les points mentionnés ci-dessus, cette thèse porte sur l'utilisation de signatures structurelles et statistiques ainsi que leurs combinaisons possibles et efficaces. Faisons remarquer les commentaires qui ont été faits récemment [Tombre, 2010], à partir de laquelle la thèse est inspirée:

- «... le souhait récurrent de méthodes capables de combiner efficacement structurelles et les méthodes statistiques» et
- «le caractère très structurelle et spatiale de l'information que nous fait travailler avec méthodes structurelles tout à fait naturel dans la communauté».

Basé sur les points susmentionnés, l'intégration efficace à partir de laquelle nous pouvons profiter des avantages complète des approches statistiques et structurelle, est le principal objectif de la thèse. Nous présentons une méthode pour la reconnaissance de symboles basée sur la similitude entre l'organisation spatiale et les caractéristiques de la forme d'éléments visuels qui composent le symbole. Nous nous adressons à l'utilisation des relations spatiales entre des parties visuelles en premier et ensuite les combiner avec des fonctions de forme dans le cadre d'ARG. En raison d'un choix particulier dans l'étiquetage des sommets représentant des parties extraites connues, leur mise en correspondance devient triviale et peut être faite en un temps quasi-constant, évitant ainsi le problème NP-difficile de mise en correspondance de graphes. En parallèle, nous introduisons un descripteur DTW-Radon pour la reconnaissance de graphiques. Les méthodes proposées fournissent une meilleure performance de reconnaissance par rapport aux algorithmes de l'état l'art. Par ailleurs, cette thèse aborde également une technique d'apprentissage qui vise à la reconnaissance de symboles sémantiques liés. Il utilise la programmation logique inductive (PLI) pour apprendre automatiquement des descriptions non triviales de symboles, basé sur une description formelle dérivés à partir des informations structurelles et statistiques mentionnées précédemment. Une telle description de symboles est exprimée par un certain nombre de primitives visuelles et de leurs relations possibles. Comparé à des modèles statistiques d'apprentissage, le système s'adapte à la complexité de la classification en ce qui concerne les données d'apprentissage.

Une liste détaillée de nos contributions sera donnée dans la Section 1.2. Avant cela nous allons revoir et mettre en avant notre principal problème scientifique dans la section suivante.

Contexte Scientifique

Le travail est inspiré par un problème réel du monde industriel [Tombre and Lamiroy, 2008, Salmon, 2008] qui consiste à identifier un ensemble de symboles connus dans les schémas de câblage électrique des avions, afin de guider des algorithmes de simulation [FRESH, 2007]. Les défis viennent essentiellement du fait que les symboles de test sont dans des formes variées. Ces symboles peuvent être soit très semblables dans la forme - et ne diffèrent que par peu de détails - ou soit être complètement différents d'un point de vue visuel. Les symboles peuvent également être composés d'autres symboles connus et significatifs et ne doivent pas nécessairement être connecté. Quelques échantillons sont présentés dans la 2.

Pour l'évaluation, plusieurs évaluations différentes humaine car il n'y a pas d'absolu de vérité terrain associée à notre ensemble de données – ce qui est évident en cas d'application au monde réel. Pour gérer cela, nous leur avons demandé de sélectionner manuellement ce qu'ils considèrent comme " similaires"symboles, pour toutes les requêtes en fournissant les informations suivantes. En ce qui concerne la requête fournie, le candidat choisi doit avoir soit

- similaires apparence visuelle globale ou
- similaire parties significatives ou d'éléments graphiques en son sein.

En termes plus formels, pour chaque requête de la "vérité-terrain"est considéré comme l'ensemble des symboles formés par l'union de tous les ensembles humains sélectionnés. Pour mieux comprendre, la Figure 3.10 montre quelques exemples de quelques requêtes.

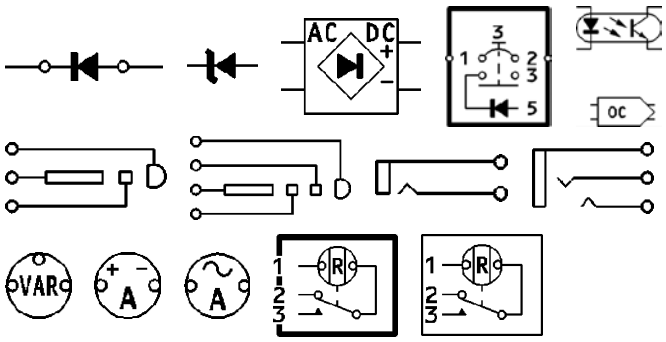


Figure 2: Un échantillon de quelques symboles électriques [FRESH, 2007].

1.2 Cadre - Contributions

La reconnaissance de symboles signifie généralement la reconnaissance des symboles isolés. Dans ce contexte, nous présentons une approche basée descripteur afin de décrire l'apparence globale des symboles graphiques.

Contribution 1. Caractéristiques du radon et dynamique descripteur de forme de programmation basé pour la reconnaissance graphique.

Nous introduisons une méthode basée sur la programmation dynamique pour faire correspondre les caractéristiques de Radon, pour la reconnaissance de graphiques. La principale caractéristique de la méthode est d'utiliser l'algorithme du Dynamic Time Warping (DTW) pour trouver des paires de fonctions de Radon correspondantes pour tous les projections possibles. Pour la rendre invariante par rotation, nous estimons l'angle d'orientation général qui permettra d'éviter un mauvais alignement. Globalement, cela permet d'exploiter la propriété de Radon que comprend à la fois les contours ainsi que la structure interne de formes, tout en évitant la compression de la représentation du motif dans un seul vecteur et donc de manquer des informations, grâce à la DTW.

Dans la Section 2.1, nous présentons le concept en détail. Nous la validons en comparant avec un ensemble de descripteurs de forme majeurs sur plusieurs ensembles de données. Dans tous les tests expérimentaux, nous nous concentrons sur l'optimisation du nombre de pas d'échantillonnement et des projections, afin de réduire le la complexité des calculs.

Dans le cas de problèmes de reconnaissance concrets du monde réel, des méthodes de reconnaissance de symboles isolés ne sont pas toujours adéquats, cependant. Dans le cas des documents composites qui contiennent des éléments textuels et graphiques, on doit être capable d'extraire et de formaliser les liens qui existent entre les images et le texte les entourant, afin d'exploiter les informations incorporées dans ces documents. Par conséquent, l'extraction correcte, la représentation des deux données visuelles, des structures textuelles et graphiques, et l'organisation sont

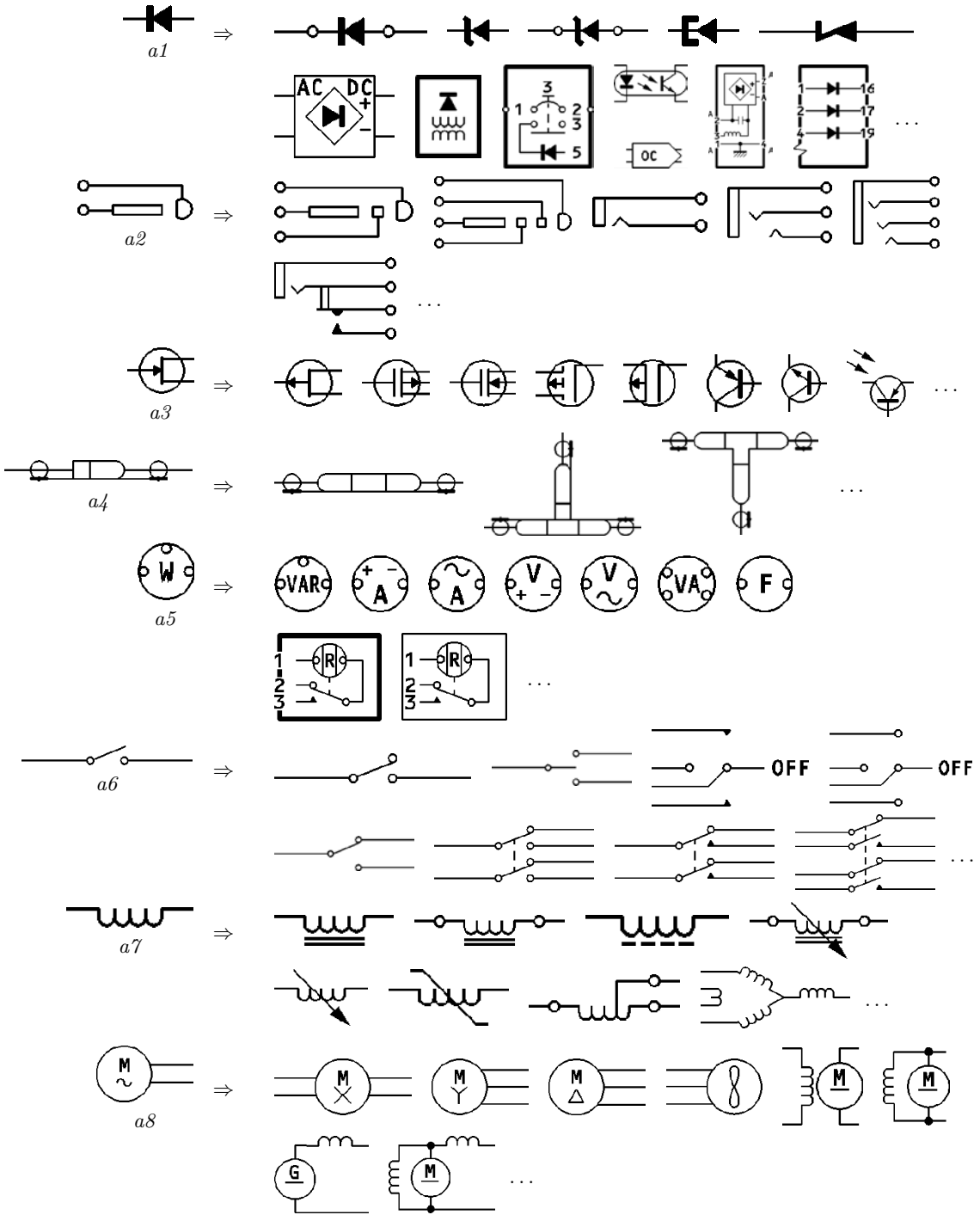


Figure 3: Un échantillon de quelques symboles électriques et de leurs images échantillon similaire. Pour chaque test symbole: *a1* à *a8*, quelques symboles pertinents sont enrôlés fondée sur l'évaluation humaine. Il se compose de linéaires ainsi que des symboles en forme composite.

les premières étapes vers la connaissance automatisé ou encore, la découverte de l'information et recherche d'information ou de data mining sur des données plus complexes qu'un simple texte.

Dans le contexte de la reconnaissance graphique, nous mettons principalement l'accent sur trois éléments principaux:

1. l'extraction des éléments visuels (vocabulaire) qui composent une image;
2. l'expression des relations visuelles entre les éléments;
3. découverte de connaissances, des techniques formelles de l'apprentissage et la classification en utilisant le vocabulaire et les relations mentionnées ci-dessus, y compris l'analyse de formes.

Les deux premiers points sont basés sur les opérations standards d'analyse et de traitement d'images [Rendek et al., 2004] alors, les contributions de la thèse s'appuient principalement sur le point 3. Sur cette base, il y a cinq principales contributions:

Contribution 2. Reconnaissance de symboles en utilisant des relations spatiales.

Nous présentons une méthode de reconnaissance de symboles basée sur la description spatio-structurelle d'un «vocabulaire» visuel extrait de pièces élémentaires. La méthode est essentiellement basée sur les relations spatiales entre les paires possibles de types de vocabulaire étiquetés (comme des cercles et les coins) qui sont utilisées comme base pour la construction d'un graphe attribué relationnel (ARG) qui décrit parfaitement le symbole. Ces relations spatiales intègrent à la fois la topologie et l'information directionnelle. Grâce à notre étiquetage des types d'attribut, l'appariement des relations correspondantes est possible entre les deux graphes, tout en évitant le problème général NP-difficile.

Dans la Section 2.2, nous fournissons une étude détaillée de la façon dont les relations spatiales sont utilisées pour représenter des symboles complets ainsi que d'une comparaison exhaustive avec d'autres modèles de relations spatiales, y compris l'état de l'art des approches pour la reconnaissance de graphiques.

Contribution 3. Intégration des relations spatiales avec des caractéristiques statistiques pour la reconnaissance de symboles.

Nous présentons la méthode de reconnaissance de symboles basée sur à la fois sur l'organisation spatiale et les caractéristiques de forme des pièces visuelles qui composent le symbole. Et maintenant la base ARG pour la description des symbole mentionnée dans la contribution de, les signatures forme sont intégrés de deux différentes manières. Premièrement, les signatures de forme sont utilisées pour étiqueter des sommets. Deuxièmement, les caractéristiques de forme sont appliquées uniquement au vocabulaire qui montrent des variations de forme significatives. Dans cette contribution, les caractéristiques statistiques sont intégrées là où les variations de forme significatives se produisent dans les parties visuelles.

Nous la présentons dans la Section 2.3. Nous nous concentrons sur les performances de reconnaissance en intégrant la forme caractéristiques sur les pièces visuelles. Pour gérer cela, nous étudions la pertinence de la plupart des grandes descripteurs de forme.

Contribution 4. Sac de caractéristiques spatiales pour la reconnaissance de symboles.

Nous introduisons un concept pour la reconnaissance de symboles qui est basé sur un sac de caractéristiques qui sont calculées à partir de primitives visuelles extraites. La principale caractéristique de l'ensemble consiste à utiliser des informations topologiques pour guider les relations métriques directionnelles. Ces indications topologiques rendent notre méthode efficace puisque la mise en correspondance n'utilisera va que les candidats pertinents qui

partagent la même configuration topologique et donc, il réduit le temps de calcul, et un recherche rapide est possible.

Dans la Section 2.4, nous l'expliquons en détail, y compris la façon de récupérer les lacunes des méthodes présentées dans les contributions précédentes. Outre les performances de reconnaissance, nous abordons les avantages et l'extension de la méthode de telle manière qu'elle peut être utilisée comme une méthode de recherche de symboles «conviviale». Par exemple, «trouver le symbole qui a une rectangle contenant un cercle».

Contribution 5. Techniques d'apprentissage formelles pour la reconnaissance de symboles sémantiquement liés.

Nous présentons l'utilisation de techniques formelles pour apprendre automatiquement des descriptions non triviales de symboles, basées sur une description formelle. L'ARG décrit précédemment à l'aide vocabulaire visuel et le positionnement relatif est transformé en une représentation en logique du premier ordre (FOL) basée sur les symboles graphiques. Cette représentation est ensuite utilisée comme une entrée d'un solveur ILP, afin d'en déduire des caractéristiques non évidentes qui peuvent conduire à un processus plus sémantique de reconnaissance.

Globalement, elle donne une caractérisation commune des sous-ensembles de symboles de l'ensemble des connaissances du domaine. Comme dit précédemment, la méthode peut s'adapter toute la complexité de la classification contrairement à des modèles statistiques d'apprentissage. Nous avons validé et expliqué l'approche dans la Section 2.5.

2 Contributions à la Thèse

2.1 Programmation Dynamique et Caractéristiques de Radon pour la Reconnaissance de Graphiques

Contexte

econnaissance automatique, description, classification et le regroupement des modèles sont des questions importantes dans de nombreuses disciplines telles que la biologie, la vision par ordinateur, l'intelligence artificielle ou la télédétection, où l'analyse de forme joue un rôle important [Loncaric, 1998, Zhang and Lu, 2004]. Patterns [Watanabe, 1985] par exemple, peuvent être des éléments graphiques dans le document-comme des symboles de toute nature, y compris caractères cursifs, les chaussures et les empreintes digitales.

Pour ce faire la reconnaissance des formes à travers l'analyse de forme, deux étapes majeures sont la représentation la forme et l'appariement. Dans le cadre, les questions suivantes seront prises en compte.

1. Représentation de forme

   - la représentation forme globale est largement utilisé en raison de sa simplicité. En outre, il ne nécessite pas de pré-traitements supplémentaires ou de segmentation, contrairement au modèle local représentation.
   - La représentation des formes doit être suffisamment enrichie et le contenu interne important ne doit pas être manqué.
   - Un autre problème de mise en œuvre commune est l'incapacité d'assumer la distribution des motifs dans l'espace des caractéristiques. Par conséquent, les méthodes non paramétriques sont beaucoup plus pratiques car ils utilisent des fonctions linéaires pour décrire les classificateurs.

2. Matching

   Les techniques correspondants sont souvent induits par la manière dont les modèles sont représentés. Par exemple, les renseignements concernant le modèle de normalisation dans un vecteur de taille fixe unique (comme dans le signal global basé descripteurs [Zhang and Lu, 2004]) fournit des correspondances immédiates.

En outre, les propriétés d'invariance aux transformations affines doit être incluse. Dans la littérature existante, des procédures de normalisation rend le vecteur de caractéristiques invariant à la rotation mais, de l'autre côté, il ne fournit pas d'informations sur la forme complète. Par conséquent, pour ce qui concerne les points mentionnés ci-dessus au point 1, les caractéristiques ne doivent pas être normalisés afin d'exploiter l'information de détail de la forme, y compris le contenu interne. Si c'est le cas, le coût correspondant sera augmenté. Par conséquent, un autre paramètre important d'invariance à la rotation doit être intégré.

Un état de l'art des descripteurs de forme d'analyse des documents a été faite dans [Terrades et al., 2007a], notamment concernant l'ensemble susmentionnés important de descripteurs de signal global. Parmi celles-la, l'intérêt pour la transformée de Radon peut être observée à partir des ouvrages suivants traitant de l'appariement d'images de lignes de dessin [Fränti et al., 2000], la catégorisation de formes 2D et de symboles [Leavers, 2000, Tabbone et al., 2006] et la démarche de reconnaissance [Boulgouris and Chi, 2007]. Ces travaux attestent de la solidité de la transformée de Radon sur des échantillons bruites, dégradées ainsi que comportant des occultations.

Pour toutes ces raisons, nous profitons de la transformée de Radon [Deans, 1983] pour représenter. Le modèle de programmation dynamique (DP) est utilisé pour faire correspondre les modèles de toute taille en évitant la compression des motifs représentés dans un seul vecteur, contrairement à l'utilisation de la $\mathcal{R}-$transform [Tabbone et al., 2006], par exemple. Le travail est le prolongement des travaux antérieurs conçu pour la vérification de signatures hors ligne [Coetzer, 2005, Jayadevan et al., 2009] ainsi que l'estimation que l'angle d'orientation dans [J-K and S-Z, 2005]. Globalement, une sélection optimale du nombre de bacs pour recueillir les caractéristiques de Radon sera une partie intéressante du travail, en plus de l'intégration appropriée des orientation estimée. Par ailleurs, nombre de bacs qui détermine l'intérêt optimal entre les deux: l'information détaillée de la forme et le temps d'exécution.

Méthode Proposée

Dans le cadre des points soulignés ci-dessus, nous utilisons la transformée de Radon [Deans, 1983] pour représenter les motifs. La correspondance des modèles est alors faite par le biais de DTW [Bellman and Kalaba, 1959, Kruskall and Liberman, 1983, Keogh and Pazzani, 1999] entre les caractéristiques du radon correspondante à chaque projection spécifiée. Les transformés de Radon sont essentiellement un ensemble d'histogrammes ou de caractéristiques paramétrisés. Par conséquent, la méthode traite du choix optimal du nombre de bacs contrairement à des versions plus simples utilisant la transformée de Radon discrétisée. À savoir, il existe un compromis entre les temps de calcul et le nombre optimal de sélection des bacs.

Avant de s'aligner, le principal problème est de rendre la fonction invariante aux transformations affines. Notamment la propriété de rotation a été soigneusement intégré avec des fonctions de Radon. La translation restante et les propriétés d'échelle sont manipulées en utilisant le centroïde de l'image.

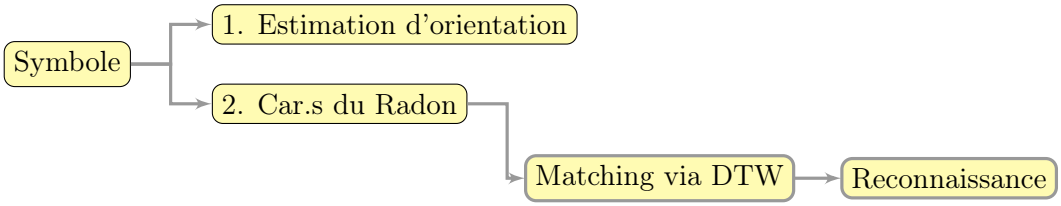


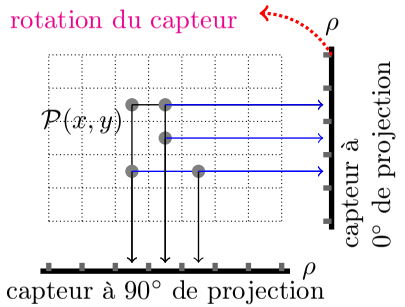
Figure 4: Une architecture.

La Figure 4 montre une architecture de la méthode. En particulier, la méthode est capable de récupérer l'inconvénient de la $\mathcal{R}-$signature classique [Tabbone et al., 2006] qui n'exploite pas l'information de la forme complète parce que l'information de Radon à chaque angle saillant est compressé en une seule valeur via la $\mathcal{R}-$Transform. La Figure 5 montre une illustration complète de la transformation de radon discrète.

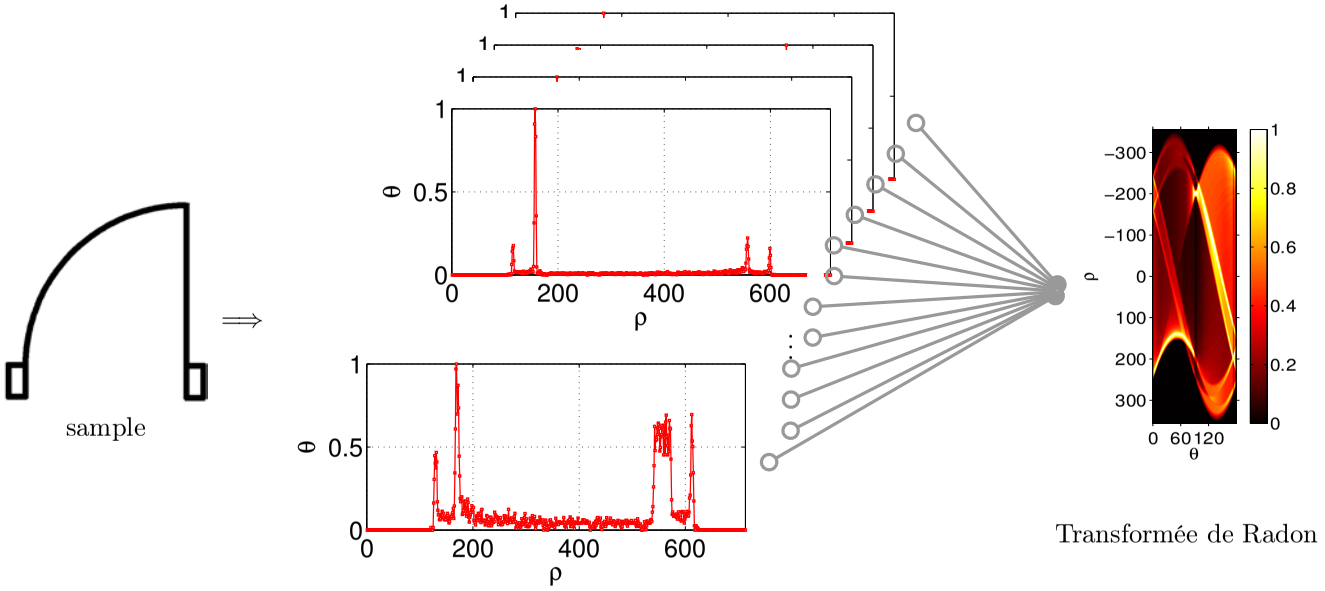
Validation

Pour valider la méthode, nous utilisons plusieurs ensembles de données différents.

1. FRESH [FRESH, 2007]: Un échantillon de quelques images peut être vu à la Figure 2. Comme dit précédemment, le jeu de données est composé de symboles isolés ainsi que composites. Dans ce test, nous visons à récupérer les symboles basé sur l'apparence globale.



(a) de projection par exemple.



(b) Caractéristiques de Radon montrant les projections, pour l'échantillon correspondant

Figure 5: Transformée de Radon - une illustration complète: (a) le modèle de projection et (b) caractéristiques de Radon pour toutes les projections possibles sur $[0, \pi[$ et une transformation complète de Radon.

2. GREC [GREC, 2003]: Dans ce jeu de données (à partir du concours international de reconnaissance de symboles GREC), nous avons utilisé les différentes catégories suivantes: idéal, rotation, échelle, la distorsion ainsi que dégradation.

   Au total, il ya 50 symboles modèles différents. Ces symboles sont regroupés en trois ensembles, contenant des symboles du modèle 5, 20 et 50. Chaque symbole du modèle dispose de 5 images de test dans chaque catégorie, sauf l'idéal. Les images de test idéales sont directement pris dans l'ensemble du modèle symboles et donc le test est d'évaluer la capacité de discrimination de forme simple en fonction du nombre de symboles. Depuis la distorsion vectorielle fonctionne uniquement avec des symboles avec des lignes droites, des arcs et non, il est appliqué à un sous-ensemble de 15 symboles modèle. En outre, il ya 9 modèles de dégradation, visant à évaluer la robustesse en fonction de la dégradation. La Figure 6 montre quelques exemples de données GREC.

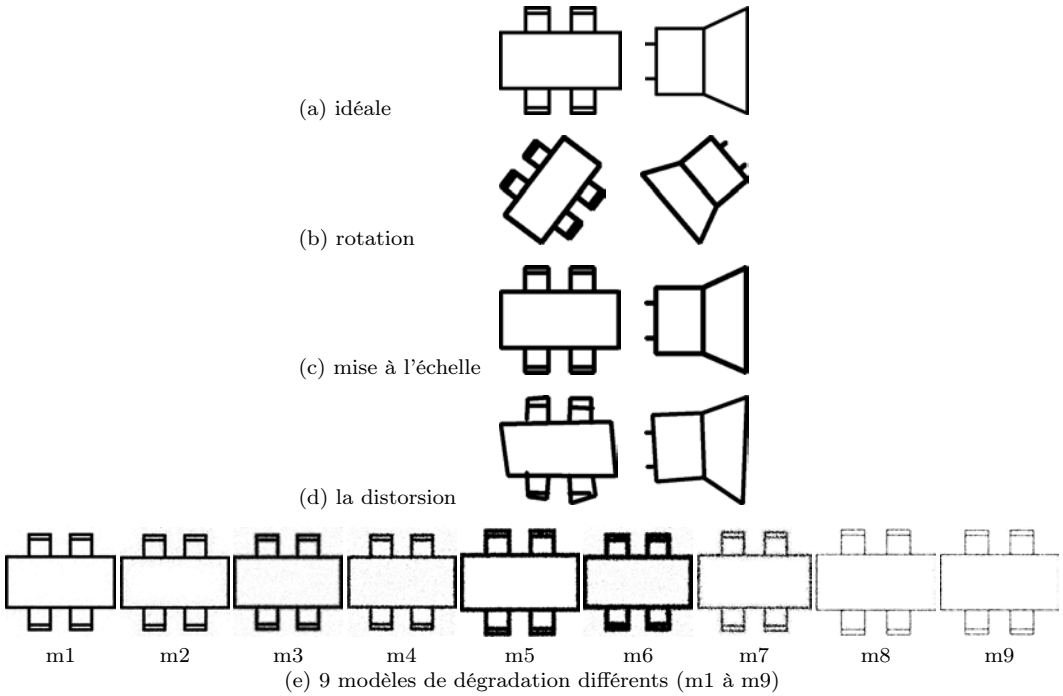


Figure 6: GREC2003 échantillons de symboles graphiques..

3. CVC [Wendling et al., 2008, Fornés et al., 2010]: Dans ce jeu de données, nous avons testé 10×300 l'échantillons d'images de 10 différentes classes connues de symboles architecturaux dessinés à la main avec 300 cas chacun. Les symboles ont été établis par 10 personnes différentes en utilisant un stylo «Anoto». Les échantillons présentent des distorsions, des trous, des chevauchements ainsi que les pièces manquantes dans les formes. La Figure 7 montre quelques exemples de l'ensemble des données.

4. Jeux de données numériques manuscrites: Plusieurs ensembles de données différents de différentes écritures indiennes telles que le bengali, devanagari et l'oriya [Bhowmik et al., 2006, Bhattacharya and Chaudhuri, 2009] sont utilisés. En outre, les données en chiffres romains est également testée (ETL3, Roman ensemble de données, l'AIST, Japon). La Figure 8 montre quelques images des échantillons dans chaque classe.

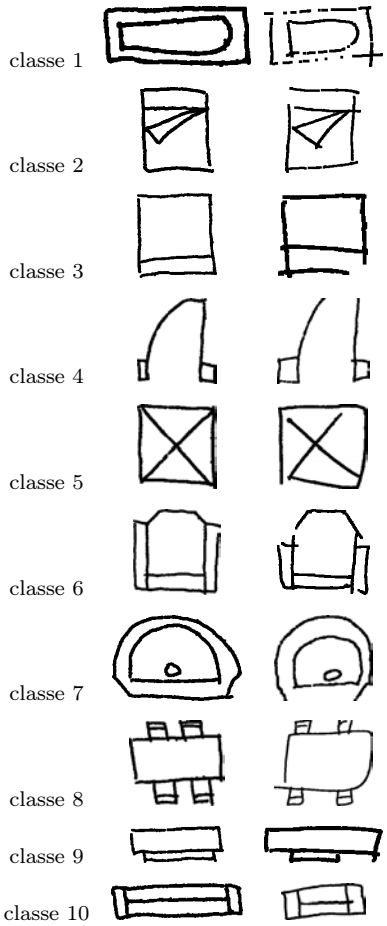


Figure 7: Des échantillons dessinés à la main provenant de 10 différentes classes connues.

5. Empreintes de semelles: Dans ce jeu de données [1], nous avons 64 empreintes de semelles de chaussures ou de pieds l'usure des classes d'impression, chacun avec les instances de hauteur. Les Images des empreintes sont classées en empreintes complètes et empreintes partielles. La Figure 9 montre des images de l'échantillon pour quelques classes. Dans cette illustration, nous avons mis un cadre autour de chaque échantillon afin de distinguer clairement les empreintes pleines des empreintes partielles en plus de la forme et la taille. Comme indiqué dans la Figure 9, il y a quatre images dans la catégorie complète, y compris la rotation tandis que les autres sont partielles estampes. Les empreintes partielles contiennent également des portions de pointe et au talon.

A titre de comparaison, nous prenons un ensemble de descripteurs de forme majeurs comme avant. Ils sont

1. Moments de Zernike (ZM) [Kim et Kim, 2000],
2. Le Descripteur Générique de Fourier (GFD) [Zhang and Lu, 2002],
3. Contexte de forme (SC) [Belongie et al., 2002] et

[1] Merci au Signal Processing Laboratory de l'université Kinki, au Japon pour avoir fourni cet ensemble de données.

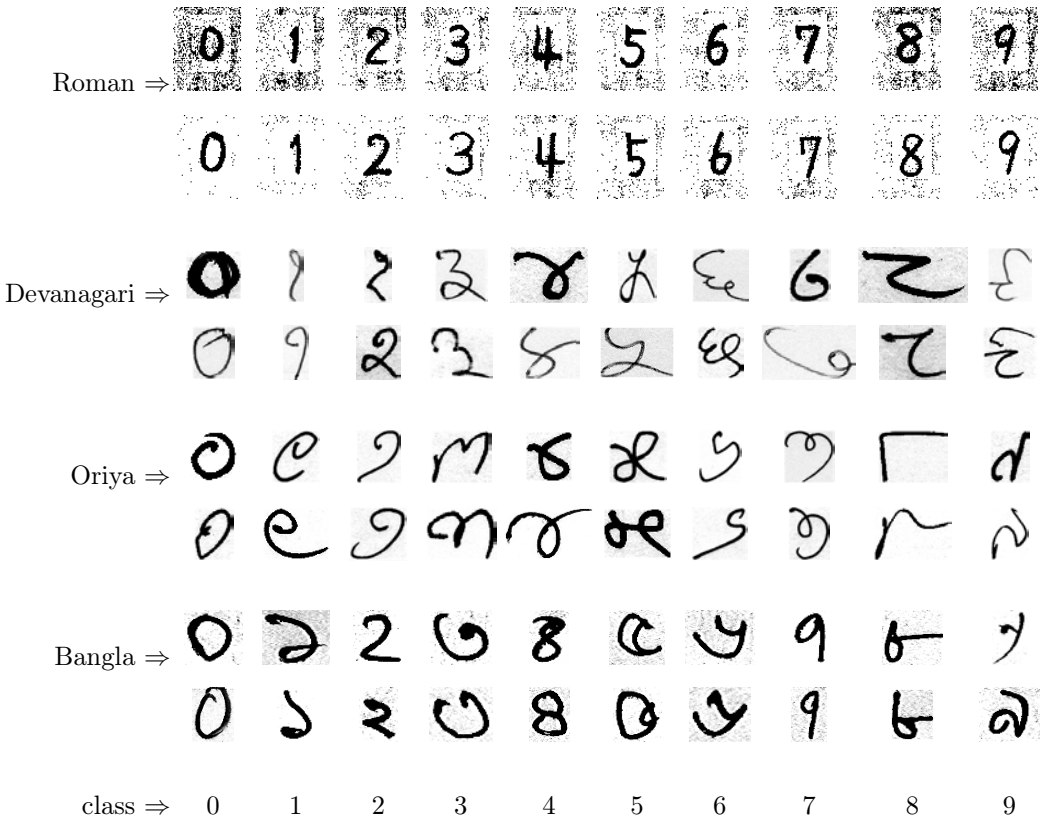


Figure 8: 2 échantillons de chiffres de 4 différents scripts connus pour 10 (0 - 9) classes.

4. $\mathcal{R}-$signature [Tabbone et al., 2006].

Résumé des Résultats. Tout au long des tests expérimentaux, l'image brute a été appliquée sans pré-traitement, à l'exception des ensembles de données de caractère. Dans ce qui suit, nous allons brièvement présenter les résultats des toutes les données susmentionnées.

1. Données FRESH: Comme il n'y a pas vérité terrain absolue associée à nos données, nous avons procédé en utilisant une validation humaine, mais en prenant soin d'éliminer les biais subjectifs. Pour ce faire, nous avons demandé à six bénévoles de sélectionner manuellement ce qu'ils considèrent comme symboles «similaires», pour toutes les requêtes exécutées dans cette section. Ces évaluateurs humains ont choisi les candidats qui ont la même apparence visuelle d'ensemble ou qui ont des parties sensiblement semblables à l'égard de la requête choisie. Comme le nombre de vérités terrain est variable d'une requête à l'autre, nous utilisons l'efficacité de récupération pour une demande short-list. Pour une demande short-list 10, $\mathcal{D}-$Radon a des rendements d'efficacité de récupération de 73%, ce qui constitue une différence notable avec les autres benchmarking de descripteurs de forme de l'état de l'art.

2. Données GREC: Les taux de reconnaissance dans toutes les catégories sont jugés similaires, sauf en cas des dégradations binaires. $\mathcal{D}-$Radon donne des résultats intéressants par rapport aux autres.

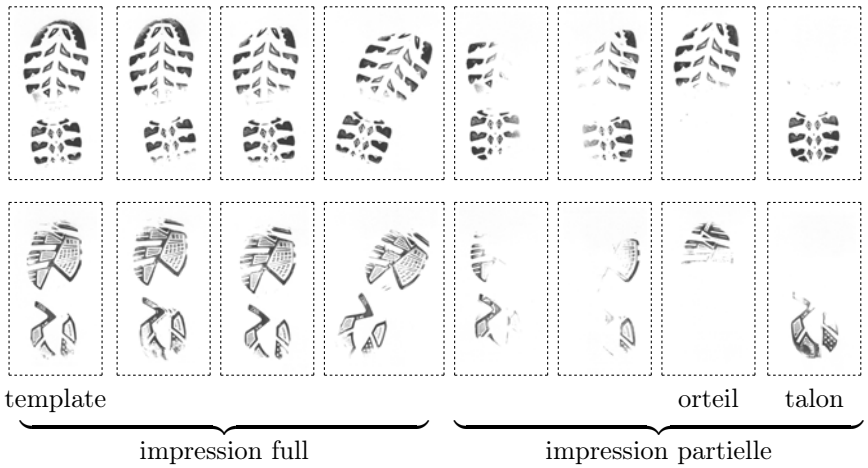


Figure 9: 8 échantillons de 2 différentes classes d'empreinte de semelles.

3. Données CVC: Taux de récupération à partir du $\mathcal{D}-$Radon donne taux de récupération de 86% pour une demande short-list de 300 à savoir top-300. Globalement, il fournit une différence de plus de 15% avec les autres.

Afin d'établir l'intérêt ainsi que l'extensibilité de la méthode, plusieurs ensembles différents de données sont également été utilisés. Plus précisément, nous avons surtout travaillé dans deux contextes différents.

1. Reconnaissance de caractères isolés: $\mathcal{D}-$Radon remplace toutes les méthodes, tout en offrant une différence marginale.

2. Vérification des empreintes de semelles: $\mathcal{D}-$Radon a des rendements plus élevés de taux de reconnaissance qui n'est pas plus de 60% en moyenne. En revanche, le taux le plus élevé de reconnaissance est d'environ 40% meilleur par rapport aux descripteurs de forme existants de l'état de l'art.

La méthode proposée montre clairement le pouvoir discriminant des caractéristiques de Radon par l'intermédiaire l'utilisation de la DTW. Il surpasse toutes les autres méthodes. Cependant, il offre un temps de calcul de haute complexité par rapport au vecteur de éléments à dimension fixe.

2.2 La Reconnaissance de Symboles à l'aide de Relations Géométriques

Contexte

La recherche sur la reconnaissance graphique a une très riche état de l'art de la littérature, visant à localiser ou reconnaître les symboles selon les applications. Selon [Cordella and Vento, 2000b, Lladós et al., 2002], ces méthodes sont particulièrement adaptées pour les symboles de ligne isolée, non pas pour les symboles composés reliés à un environnement complexe. Afin d'exploiter les informations incorporées dans ces documents, on doit être capable d'extraire des pièces visuelles et formaliser les possibles liens qui existent entre eux. Cette combinaison de la localisation basée sur le symbole extrait visuelle pièces va être au cœur de cette contribution.

Dans les approches structurelles, l'idée principale est de décomposer les symboles en primitives vectoriel soit basée comme des points, des lignes et des arcs ou en parties significatives, comme les cercles, des triangles et des rectangles. Ces primitives sont ensuite utilisés pour représenter des symboles en termes de graphes relationnels attribués (ARG) [Bunke and Messmer, 1995, Conte et al., 2004], graphes d'adjacence Région (RAG) [Lladós et al., 2001] réseaux de contraintes, [Ah-Soon and Tombre, 2001] comme ainsi que des modèles déformables [Valveny and Martí, 2003]. Leur inconvénient commun vient de erreurs raster-vecteur de conversion. Ces erreurs peuvent augmenter les confusions entre les différents symboles. Par ailleurs, la variabilité de la taille des graphes conduit à une complexité de calcul dans la mise en correspondance. Cependant, les approches structurelles fournissent une représentation puissante, permettant d'exprimer comment les parties sont reliées les unes aux autres, tout en préservant la généralité et l'extensibilité.

Pour décrire les symboles, il est nécessaire de gérer les relations entre les parties décomposées. Le paragraphe suivant donne un aperçu des travaux existants sur les relations spatiales et leur bonne usages.

Effet de relations spatiales sur les performances de reconnaissance ont été examinés en détail pour les compréhension scène [Bar and Ullman, 1993], l'analyse des documents et des tâches de reconnaissance [Garnesson and Giraudon, 1990, Pham and Smeulders, 2006]. Relations spatiales peuvent être soit topologiques [Egenhofer and Herring, 1991, Renz and Nebel, 1998], directionnel [Matsakis and Wendling, 1999, Mitra, 2002] et métriques dans la nature, leur choix en fonction du type d'application. Par exemple, dans [Xiaogang et al., 2004], configurations topologiques sont traitées avec un peu de prédicats, comme l'intersection, l'interconnexion, tangence, de parallélisme et concentricité exprimé selon une la norme des relations topologiques décrite dans [Egenhofer and Herring, 1991].

De la même manière, les différents modèles de relations directionnelles ont été développés pour une large gamme de situations différentes.

- Si les objets sont assez loin les uns des autres, leurs relations peuvent être approchées par leur centres basé sur l'angle discrétisé [Miyajima and Ralescu, 1994, Mitra, 2002]. Cette approche est robuste aux petites variations de forme et la taille des objets étudiés.

- Si elles ne sont ni trop loin ni trop près, les relations peuvent être approchées par leur rectangle minimal de délimitation (MBR) tant qu'ils sont réguliers [Peuquet and CI-Xiang, 1987, E.Jungert, 1993, Papadias and Theodoridis, 1997]. En d'autres termes, la qualité de la MBR dépend de la compacité de la tuile.

- Des approches comme Angle Histogrammes [Wang and Keller, 1999] ont tendance à être plus capables de traiter le chevauchement, ce que avec, les approches précédentes ont des difficultés. Toutefois, car ils considèrent tous les pixels, leur coût de calcul augmente considérablement.

- D'autres méthodes, comme par exemple, les F-Histogrammes [Matsakis and Wendling, 1999] utilisent des paires des sections longitudinales au lieu de paires de points, même au prix d'une complexité temps.

- Une autre approche bien connue utilise de la logique floue [Bloch, 1999], et est basé sur des opérateurs morphologiques flous.

Des approches mentionnées précédemment abordent soit seulement les relations topologiques, soit les directionnelles. Gésrer les des deux en même temps vient avec des coûts élevés de calcul. Même alors, aucun modèle existant intègre pleinement la topologie. Ils ont plutôt divers degrés de sensibilité ou de prise en compte des relations topologiques. Alors que des méthodes comme [Xiaogang et al., 2004] se focalisent sur des informations topologiques seulement, l'approche que nous développons dans cette thèse unifie l'information à la fois topologique et directionnelle dans un descripteur [K.C. et al., 2009c] sans aucun frais de temps de fonctionnement supplémentaire.

Pour placer les relations spatiales dans le contexte de la reconnaissance et la description des symboes, il convient de noter que les relations spatiales ont aussi une composante fondée sur la langue (liée à la compréhension humaine par exemple, "à la droite de") qui peut être formalisé de façon mathématique (par exemple, les 512 relations du modèle 9-intersections [Egenhofer and Herring, 1991]). Par conséquent, les relations qualitatives et quantitatives sont une autre façon de faire la catégorisation des relations spatiales. Prenons un exemple, un objet A qui s'étend à droite (98%) et en haut (2%) par rapport à B est exprimé en droit–Top(A,B). Ce prédicat spatial reste inchangé jusqu'à un changement raisonnable de la forme et la position des objets. Compte tenu de cela, notre travail utilise des relations plus naturelles que le tout-ou-rien des relations standard [Freeman, 1975].

Méthode Proposée

Notre méthode est fondée sur la reconnaissance spatio-structurelles de parties visuelles extraites qui composent un symbole. Cela signifie que, pour décrire un symbole, nous calculons les relations spatiales entre parties visuelles précédemment extraites. Le schéma complet est montré dans la Figure 10.

Sans aucune autre considération, il est évident que la taille du graphe relationnel résultant est potentiellement très large et variable d'un symbole à un autre. Toutefois, lorsque l'on regroupe les parties visuelles ensemble selon leurs types (par exemple, les cercles, les coins ...), nous pouvons éliminer tous les problèmes combinatoires inhérents à appariement de graphes, sans pour autant sacrifier à la qualité de la reconnaissance ou la puissance expressive. Nous calculons les relations spatiales entre les différents attributs étiquetés pour construire un graphe attribué relationnel.

Globalement, il ya deux étapes.

étape 1 La méthode consiste d'abord à identifier les éléments de vocabulaire dans différents groupes selon leur types (par exemple, un cercle, angle). Deux exemples sont présentés dans la Figure 4.15.

Nous définissons un ensemble de propriétés visuelles bien contrôlées des pièces élémentaires comme un vocabulaire [K.C. et al., 2009a]. Dans le cas générale, ce vocabulaire peut être de toute nature de tout type de "sacs de caractéristiques", liés à ce qui est visuellement pertinent dans le contexte de l'application. Notre vocabulaire courant est lié à des symboles

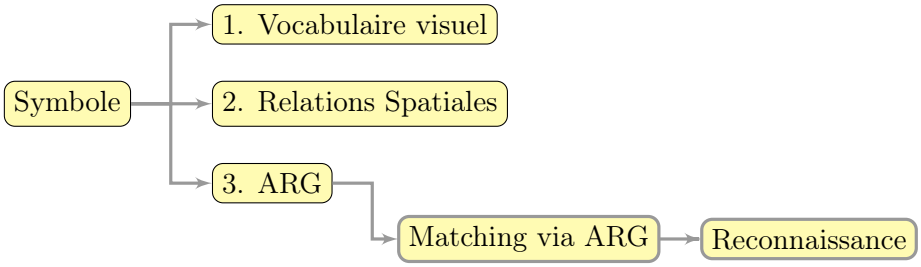


Figure 10: Une architecture pour la description et la reconnaissance de symboles en utilisant des relations spatiales. Elle utilise une description de symbole basée sur ARG en utilisant un vocabulaire visuel et des relations deux-à-deux possibles. Ces relations sont utilisées pour l'appariement.

électriques. Il peut être facilement étendu pour s'adapter à d'autres domaines. Ces visuels sont extraits des pièces élémentaires avec l'aide d'opérations de traitement d'image tels que décrits dans l'analyse [Rendek et al., 2004]. Plutot que d'utiliser tous les éléments détectés en tant que base pour exprimer les relations spatiales et de l'informatique, nous les regroupons par type comme indiqué dans la Figure 4.15. Nous noterons l'ensemble de ces produits groupes, $\sum_{\mathbb{T}} = \{\mathbb{T}_{thick}, \mathbb{T}_{circle}, \mathbb{T}_{corner}, \mathbb{T}_{extremity}\}$.

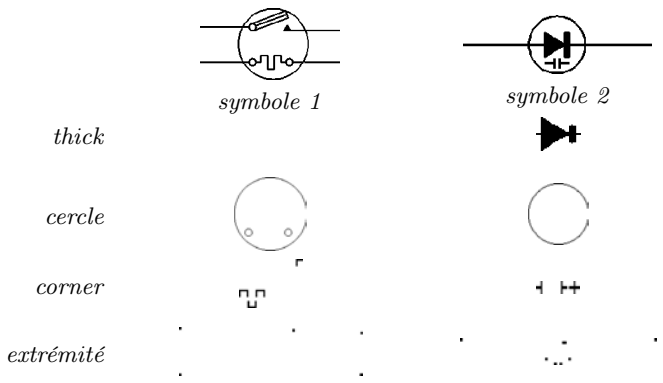


Figure 11: Illustration du type de vocabulaire.

étape 2 Nous calculons ensuite les relations spatiales entre les paires possibles de types de vocabulaire étiquetés qui sont utilisées comme une base pour construire un graphe attribué relationnel (ARG) qui décrit parfaitement le symbole.

Formellement, nous définiossons l'ARG comme un quardruple $G = (V, E, F_A, F_E)$ où

V est l'ensemble des nœuds;

$E \subseteq V \times V$ est l'ensemble des arrêtes;

$F_A : V \to A_V$ est une fonction qui affecte des attributs étiquetés nœuds et où A_V est l'ensemble des attributs $\sum_{\mathbb{T}}$ (*cf.* point 1) et

$F_E : E \to \Re_E$ est une fonction qui affecte des étiquettes aux arrêtes et où $\Re$ represente la relation spatiale entre les nœuds aux extrémités de l'arrête E.

Par exemple, en utilisant le symbole 1 dans la in Figure 4.15 comme exemple, sa représentation correspondante ARG est montrée dans la Figure 4.17.

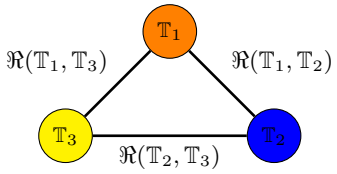


Figure 12: Graphe Attribué Relationnel (ARG) pour le symbole 1 dans la Figure 4.15 – un example.

Les modèles relationnelles existantes abordent seulement soit les relations topologiques, soit directionnelles. La gestion des deux vient à des coûts élevés de calcul. Même alors, aucun modèle existant n'intègre pleinement a topologie. Ils ont plutôt divers degrés de sensibilité à la prise de conscience des relations topologiques. Alors que des méthodes comme [Xiaogang et al., 2004] se focalisent sur des informations topologiques seulement, notre approche unifie les informations topologiques et directionnelles dans un descripteur [K.C. et al., 2009c] sans aucun frais temps de fonctionnement supplémentaire.

Notre modèle relationnel peut être résumé comme suit. D'abord nous cherchons le point de référence $\mathbb{R}_{p_c}$ à partir des configuration topologiques deux à deux des objets considérés. Ensuite des relations directionnelles sont calculées. Pour chaque point de référence $\mathbb{R}_{p_c}$, nous couvrons l'espace environnant à des intervalles radiaux réguliers de $\Theta = 2\pi/m$. Comme montré dans la Figure 4.7 (a), un rayon tourne autour du point de référence et lorsqu'il s'intersecte avec l'objet $\mathbb{X}$ ($\mathbb{A}$ or $\mathbb{B}$), genère ainsi un histogramme booléen $\mathcal{H}$,

$$\mathcal{H}(\mathbb{X}, \mathbb{R}_{p_c}) = [I(\mathbb{R}_{p_c}, j\Theta)]_{j=0..m-1}$$

où

$$I(\mathbb{R}_{p_c}, \theta_i) = \begin{cases} 1 & \text{if } line(\mathbb{R}_{p_c}, \theta_i) \cap \mathbb{X} \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Cet histogramme booléen exprime le fait qu'il y ait des pixels noirs dans une direction donnée θ_i. Cette notion d'histogramme peut être étendue sans perte de généralité à un histogramme couvrant des secteurs plutôt que des rayons, entre deux valeurs d'angles successifs : θ_i and θ_{i+1} et qui est normalisé par rapport à la surface de l'objet couvert de sorte à ce que $\sum \mathcal{H}(.) = 1$.

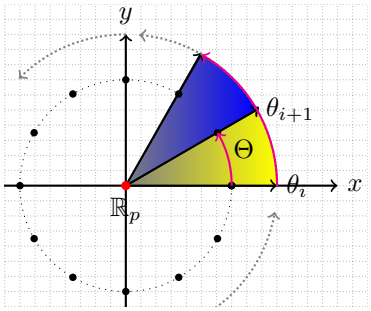
Globalement, nous évitons le problème général NP-difficile de mise en correspondance de sous-graphes en ayant des nœuds étiquetés uniques. Notre mise en correspondace d'ARG se fait donc en temps quasi-constant. Contrairement à des méthodes actuelles, notre méthode est donc en mesure de récupérer des parties isolées, ainsi que des pièces connues ou importantes des symboles de ligne intégrées dans d'autres symboles, composites.

Validation

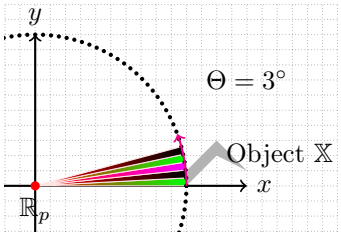
Pour valider notre méthode, nous menons une série d'expériences sur les données FRESH [FRESH, 2007] et la comparons avec l'état de l'art des algorithmes.

Notre méthode de relations spatiales est comparée avec d'autres méthodes :

1. centres basé sur l'angle discrétisé [Miyajima and Ralescu, 1994],
2. histogramme d'angle [Wang and Keller, 1999] and



(a) Rotation de ligne radiale ⟳.



Histogramme à $\Theta = 3°$ resolution:

$\texttt{boolean}\ \mathcal{H}(\mathbb{X}, \mathbb{R}_p) = \texttt{[1 1 1 1 1 0 0 0 ... 0]}_{1\times 120}$
$\texttt{metric}\ \mathcal{H}(\mathbb{X}, \mathbb{R}_p) = \texttt{[0.14 0.21 0.24 0.28 0.12 0 0 0 ... 0]}_{1\times 120}$

(b) Un exemple de calcul d'histogrammes: booléen et en pourcentage.

Figure 13: Calcul des relations spatiales utilisant la rotation de ligne radiale.

3. rectangle minimal de délimitation (MBR) [Papadias and Theodoridis, 1997].

Ensuite, nous avons effectué une autre évaluation afin de rendre la comparaison de la méthode complète avec les approches de l'état de l'art (cf. Section 2.1)). Pour cela, nous prenons un représentant de quelques descripteurs de signal globaux, appliqués directement sur le symbole. Ils sont basés région

1. moments de Zernike (ZM) [Kim and Kim, 2000],
2. le descripteur générique de Fourier (GFD) [Zhang and Lu, 2002],
3. contexte de forme (SC) [Belongie et al., 2002] and
4. $\mathcal{R}-$signature [Tabbone et al., 2006].

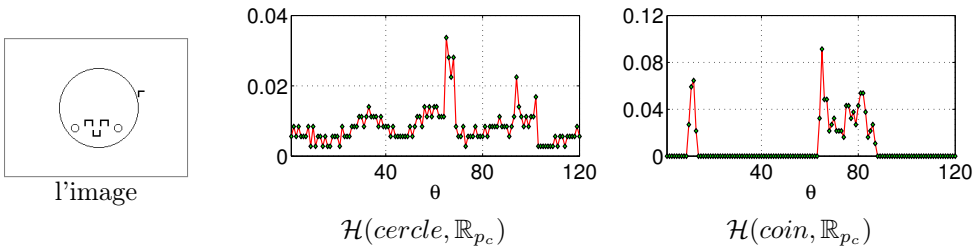


Figure 14: Histigrammes relationels pour une paire de types de vocabulaire: *cercle* et *coin*, pris du *symbole 1*, Figure 4.15.

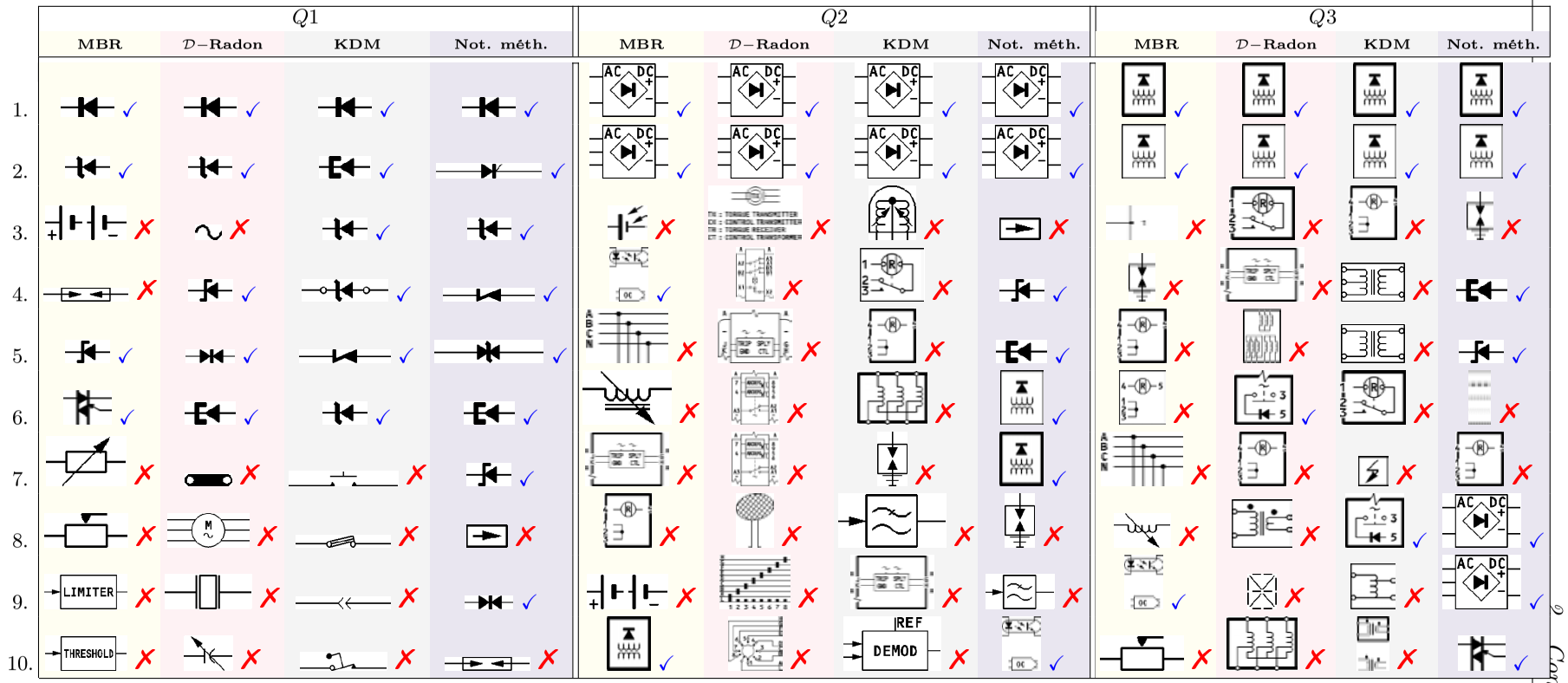


Figure 15: Illustration visuelle de la récupération de symboles et de classement pour quelques requêtes, montrant la récupération vrai (✓) et le faux (✗).

Cette partie de l'expérience est tirée de la contribution présentée àdans la Section 2.1. Enfin, des approches récentes fondées sur la reconnaissance de pixels et dédiées aux symboles sont comparées également. Elles sont

1. intégration de la statistique du tableau d'histogramme (SIHA) [Yang, 2005] et
2. noyau de la densité $2D$ [Zhang et al., 2006] une représentation basée symbole.

Sommaire des résultats. Afin de comparer notre modèle de relations spatiales avec les autres, nous avons adapté nos ARG pour fonctionner avec ces modèles fondamentaux susmentionnés. Parmi eux, le MBR surclasse dans toutes les situations. Dans le deuxième cadre, GFD semble être le meilleur parmi tous les descripteurs de signaux testés dans notre contexte. Par conséquent, pour une comparaison rapide, nous avons pris le meilleur des deux classes: MBR pour les relations spatiales de base, GFD à partir des descripteurs du signal global et KDM pour les approches pixel approches.

Notre méthode surpasse tous les modèles de relation spatiale ainsi que l'état de l'art des approches. Notre méthode surpasse à la fois avec une différence significative dans l'efficacité de récupération. La différence approximative avec GFD est de plus de 15% et 30% avec le modèle MBR. KDM et GFD montrent un comportement similaire. La Figure 4.27 montre la démonstration visuelle pour la comparaison.

2.3 Intégration des Signatures de Forme dans la Représentation ARG pour la Reconnaissance de Symboles

Contexte

Les approches structurelles fournissent des représentations puissantes, et expriment comment les pièces sont connectés les uns aux autres. Toutefois, les signatures relationnelles n'exploitent pas l'information ausi finement que des descripteurs de forme peuvent le faire. D'autre part, les descripteurs de signaux globaux forme basée ne peut pas fournir de récupération optimale performances puisque les symboles ne sont pas seulement sous forme linéaire et isolés, ils sont composés avec des de nombreux éléments à la place. Par conséquent, la sélection optimale des caractéristiques de forme et où s'appliquent sont les deux tâches principales dans le cadre ARG comme discuté auparavant.

Considérant le problème de la localisation de symboles dans des documents réels, composé de différents pièces contraints par des relations spatiales, on doit être capable de formaliser les relations qui existent entre les parties extraites visuelle en plus de la description de formes individuelles. Ce l'intégration des relations spatiales et la description la forme des pièces extraites visuelle va être la partie essentielle de cette contribution. Cela signifie que nous intégrer efficacement structurelles et caractéristiques statistiques dans nos méthodes.

Par conséquent, une description de l'image appropriée est nécessaire pour les intégrer où il représente les propriétés généralité et l'extensibilité des approches structurelles. Il a également été clairement mentionnés dans [Tombre, 2010]:

> «... la nature très structurelle et spatiale de l'information que nous fait travailler avec méthodes structurelles tout à fait naturel dans la communauté. Leur intégration efficace dans méthodes qui ont également de profiter pleinement de l'apprentissage statistique et de classification est certainement le droit chemin à prendre.»

Un exemple intéressant qui utilise des descriptions de forme et de relations pour former adjacence de régions Graphique (GCR) est trouvée dans [Bodic et al., 2009]. Le GCR vectoriel est basé sur les régions segmentées qui sont étiquetés comme des sommets et les propriétés géométriques d'une relations d'adjacence sont utilisées pour bords des étiquettes. L'approche est limitée une fois les régions segmentées changer avec les transformations d'image et donc il utilise les classes du modèle de localiser quelques symboles dans les documents techniques. En revanche, notre intégration est différente de la façon dont nous appliquons les descripteurs de forme et l'utilisation de données relationnelles signatures. La méthode proposée sera discutée dans la section suivante.

Méthode proposée

Dans cette contribution, nous visons à combiner le meilleur des deux mondes: structurelle et statistique, et essayez d'éviter les défauts de chacun d'eux. Pour ce faire, nous décomposons en exprimant des symboles leurs diverses parties dans un vocabulaire fixe visuelle, en utilisant les relations spatiales, de graphiques et de signal basé descripteurs pour décrire la forme entière.

Globalement, nous nous concentrons sur deux contributions majeures. Ils sont:

1. signatures de vertex à travers les descriptions forme dans ARG.

2. regroupement vocabulaire par l'analyse de forme - elle est appliquée à des modèles d'épaisseur.

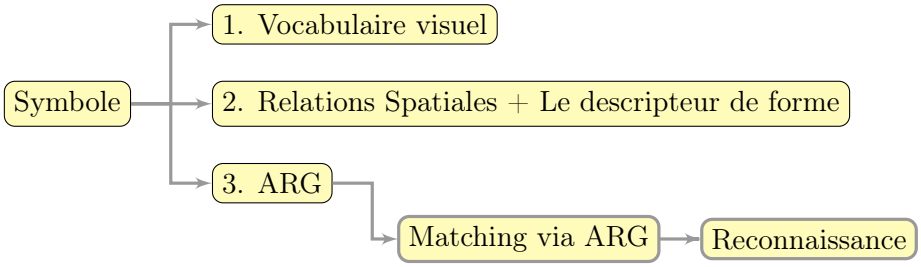


Figure 16: Une architecture pour la description des symboles et la reconnaissance utilisant des caractéristiques de forme et des relations spatiales. ElleIl utilise une description de symboles ARG en utilisant un vocabulaire visuel et de leurs possibles relations. Des signatures de forme sont utilisées pour discriminer les pièces visuelles. Les deux, relations comme les signatures de forme sont utilisées pour l'appariement des symboles principalement via ARG.

Validation

Puisqu'il s'agit ici d'une extension des travaux précédents mentionnés dans la Section 2.2, nous avons testé son efficacité sur les données FRESH [FRESH, 2007]. Dans tous les tests, nous prenons un représentant de quelques descripteurs de signal globaux, appliqués directement sur le symbole, tout en intégrant des relations spatiales. Ils sont

1. moments de Zernike (ZM) [Kim and Kim, 2000],
2. le descripteurs générique de Fourier (GFD) [Zhang and Lu, 2002],
3. contexte de forme (SC) [Belongie et al., 2002],
4. $\mathcal{R}-$*signature* [Tabbone et al., 2006] and
5. $\mathcal{D}-$Radon (voir la Section 2.1).

Sommaire des résultats. Globalement, nous présentons une méthode de reconnaissance de symboles sur la base des similitudes entre l'organisation spatiale et les caractéristiques de forme des pièces visuelles qui composent le symbole.

Les différences substantielles en taux de reconnaissance entre les signatures forme intégrés dans les sommets des l'ARG par rapport aux signatures de formes globaux, fournit des éléments permettant d'affirmer que ces ajouts de description du symbole sont très importantes. Toutefois, elle est mathématiquement coûteuse car elle nécessite des t signatures différentes pour le nombre t de types de vocabulaire. Une telle signature pour les sommets est alors intelligemment intégré à la signature des arrêtes. Tout en les intégrant, nous avons obtenu de meilleurs résultats en utilisant seulement des méthodes de *clustering* des signatures de formes sur les sommets étiquetés avec des traits épais épaisseur ou des cercles. Par conséquent, elle réduit le coût correspondant des signatures correspondantes entre les sommets. Avec la mise en correspondance de signatures relationnelles, une requête de sélection du motif d'épaisseur via le *clustering* a grandement amélioré la performance. Cependant, la méthode ne fournit pas une différence surprenante en raison du fait que tous les symboles de requête contiennent profil de type trait d'épaisseur" dans leurs ensembles de type de vocabulaire. En d'autres termes, l'absence d'un vocabulaire de type "trait d'épaisseur" signifie que le classement a été fait par l'alignement des signatures relationnelle seulement, ne changeant finalement pas la performance de récupération.

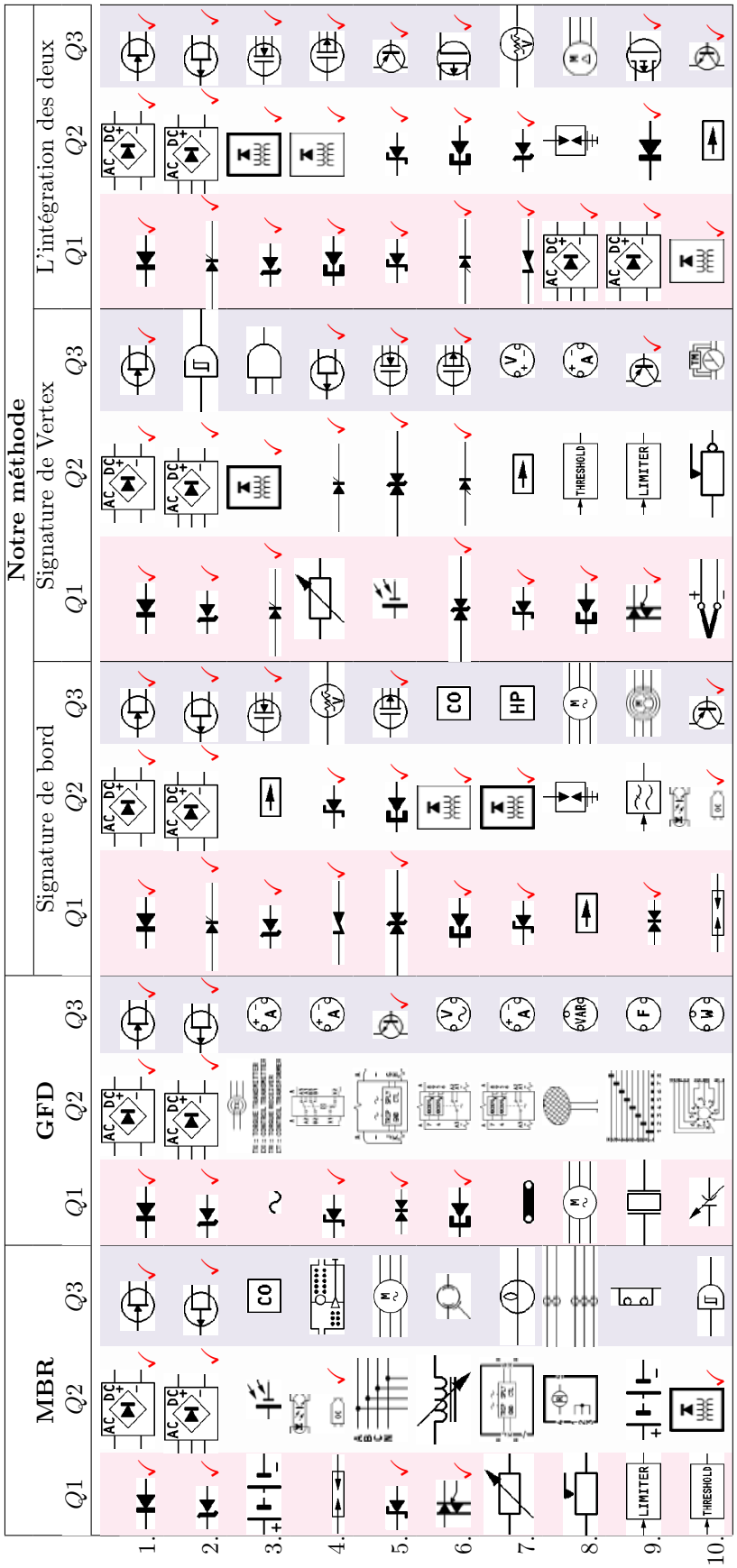


Figure 17: Illustration visuelle de classement symbole à la sortie pour quelques requêtes: $Q1$, $Q2$ et $Q3$, montrant✓ pour la récupération vrai et le faux, sinon. Le premier symbole sur le dessus correspond toujours à la requête choisie. Les symboles sont classés de haut en bas repose sur l'ordre décroissant de similarité.

Deux descripteurs de signal globaux: $\mathcal{D}-$Radon et GFD montrent un comportement presque similaire sauf une différence de temps d'exécution, tandis que d'autres n'ont pas vraiment élevé les performances. SC et ZM sont inférieurs à $\mathcal{D}-$Radon et GFD. Dans notre application, SC est fortement limitée par le nombre d'échantillons. Toutefois, nous avons essayé de le maximiser en choisissant différemment dans différents sommets, par exemple. $\mathcal{R}-$signature se trouve à être toujours à la traîne.

Pour une meilleure compréhension, la Figure 17 montre une comparaison des performances de récupération pour les quelques requêtes.

2.4 Sac de Caractéristiques Spatiales pour la Reconnaissance de Symboles

Contexte

L'appariement de relations spatiales entre le vocabulaire qui compose un symbole est examinée dans Section 2.2. Dans la description ARG correspondante, les sommets représentés contiennent des éléments de types de vocabulaire regroupés plutôt que d'utiliser tous les pièces élémentaires individuelles détectées. Cela a permis d'éviter les promblèmes NP-difficiles de mise en correspondance de sous-graphes. Une telle description est limitée aux seuls symboles qui ont au moins deux types de vocabulaire. Le problème est que ceci n'est pas toujours le cas. Prenons la situation suivante à l'aide d'une illustration. Dans la Figure 18, deux types vocabulaire différents (cercle et angle) sont détectés, et sont utilisés pour représenter deux sommets différents dans notre description symbole ARG. En l'absence de cercle ou pour représenter un rectangle uniquement via ARG il n'existe pas de relation.

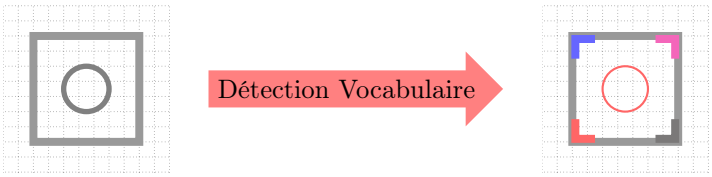


Figure 18: Un symbole et ses parties élémentaires: le cercle et quatre coins.

Pour gérer une telle situation, une solution immédiate est d'utiliser le type de relations intra-spatiale, qui a encore besoin d'avoir au moins deux pièces visuelles élémentaires dans un type de vocabulaire unique. Par conséquent, nous avons intégré la description de la forme des sommets que nous avons examiné dans la Section 2.3. L'intégration des signatures de forme avec les signatures relationnelles donne une description ARG complet du symbole. En outre, les signatures de forme peuvent être modifiées pour une certaine mesure, si une des parties du vocabulaire est oubliée lors de l'extraction. La partie essentielle de cette contribution sera d'utiliser une approche de type sac-de-Caractéristiques (BOF) utilisant les relations spatiales pour toutes les combinaisons possibles des parties visuelles.

Prenons une définition rapide du modèle BOF avec un exemple. Une image peut être traitée comme un document, et des caractéristiques extraites de l'image sont considérées comme les 'mots visuels'. Il permet une modélisation basé sur des dictionnaires, et chaque document ressemble à un 'sac', qui contient quelques des mots du dictionnaire. En général, l'extraction de plusieurs points–clés locales (ou régions) sont considérés comme des éléments de base, les 'mots'. À titre d'exemple, les méthodes telles que la Hesse-Affine descripteur [Mikolajczyk and Schmid, 2004] et EIPD fonctionnalités [Lowe, 2004] ont été les plus utilisés. Par ailleurs, la grille régulière basée sur la méthode comme [Li et al., 2005], est une autre idée simple pour détecter des caractéristiques.

Dans la reconnaissancees de graphiques, la démarche Sac de Caractéristiques Spatiales n'existe pas vraiment dans la littérature. Des chercheurs de vision par ordinateur utilisent une idée similaire pour la représentation de l'image. Dans la récente littérature, Sac de Caractéristiques Spatiales a été étudié pour coder les informations géométriques d'objets et pour l'utiliser dans la récupération d'images efficace [Cao et al., 2010]. Il utilise la projection de caractéristiques locales dans un certain nombre de baquets. Ces baquets ou urnes peuvent être une projection linéaire de 10 angles équidistants de $[0°, 180°]$, ou le partitionnement les images en plus fines sous-régions et les histogrammes des caractéristiques locales [Lazebnik et al., 2006], par exemple.

Inspiré par le concept ci-dessus, nous l'introduisons dans la reconnaissance des graphismes

une approche différente. Dans notre contexte, les relations directionnelles entre paires sont considérées comme les candidats des 'caractéristiques'. Nous avons d'abord classé les pièces élémentaires visuelles sur la base des relations topologiques. Pour l'illustrer, la Figure 19 montre quelques catégories topologiques. Ensuite nous calculons les relations directionnelles pour donner des détails métriques entre eux. Tout en faisant la reconnaissance, les relations directionnelles correspondent a lieu entre les candidats qui partagent les mêmes configurations topologiques. Une telle utilisation d'indications topologiques réduit considérablement le temps correspondant.

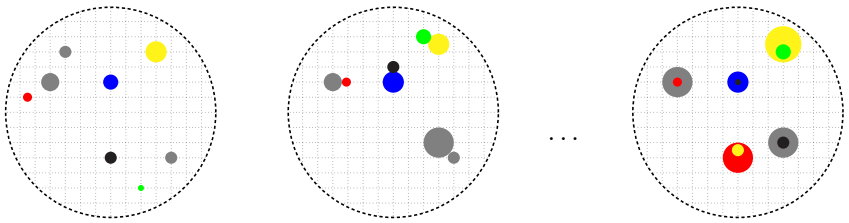


Figure 19: Modèle de Sac de Caractéristiques Spatiales – un exemple. Chaque élément représente une primitive visuelle.

Méthode proposée

Notre méthode est basée sur BOF qui utilise les relations spatiales entre les parties visuelles élémentaires extraites qui composent un symbole. Dans la Section 2.2, nous expliquons le vocabulaire visuel et la manière pour calculer les relations spatiales. Dans ce qui suit, nous allons vous expliquer comment nous les traitons dans le modèle BOF.

Notre démarche sac de caractéristiques (BOF) gère toutes les combinaisons possibles entre parties visuelles élémentaires. Les principales caractéristiques du processus global sont double:

1. Catégorisation du vocabulaire et
2. Raisonnement spatial.

Nous avons d'abord classé les éléments visuels dans des paires, basées sur des relations topologiques (*cf.* La Figure 19). Nous calculons ensuite les relations directionnelles entre elles. Cela signifie que les relations directionnelles sont calculées en fonction de leurs relations topologiques booléennes. Par conséquent, les relations directionnelles sont mises en correspondance uniquement avec ceux qui partagent les mêmes configurations topologiques. L'ensemble du processus traite le vocabulaire de type intra-et inter-type séparément. La Figure 20 montre un bloc–diagramme pour illustrer le concept de la méthode proposée.

Validation

Pour valider, en plus des images FRESh [FRESH, 2007], nous avons pris GREC [GREC, 2003]. Pour la comparaison, comme dans les sections précédentes, nous effectuons une série de tests expérimentaux afin de confronter nos méthodes proposées avec celles existantes. Nos tests consistent essentiellement en deux parties.

1. Méthode sac de caractéristiques spatiales

   il est important de rappeler que la méthode du sac de caractéristiques spatiales utilise les modèles directionnels suivants:

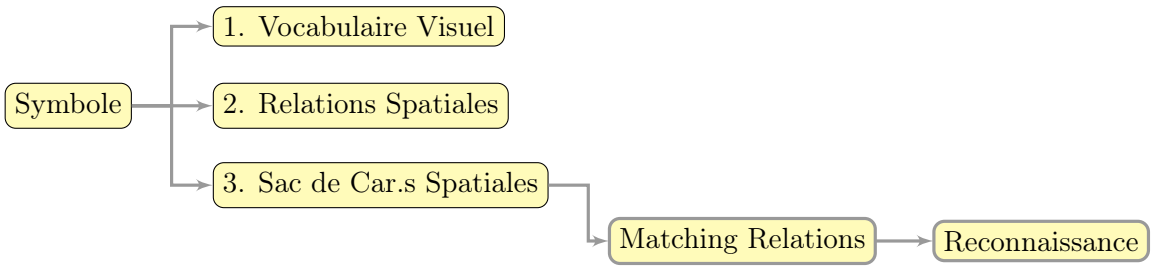


Figure 20: Une architecture de la méthode de reconnaissance des symboles basés sur Sac de Caractéristiques Spatiales.

(a) modèle de projection,
(b) modèle de la ligne radiale and
(c) Des modèles basiques
- centres basé sur l'angle discrétisé [Miyajima and Ralescu, 1994],
- histogramme d'angle [Wang and Keller, 1999] and
- rectangle minimal de délimitation (MBR) [Papadias and Theodoridis, 1997].

Comme mentionné dans la section 2.2, nous répétons l'utilisation de tous les modèles de relations spatiales. La différence vient principalement de la façon dont ils sont utilisés. Dans la section 2.2, les relations sont utilisées entre types de vocabulaire, tandis qu'ici elles sont utilisées entre chaque partie visuelle individuellement.

De façon similaire, nous avons également effectué des évaluations en utilisant l'ensemble des descripteurs globaux déjà utilisé précédemment:

2. Les descritpeurs globaux sont :

(a) moments de Zernike (ZM) [Kim and Kim, 2000],
(b) le descripteur générique de Fourier (GFD) [Zhang and Lu, 2002],
(c) contexte de forme (SC) [Belongie et al., 2002],
(d) $\mathcal{R}-$*signature* [Tabbone et al., 2006] and
(e) $\mathcal{D}-$Radon (voir la Section 2.1).

Comme dans les sections 2.2 et 2.3, nous répétons les expériences afin de pouvoir établir une comparaison point par point.

Sommaire des résultats. Comme mentionné précédemment, nous avons testédeux ensembles de données différents: FRESH [FRESH, 2007] et GREC [GREC, 2003]. .

Dans le cas de données GREC, il ne donne pas de taux de reconnaissance élevé. Les descripteurs de forme donnent de meilleurs résultats car l'ensemble est composé de dessin des symboles de lignes. L'amélioration des performances de récupération de la méthode est clairement visible dans le cas des données FRESH. Cela signifie que la méthode peut effectivement récupérer des symboles qui se trouvent dans une forme composite. Globalement, l'espace-sac-de- méthode de fonctionnalités fournit des résultats intéressants, pour tous les modèles spatiaux relationnels. En outre, la méthode peut être utilisée comme une approche « conviviale » de demande de récupération symboles parce nous pouvons choisir les caractéristiques visuelles et l'utilisation de leur organisation spatiale pour faire une requête. Par exemple :

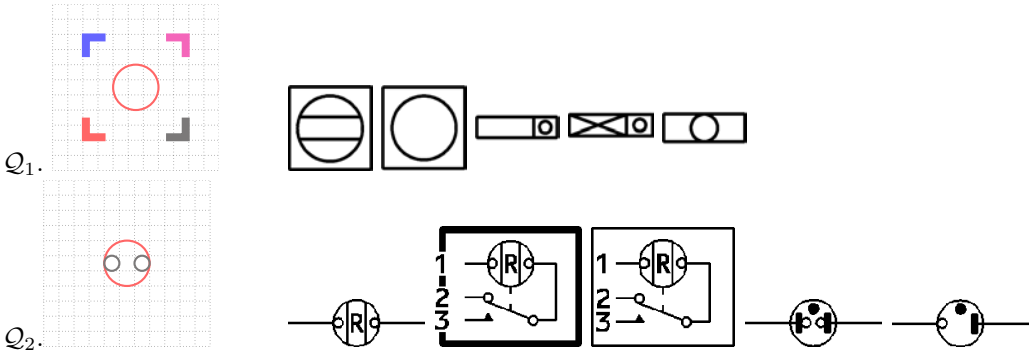


Figure 21: Récupération de symboles pour deux ensembles de symboles: GREC et FRESH basée sur l'organisation spatiale d'un ensemble de parties visuelles comme requête.

- Trouver le symbole qui contient une épaisseur primitive intérieur d'un cercle primitif.
- Trouver le symbole qui a un rectangle contenant un cercle.

De telles requêtes récupèrent tous les symboles du jeu de données ayant une organisation spatiale similaires. Figure 21 montre quelques exemples. Ces exemples attestent des avantages ainsi que la généralité de la méthode.

2.5 Apprentsissage de Symboles par programmation logique inductive basée sur des descriptions formelles

Contexte

L'objectif principal de l'analyse d'image est d'éventuellement trouver le moyen de combler le fossé sémantique entre les descriptions de bas niveau de l'image et les concepts de ce qui est présenté en son sein. L'hypothèse courante dans l'état de l'art est que cela peut être obtenue par l'interaction adéquate entre les descriptions des formes et la comparaison ou la distance entre eux d'une part, et les techniques de classification ou le regroupement pour associer ces descriptions à des concepts de plus haut niveau, d'autre part.

Tenter d'exprimer les informations visuelles à l'aide de descriptions «naturelles» a été l'origine qui sous-tend l'engouement derrière l'analyse de modèles structurels. Le plus souvent, cela se fait d'abord en lisant le faible niveau des indices visuels qui forment la base de données lexicales, et en procédant par un certain regroupement algorithmique afin d'exprimer des relations ou des propriétés qui sont ensuite traduits dans des «vocabulaires» de plus et plus complexes qui déclenchenet des règles de niveau supérieur, pour, à la fin, exprimer des concepts terminaux [Mas Romeu et al., 2007].

Notre approche est légèrement différente dans la façon dont nous n'essayons pas de construire une chaîne de règles syntaxiques de déclenchement, mais plutôt construire notre vocabulaire sur l'extraction directe de structures (plus ou moins) complexes dans les images. Ce vocabulaire caractérise les symboles par un ensemble de structures locales très robustes. Cela nous permet d'exprimer des descriptions des symboles en prédicats de logique du premier ordre (FOL), exprimant leur type selon le vocabulaire et l'expression du positionnement relatif des uns aux autres.

Méthode proposée

Nous utilisons la programmation logique inductive (PLI) pour apprendre automatiquement des descriptions non triviales de symboles, basées sur une description formelle. Figure 22 montre une architecture de la méthode.

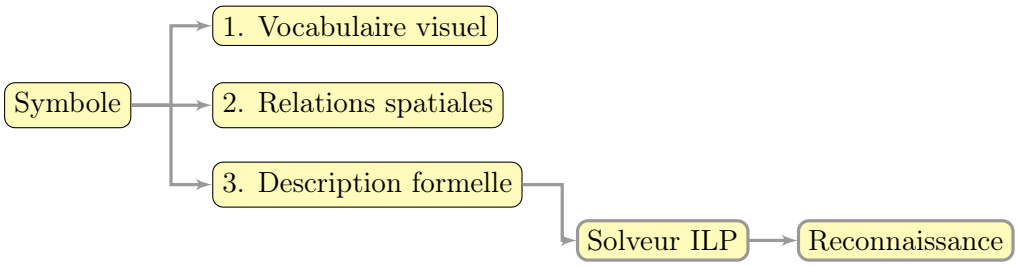


Figure 22: Une architecture pour la reconnaissance de symboles sémantiques liées à l'aide logique inductive Programmation basée sur la description formelle.

L'objectif global de la démarche consiste à exprimer des symboles graphiques par un certain nombre de primitives visuelles qui peuvent être de n'importe quelle complexité (ie, pas nécessairement juste des lignes ou points) et les relations qui peuvent être déduites par traitement d'image simple utisisant des outils d'analyse de l'état de l'art. Cette représentation est ensuite utilisée comme une entrée pour un solveur ILP, afin d'en déduire des caractéristiques non évidentes qui peuvent conduire à un processus de reconnaissance plus sémantique.

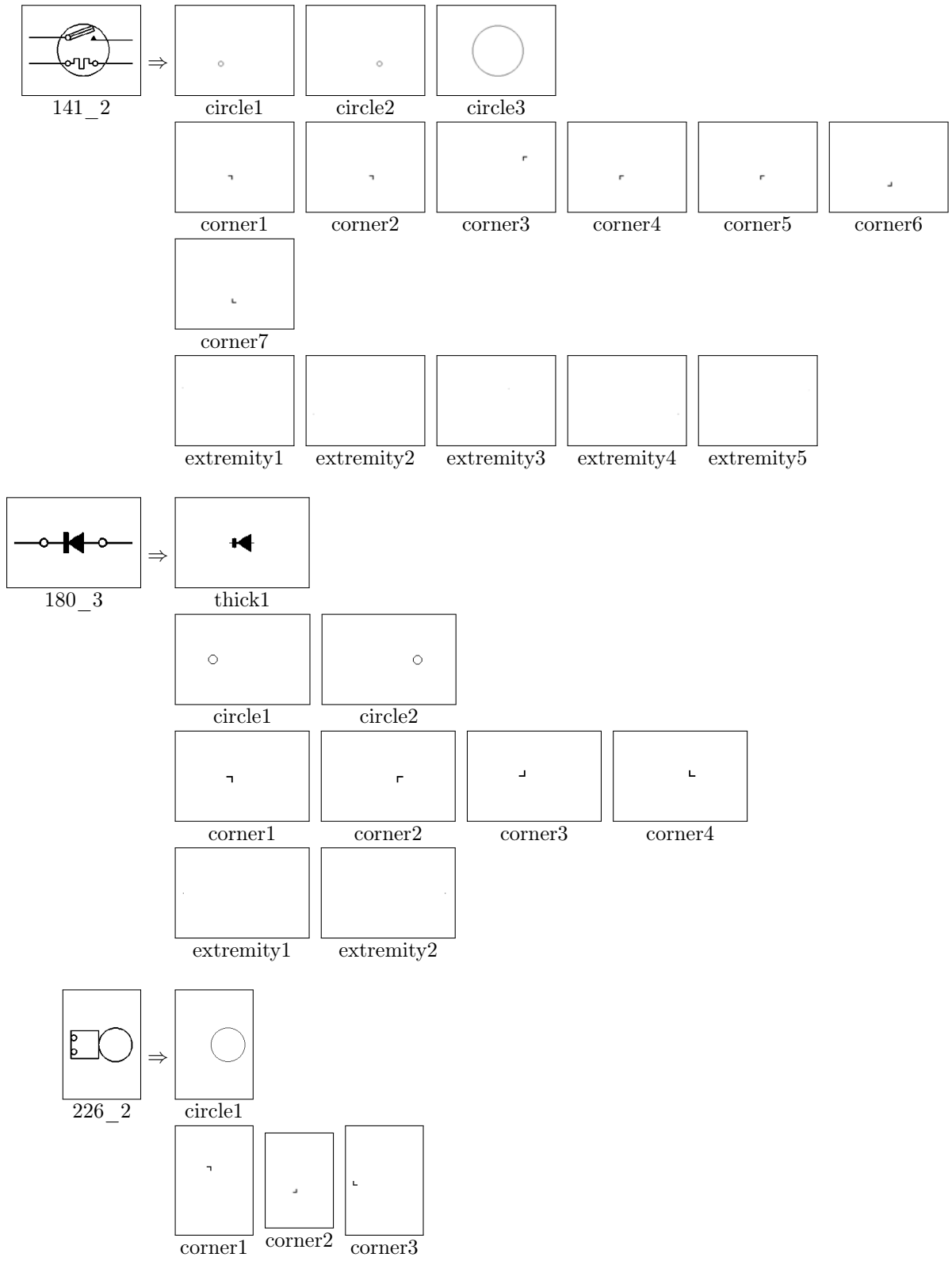


Figure 23: Primitives visuelles ou types de vocabulaire: *thick*, *circle*, *corner* and *extremity* pour les symboles correspondants.

Laissez-nous vous expliquer comment nous décrivons un symbole. La Figure 23 montre le nombre de primitives visuelles détectées avec leur organisation spatiale. Basés sur elles, la description symbole à l'aide d'un vocabulaire FOL sera expliquée ensuite. Prenons le symbole 226_2 de la Figure 23 pour voir comment nous l'exprimons en FOL.

```
%    symol 226_2**********************************
type(prim_174,circle).
type(prim_175,cornerne).
type(prim_176,cornerse).
type(prim_177,cornersw).

has_element(img_226_2,prim_174).
has_element(img_226_2,prim_175).
has_element(img_226_2,prim_176).
has_element(img_226_2,prim_177).

e(prim_174,prim_175). e(prim_174,prim_176).
inside(prim_174,prim_177). w(prim_175,prim_174).
n(prim_175,prim_176). nw(prim_175,prim_177).
w(prim_176,prim_174). s(prim_176,prim_175).
w(prim_176,prim_177). inside(prim_177,prim_174).
se(prim_177,prim_175). e(prim_177,prim_176).
%    fin 226_2**********************************
```

Basé sur la description décrite précédemment, les quatre premières lignes affectent les types aux primitives visuelles extraites du symbole, notamment

```
type(prim_XX, visual_primitive).
```

Ensuite, les quatre lignes suivantes, défissent l'image dont elles sont originaires (`img_name`)

```
has_element(img_name,prim_XX).
```

Les six dernières lignes fournissent les relations deux à deux entre les primitives :

```
nw(prim_XX,prim_YY)
```

par exemple, `prim_XX` se trouve au *nord-ouest de* `prim_YY`.

Validation

Comme avant, pour validation, nous utilisons des données expérimentales issues de FRESH [FRESH, 2007]. Dans ce qui suit, nous expliquons notre démarche pour l'apprentissage des propriétés communes à partir de symboles choisis afin d'exprimer des connaissances non triviales de représentations visuelles basées sur le concept sémantique.

Afin de montrer quel genre de données que nous avons fait manipuler, nous avons sélectionné des symboles 225_2 226_2 de la Figure 24 comme des exemples positifs. Tous les autres sont considérés comme des contre-exemples.

La sortie du solveur ILP se compose d'une section `[théorie]`, contenant les règles qui définissent l'exemple positif donné. Pour chaque règle de la théorie, le solveur donne des statistiques correspondant, indiquant la précision des règles (combien d'exemples positifs couverts, et combien négatives des exemples). Pour un match parfait, la partie théorique devrait consister en une règle unique couvrant tous les exemples positifs et aucun des exemples négatifs. Des expériences

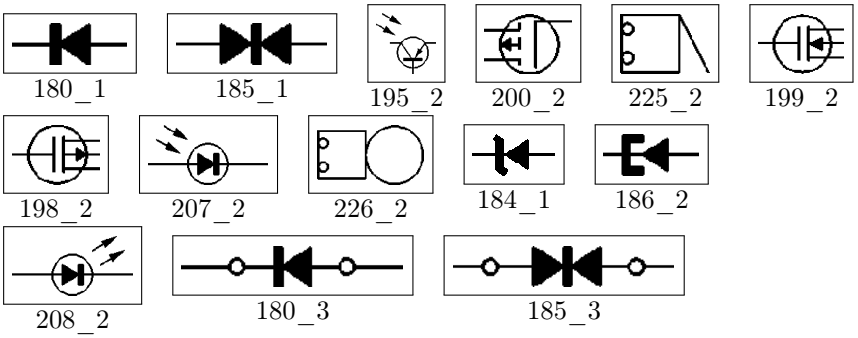


Figure 24: Un petit échantillon utilisé pour les expérimentations.

complémentaires devront montrer que ce n'est pas toujours réalisable. Parfois, la théorie est composé de plusieurs règles, dont chacun couvre un sous-ensemble des exemples positifs. Parfois, des exemples négatifs sont aussi couverts par la théorie. Dans notre exemple, cela donne:

```
[theory]
[Rule 1] [Pos cover = 2 Neg cover = 0]
symbol(A):-
    has_element(A,B), type(B,cornerne),
    has_element(A,C), n(B,C), type(C,cornerse).

[positive examples covered]
    symbol(img_225_2).
    symbol(img_226_2).
[negative examples covered]

test
[covered]
symbol(img_225_2):-
     has_element(img_225_2,prim_170),
     type(prim_170,cornerne),
     has_element(img_225_2,prim_172),
     n(prim_170,prim_172),type(prim_172,cornerse).
```

Après la section `[theory]` viennent deux sections décrivant les examples couverts par la théorie : `[positive examples covered]` ainsi que les exemples négatifs éventuellement couverts par erreur : `[negative examples covered]`. Ces deux sections explicitent les instances des exemples couverts par l'ensemble des règles.

La dernière partie `[covered]` est tout simplement un exemple d'une des instances couvertes en guise d'exemple de vérification.

L'interprétation complète de la sortie du solveur est que les symboles 225_2 and 226_2 peuvent être distingués complètement et formellement des autres symboles par le fait qu'ils disposent de deux coins alignés verticalement comme montré ci-dessous :

- ⌝ nord-est coin (*i.e.*, `cornerne`)
- ⌟ sud-est coin (*i.e.*, `cornerse`)

Une vérification visuelle de la Figure C.8 ne permet pas de trouver de contre-examples, et une vérification plus formelle le confirme.

Sommaire des résultats. La manipulation des pièces visuelles élémentaires de symboles graphiques, plutôt que d'en utiliser la forme générale, a plusieurs avantages. Les avantages peuvent se résumer en deux principaux points:

1. L'avantage principal est que les symboles peuvent être localisés dans une forme composite.
2. La description fournit une représentation puissante des symboles à l'aide des relations entre ses composants.

Un exemple de déduire la caractérisation commune de tous les symboles:

```
symbol(A):- has(A, primitive_a), type(primitive_a, corner1),
            has(A, primitive_b), type(primitive_b, corner2),
            North(primitive_a,primitive_b).
```

En particulier, nous soulignons la méthode en deux points majeurs.

1. Le principal avantage de cette approche est que l'information ne doit pas nécessairement être visuellement représentée (par exemple, du texte environnant), et elle ouvre ainsi un nouveau champ de possibles combinée texte/image pour la caractérisation et la notion d'apprentissage.
2. Par rapport à des modèles statistiques d'apprentissage, le système s'adapte à la complexité de la classification à l'égard des données d'apprentissage, sans besoin d'aucun paramétrage de toutes sortes.

End of Thesis Summary in French

Part II

Complete Thesis in English

1 Introduction

Contents

1.1 Organisation of the Chapter 43

1.2 Graphics Recognition 44

1.2.1 Motivation 44

1.2.2 Scientific Context 46

1.2.3 Framework – Contributions of the Thesis 47

1.3 What is Next? 49

Foreword. This thesis is about graphics recognition i.e., the recognition of graphical symbols. Symbol recognition does not mean only to recognise isolated symbols but also those found in composite form, or connected with other graphical elements in addition to texts. Therefore, the thesis aims to retrieve those symbols which are similar either in terms of the whole appearance or of significant parts of it.

The experiments supporting our work primarily use electrical symbols from wiring diagrams. These diagrams are from an unpublished dataset called the FRESH dataset previously used in [Salmon, 2008]. The approaches developed in this thesis are also validated on other datasets such as architectural symbols, hand-drawn symbols, handwritten character datasets for several different scripts such as Indian, Japanese–Katakana and Roman, as well as a foot-wear impression evidence dataset. ■

1.1 Organisation of the Chapter

This chapter provides an overall concept of the thesis. We start with a positioning of graphics recognition within the larger document image analysis domain in Section 1.2. We also include motivation, scientific context and contributions dedicated to the thesis. We present a chapter-wise description of our contributions. In Section 1.3, we point out what comes next.

1.2 Graphics Recognition

1.2.1 Motivation

Document analysis or processing is mainly related to texts and graphics. It concerns separation, localisation and recognition. According to [Nagy, 2000], document analysis is related to document image analysis (DIA) since the overall research works have been concerned with document image interpretation. Basic tasks are image segmentation, layout understanding and symbol recognition. In a similar manner, [Kasturi et al., 2002] categorises document image analysis into two domains:

1. textual processing and
2. graphical processing.

Figure 1.1 shows a perspective of it along with a further sub-categorisation. Graphics processing refers to lines, corners and curves analysis on the one side while on the other side, it is concerned with regions and symbol analysis and recognition.

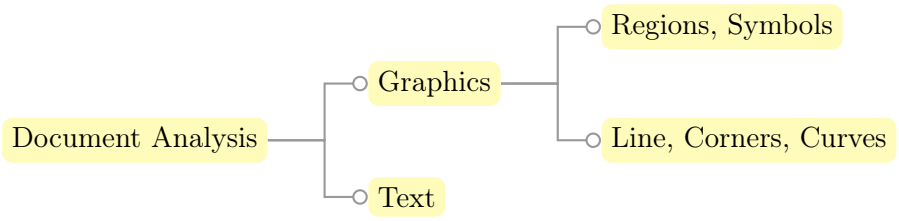


Figure 1.1: Document analysis or processing [Kasturi et al., 2002].

One part of document image analysis means to recognise graphical elements and or to localise significant or known visual part of the images. Within the aforementioned context, graphics recognition (i.e., recognition of graphical symbols) has been an intensive research activity in the community of pattern recognition and document analysis since a few decades. Symbol recognition has a long history since the 70's. In 1998, [Tombre, 1998] has made a statement: '*none of these methods works*'. Since then, it has been actively extended.

Symbol recognition – the core part of graphical document image analysis and recognition systems – plays an important role in a variety of applications such as automatic recognition and interpretation of

- circuit diagrams [Okazaki et al., 1988, Feng et al., 2009],
- engineering drawings [Yang et al., 2007] and architectural drawings [Lladós et al., 2001, Valveny and Martí, 2003],
- musical notations [Rebelo et al., 2010],
- maps [Samet and Soffer, 1996a],
- mathematical expressions [Chaudhuri and Garain, 2000], as well as
- optical characters [Yuen et al., 1998, Heutte et al., 2004].

Therefore, a symbol can be defined as a graphical entity with a particular meaning in the context of a specific domain. As such, and besides preprocessing, the whole symbol recognition process is either based on either

1. feature matching between a query and dataset symbols or

2. comparison of decomposed parts like *lines* and *arcs* as well as their relations between them.

Shape analysis for symbol representation is an example of the approaches, related to *item 1.* An interesting overview of the performance of the most commonly used shape descriptors for symbol representation is given in [Valveny et al., 2007]. They are commonly referred to as statistical approaches. On the other hand, *item 2* refers to structural approaches. They mainly include embedded graph based classification problems like attributed relational graphs (ARG), region adjacency graphs (RAG) [Lladós et al., 2001] and constraint networks [Ah-Soon and Tombre, 2001]. In both cases, recognition performance accounts mainly two issues: recognition rate or efficiency and computational complexity. The former issue is largely based on how symbols are represented i.e., quality of symbol representation (either just a contour information or global appearance is taken into account) and computational complexity is based on matching techniques. Also, in general, matching techniques are often induced how symbols are represented. Statistical signatures are found to be a better choice where linear or isolated symbol are treated and also when time complexity is an issue. Structural techniques are able to handle all types of symbols (e.g., isolated, composite), and have a powerful relational representation. However, they suffer from intense computational complexity due to the general NP-hard problem of sub-graph matching resulting from the variation of graph structure with the level of noise, occlusion, distortion *etc.*

In the current state-of-the-art, symbol recognition usually means recognising isolated symbols. This is not always suitable for solving real-world problems, however. Consider a situation where both isolated symbols as well as composite symbols that are connected with other graphical elements. In such a case, statistical signatures using shape descriptors, for instance, do not perform well because they take global appearance into account. Structural approaches do not perform well because their usability is limited due to intense computational complexity. Within this context, very recently, [Tombre, 2010] stressed the fact that '*which features distinguish graphics recognition from general pattern recognition problems?*' since it has been stated that methods vary from one application to another. Therefore, there still exists growing interest in graphics recognition domain.

Considering all aforementioned points, this thesis addresses the use of structural and statistical signatures as well as their possible and efficient combinations. Let us point out contemporary issues that have been made recently [Tombre, 2010], from which the thesis is inspired:

- *'... the recurring wish for methods capable of efficiently combining structural and statistical methods'* and
- '*the very structural and spatial nature of the information we work with makes structural methods quite natural in the community.*'

Based on the aforementioned points, efficient integration from which we can take full advantages of statistical and structural approaches, is the primary goal of the thesis. We present a method for symbol recognition based on the similarity between spatial organisation and shape features of the visual primitives that compose the symbol. We address the use of spatial relations between the visual primitives first and then combine them with shape features in the framework of ARG. Because of the particular choices in the labelling of the vertices representing known visual primitives, vertex and edge matching becomes trivial and can be done in a near-constant time, thus avoiding NP-hard graph matching problem. In parallel, we introduce a DTW–Radon descriptor for graphics recognition. The proposed methods provide better recognition performance compared to the major state-of-the-art algorithms. Furthermore, this thesis also addresses a

knowledge learning technique that aims for semantic related symbol recognition. It uses inductive logic programming (ILP) to automatically learn non-trivial descriptions of symbols, based on a formal description derived from the previously mentioned structural and statistical information. Such a description of the symbol is expressed by a number of visual elementary primitives and their possible relations. Compared to statistical learning models, the system adapts the complexity of the classification with respect to the learning data.

A detailed list of our contribution will be given in Section 1.2.3. Before that let us revisit and put forward our primary scientific problem in the following section.

1.2.2 Scientific Context

The work is inspired by a real world industrial problem [Tombre and Lamiroy, 2008, Salmon, 2008] which consists in identifying a set of known symbols in aircraft electrical wiring diagrams, in order to bootstrap simulation algorithms [FRESH, 2007]. The challenges primarily lie in the fact that the test symbols are in varied forms. Symbols may either be very similar in shape – and only differ by slight details – or either be completely different from a visual point of view. Symbols may also be composed of other known and significant symbols and need not necessary be connected. A few samples are shown in Figure 1.2.

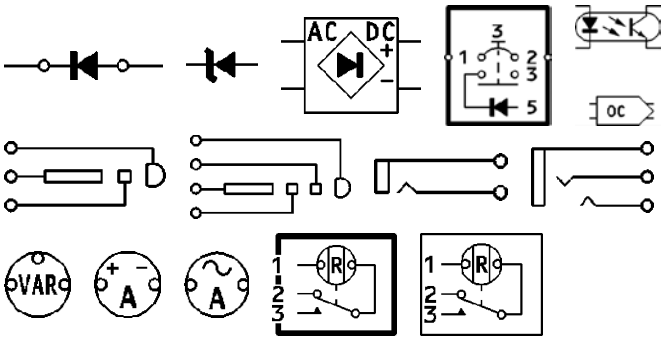


Figure 1.2: A sample of few electrical symbols [FRESH, 2007].

For evaluation, several different human evaluators are used to form ground-truths since there is no absolute ground-truth associated to our dataset – which is obvious in case of real-world application. In order to handle this, we have asked them to manually select what they consider as "similar" symbols, for all queries by providing the following information. With respect to the provided query, chosen candidates must have either

- similar visual overall appearance or
- similar significant parts or graphical elements within it.

In more formal terms, for each query the "ground-truth" is considered to be the set of symbols formed by the union of all human selected sets. These evaluators were not required to provide any ranking order nor degree of visual resemblance. As an example, we consider the first sample in Figure 1.3 as a query and all others are similar selected database symbols. In this example, we try to address our specific problem together with difficulties associated with it. Moreover, it provides how our problem can be made different with the general symbol recognition problem.

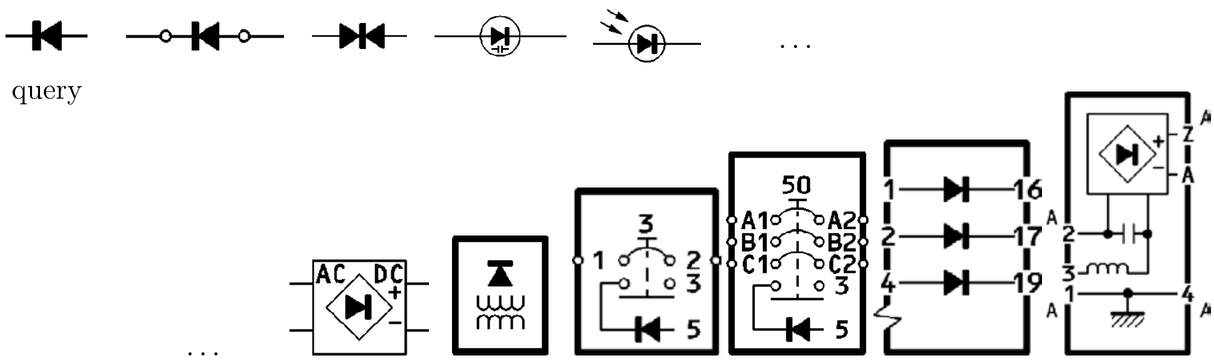


Figure 1.3: Selection of similar database symbols for a chosen query. The first symbol is considered to be a query symbol and remaining are 'ground-truths' associated with it. Ground-truth associated with a chosen query is not only concerned with similar visual overall appearance but also significant known parts within it, that may or may not be connected with other graphical elements and text.

1.2.3 Framework – Contributions of the Thesis

Symbol recognition usually means recognising isolated symbols. In this context, we present a feature based descriptor in order to describe global appearance of the graphical symbols.

Contribution 1. DTW–Radon: A new shape descriptor for graphics recognition.

We introduce a method based on dynamic programming for matching the Radon features, for graphics recognition. The key characteristic of the method is to use dynamic time warping (DTW) algorithm to match corresponding pairs of the radon features for all possible projections. To make it rotation invariant, we estimate the orientation angle of the pattern that will avoid incorrect alignment. Overall, this allows to exploit the Radon property that includes both boundary as well as internal structure of shapes, while avoiding compressing pattern representation into a single vector and thus miss information, thanks to the DTW.

In Chapter 3, we present the concept in detail. We validate by comparing it with a set of major shape descriptors over several different datasets. In all experimental tests, we focus on the optimisation of the number of bins i.e., projections in order to reduce the computational complexity.

Considering the real-world problem, isolated symbol recognition methods are not always suitable, however. In case of composite documents that contain textual and graphical elements, one needs to be able to extract and formalise the links that exist between the images and the surrounding text, in order to exploit the information embedded in those documents. Therefore, correct extraction, representation of both visual data, textual and graphical structures, and organisation are the first steps towards further automated knowledge, information discovery and information retrieval or data mining on more complex data than just text.

Within this context, for graphics recognition, we primarily focus on three main items:

1. the extraction of visual elements (vocabulary) that compose an image;
2. the expression of visual relations between the elements;
3. knowledge discovery, formal learning techniques and classification using the vocabulary and relations mentioned above, including vocabulary shape analysis.

The first two items are based on standard image treatment analysis operations [Rendek et al., 2004] while, contributions of the thesis mainly rely on *item 3*. Based on this, there are five main contributions:

Contribution 2. Symbol recognition using spatial relations.

We present a method for symbol recognition based on the spatio-structural description of a 'vocabulary' of extracted visual elementary parts. The method is primarily based on spatial relations between the possible pairs of labelled vocabulary types such as *circles* and *corners*, which are further used as a basis for building an attributed relational graph (ARG) that fully describes the symbol. These spatial relations integrate both topology and directional information. Thanks to our labelling of attribute types, corresponding relation alignments are possible between the two graphs while avoiding the general NP-hard graph matching problem.

In Chapter 4, we provide a detail study of how spatial relations are used to represent complete symbols as well as a comprehensive comparison with the other spatial relation models including state-of-the-art approaches for graphics recognition.

Contribution 3. Integrating spatial relations with statistical features for symbol recognition.

We present method for symbol recognition based on both the spatial organisation and shape features of the visual primitives that compose the symbol. Keeping the ARG based symbol description mentioned in *contribution 1*, shape signatures are integrated in two different ways. First, shape signatures are for labelling vertices. Second, shape features are applied only to the vocabulary which show significant shape variations.

In this contribution, statistical features are integrated with spatial relations wherever necessary *i.e.* where significant shape variations occurs in the visual primitives. We present it in Chapter 5. We concentrate on advancing recognition performance by integrating shape features on the visual primitives. To handle this, we study the pertinence of the major set of shape descriptors.

Contribution 4. Spatial-bag-of-features for symbol recognition.

We introduce a concept for symbol recognition which is based on spatial-bag-of-features that are computed from extracted visual elementary parts. The key characteristic of the overall process is to use topological relations information to guide metric directional relations. Such a topological guidance makes our method efficient since matching goes only to the relevant candidates which share same topological configuration and therefore, it reduces computational time i.e., rapid search is possible.

In Chapter 6, we explain it in detail including the way to recover the shortcomings of the methods presented in previous contributions. Besides recognition performance, we addresses the advantages and extension of the method in such a way that it can be used as a 'user-friendly' symbol retrieval application. For instance, `'find symbol which has a rectangle containing a circle'`. As in Chapter 4 and 5, fundamental spatial relation models are employed including a set of major shape descriptors for comparison.

Contribution 5. Formal learning techniques for semantic related symbol recognition.

We present the use of formal learning techniques to automatically learn non-trivial descriptions of symbols, based on a formal description. The previously described ARG using visual vocabulary and relative positioning is transformed into a first-order logic (FOL)

based description of the graphical symbols. This representation is then used as an input to an ILP solver, in order to deduce non obvious characteristics that may lead to a more semantic related recognition process.

Overall, it yields a common characterisation of the sub-set of symbols from the pool of domain knowledge. As said before, the method can adapt any complexity of the classification in contrast to statistical learning models. We have validated and explained in Appendix C.

1.3 What is Next?

In the previous section, the key contributions are mentioned and are outlined chapter by chapter. In the following chapter, before addressing these contributions in detail, we present **state-of-the-art** of graphics recognition. In Chapter 2, we particularly focus on structural, statistical and specially designed approaches for graphics recognition.

End of Chapter

Keywords

Structural Approaches – Spatial Relations, Graphs, Tree
Statistical Approaches – Shape Analysis
Graphics Recognition.

2 State-of-the-Art Approaches

Contents

2.1 Organisation of the Chapter 51
2.2 Graphics Recognition 52
2.2.1 Statistical Approaches 52
2.2.2 Structural Approaches 54
Primitive Selection 55
Spatial Relations 56
Symbol Representation and Matching 60
2.2.3 Hybrid Approaches 63
2.3 Application Domains 64
2.4 Conclusions 65
2.5 What is Next? 66

Foreword. This chapter discusses common approaches for graphics recognition including their main characteristics. More specifically, it is focussed on structural approaches using spatial relations and statistical approaches via shape analysis, as well as symbol recognition methods. At the end, applications related to graphics recognition domain will be provided. ■

2.1 Organisation of the Chapter

The chapter is organised as follows. We start with introduction to graphics recognition in Section 2.2. It includes statistical, structural and hybrid approaches. We discuss them one after another. In Section 2.2.1, we explain statistical approaches by reviewing shape analysis for graphics recognition. Similarly, in Section 2.2.2, we review structural approaches where we mainly highlight three issues: primitive selection, spatial relations, and symbol representation and matching. We review hybrid approaches in Section 2.2.3. They primarily consist of methods especially designed for graphics recognition. Section 2.3 gives more application domains, showing

the interest of close-to graphics recognition problem. Section 2.4 gives a brief summary of the chapter.

2.2 Graphics Recognition

Research on graphics recognition has an extremely rich state-of-the-art, aimed to localise/recognise symbols depending on the applications. We are not aware of published comprehensive literature review since [Cordella and Vento, 2000b, Lladós et al., 2002]. Existing approaches, specifically those based on feature based matching, can roughly be sorted into three approaches: structural, statistical and hybrid.

Statistical signatures [Jain et al., 2000] using shape descriptors for instance, do not perform well for all kinds of images because they represent global appearance by a fixed size feature vectors regardless of the size or complexity of the image. Also, in case when the studied image is characterised by their possible relations between the different parts within it, there exists no direct possibility to describe binary relations. In those cases, structural approach would be a better choice [Pavlidis, 1977, Bunke and Sanfeliu, 1990]. In structural approaches: graph-based representations [Conte et al., 2004] in particular, graphs are not constrained to a fixed size. This means that number of vertices and edges, and relationships like spatial and conceptual can be possibly varied from an image to another. Due to which, practical use of structural approaches for graphics recognition is limited due to intense computational complexity. As a consequence, several symbol recognition methods are introduced for instance, pixel-based approaches like statistical integration of histogram array (SIHA) [Yang, 2005] and Kernel density matching [Zhang et al., 2006]. Within this context, very recently, [Tombre, 2010] asks '*which features distinguish graphics recognition from general pattern recognition problems?*' since it has been observed that methods vary from one application to another. Therefore, there still exists growing interest in graphics recognition domain.

The aforementioned paragraph is just a snapshot by taking a few examples of prominent work related to graphics recognition. In what follows, we clearly explain the approaches on a detailed basis. In all cases, considering our specific problem i.e., symbol recognition, we highlight the following issues:

1. symbol representation: for both isolated as well as it is found in composite form;
2. comparison of two representations to do recognition; and
3. extensibility.

The first two items are directly related to the recognition performance of the method. Recognition performance consists of two major factors: recognition rate and computational complexity (or in our case execution time). The quality of the symbol representation determines the former factor while comparison of two representations reflects the latter one.

2.2.1 Statistical Approaches

In statistical approach, two issues are primarily involved: feature selection and model selection for recognition. Usually each pattern is represented as a n-dimensional feature vector i.e. $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ and recognition is carried out by partitioning the feature space into different known classes.

Basically, in symbol recognition, geometric features like centroids, axes of inertia, circularity, area, line intersections, holes and projection profiles, geometric moments and image transformations are used for the recognition of different patterns such as graphical symbols and characters. Within this framework, shape descriptors basically take into account the global appearance of pattern. A comprehensive explanation can be found in [da Fontoura Costa and Junior, 2001].

In this overview, we are considering shape features first. Shape descriptors are often categorised into two classes. They are contour-based and region-based descriptors. According to the geometry of primitives, $1D$ descriptors essentially correspond to contour-based and skeleton-based descriptors while $2D$ descriptors correspond to region-based descriptors.

1. Contour-based descriptors: Common contour-based descriptors are
   - Fourier descriptors (FD) [Persoon and Fu, 1986, Zhang and Lu, 2005];
   - polygonal [Attalla and Siy, 2005] and curvature [Bernier and Landry, 2003] primitives from the contour information;
   - skeleton extraction [Zhu and Yuille, 1996]; and
   - geometric invariant theory [Mundy and Zisserman, 1992].
2. Region-based descriptors – They take into account all pattern pixels. A few common methods are
   - based on the moment theory [Teh and Chin, 1988, Bailey and Srinath, 1996]; including geometric, Legendre, Zernike moments [Chong et al., 2003]; and
   - generic fourier descriptor (GFD) [Zhang and Lu, 2002].

   Contour-based and region-based descriptors can be compared as follows. Contour-based descriptors are appropriate for silhouette shapes. Consequently, they are limited since they cannot capture interior contents, disconnected shapes or shapes with holes where boundary information is not available. Furthermore, they are sensitive to noise since boundary information would be deteriorated that could significantly change the tangent values. In contrast, region-based descriptors can be applied to a broader scope of symbols since they take internal contents into account as well. For example, GFD is a generalised case of FD. To avoid the problem of rotation in the Fourier spectra, the $2D$ Fourier transform (FT) is applied on a polar-raster sampled shape. This approach outperforms common contour-based (classical Fourier and curvature approaches) and region-based ZM shape descriptors. However, they present high computational complexity compared to contour-based ones. One of the key points is: the normalisation procedure to satisfy common geometric invariance properties introduces errors as well as they are sensitive to noise, eventually affecting the whole recognition process. It applies to region-based descriptors.

A review of shape descriptors for document analysis was presented in [Terrades et al., 2007a] which highlights notion of feature extraction method, their properties and pointing out their weakness. In [Valveny et al., 2007], shape descriptors (pixel-based, contour-based and structural) are evaluated and compared. The authors mention that there is no unique shape descriptor that can fit to all types of problems. In what follows, we explain some common shape descriptors.

- Zernike moments are based on orthogonal moments [Hse and Newton, 2004]. When applied to symbol recognition, they are computed on evenly distributed data points. One of the attractive features of Zernike moments is the use of magnitudes that are invariant to

rotation [Bailey and Srinath, 1996] and can be constructed to an arbitrary order easily. In contrast to other moments, it can be used in distorted symbols [Teh and Chin, 1988].

- Fourier descriptors for symbol recognition apply to simple models [Yu, 1995] and parts of the models [Kim et al., 1993]. Moreover, signature approximation from Fourier series in order to represent symbols as presented in [Tabbone et al., 2003] has shown the geometric properties like scaling, translation, symmetry and scaling can be preserved. In addition, Fourier descriptors do not require shape contour information which is frequently encountered in other Fourier descriptors variants.

- Symbol shape representation through the Radon transform is limited to line drawing shapes [Tabbone et al., 2003]. Also, use of an $\mathcal{R}-$transform to obtain the $\mathcal{R}-$signature in particular [Tabbone et al., 2006] misses important information.

- In [Nguyen et al., 2008], shape context [Belongie et al., 2002] has been implemented together with vector model. This integration helps to reduce large set of candidates search while matching takes place.

Remarks. Global signal-based descriptors are usually quite fault tolerant to image distortions, since they tend to filter out small detail changes. However, when symbols are combined, approaches that rely on centroid detection like [Yuen et al., 1998] tend to fail. Others, like shape context [Belongie et al., 2002] are sensible to occlusions on the symbol boundaries. Furthermore, it does not satisfy scale invariant properties. In these statistical approaches, signatures are simple with low computational cost. However, discrimination power and robustness strongly depend on the selection of optimal set of features for each specific application as well as proper fusion of classifiers [Kudo and Sklansky, 2000, Ruta and Gabrys, 2000, Duda et al., 2001, Schmitt et al., 2008]. To overcome some of these drawbacks, for symbol recognition, descriptors have been combined to increase performance [Salmon et al., 2007, Barrat and Tabbone, 2010, Terrades et al., 2007b].

One of the contributions of this thesis is that we introduce a method based on dynamic programming for matching the Radon features, for graphics recognition (*cf.* Chapter 3). The key characteristic of the method is to use align corresponding pairs of radon features for all possible projections. Overall, this allows to exploit the Radon property that includes both boundary as well as internal structure of shapes, while avoiding compressing pattern representation into a single vector and thus miss information. It outperforms the state-of-the-art of common shape descriptors.

Notwithstanding this improvement, global signal-based shape descriptors difficultly accommodate with connected or composite symbols.

2.2.2 Structural Approaches

Structural approaches are based on symbolic data structures such as strings, trees and graphs. Graphs are found to be the most general one since strings and trees are always included as special cases. In document analysis, the most recent advancement in graph-based pattern recognition is presented in [Bunke and Riesen, 2011].

In graph-based representation, an image is characterised by properties of spatial objects within it and their pairwise possible connections between either the complete objects or key points and regions. region adjacency graph (RAG) [Rosenfeld, 1974] uses interest regions and adjacency relations, for instance. Let us take a few examples to illustrate the concept. Figure 2.1

shows an example of a proximity graph that is based on local descriptor via Harris-Laplace detector [Mikolajczyk and Schmid, 2004] as presented in [Rusiñol and Lladós, 2008]. Similarly, other forms of graph like attributed relational graph (ARG) or so-called line graph (specifically designed for symbol representation) [Delalandre et al., 2008] as shown in Figure 2.2, provide fundamental parameters related to structural approaches. On the whole, graphical symbols are represented by describing shapes via a suitable set of geometric shape primitives and their possible pairwise connections.

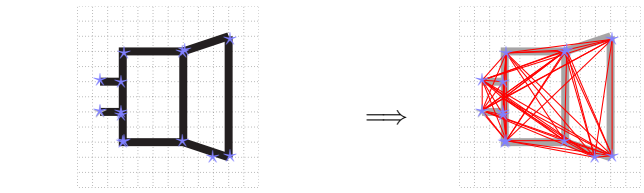


Figure 2.1: A simple example, showing proximity graph representation using interest points using local descriptor.

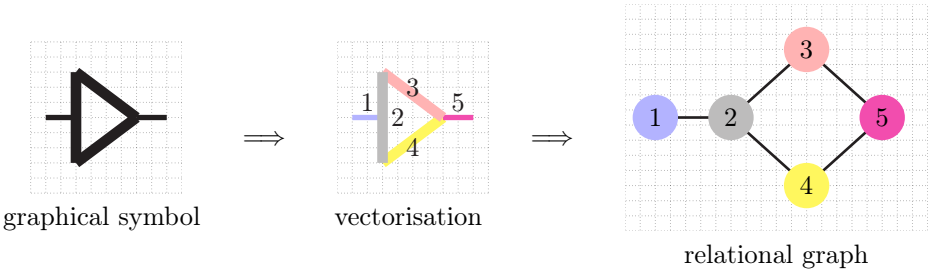


Figure 2.2: A simple example, illustrating graphical symbol representation using relational graph.

In what follows, we explain

1. primitive selection,
2. spatial relations, and
3. symbol representation and matching.

Overall, primitive selection and their connecting relationships are used to describe the whole symbol. Matching is ultimately based on how symbols are described.

Primitive Selection

A symbol is naturally represented by the number of pixels or points. As said before, use of key points using local descriptor as in Figure 2.1 is an example. However, it is unclear which local descriptors are more appropriate and how their performances depend on key points or region detectors, since descriptors usage varies from one application to another. Comprehensive explanations can be found in [Mikolajczyk and Schmid, 2005]. Vectorisation [Doermann, 1998] is another preliminary task for extracting primitives like simple *lines* [Chiang et al., 1998, Zheng et al., 2005] and *arcs* [Dori, 1995, Dosch et al., 2000, Lamiroy and Guebbas, 2009] including geometric primitives such as *loops* and *contours* or simple shapes like *circles* and *rectangles*. These are then used to describe symbol by integrating possible connections between them.

A very common drawback arising from vectorisation is error-prone raster to vector conversion. For example, primitive types such as *arc* and *corner* are not always extracted as the degradation or noise level increases. However, robust vectorisation exists in line drawing images including engineering drawings [Janssen and Vossepoel, 1997, Hilaire and Tombre, 2006, Song et al., 2002]. Consequently, use of such low-level primitives varies widely according to the nature of the symbol.

Spatial Relations

Spatial reasoning is regarded as a central skill to many human tasks, as being able to communicate about the space. A common and natural way to share spatial information is through the use of spatial predicates [Retz-Schmidt, 1988] such as *Left of* and *Right of*, in order to derive relationship between the spatial entities. To handle image recognition, partial recognition of visual primitives used to guide the recognition of remaining parts within it [Bar and Ullman, 1993]. It is based on the question i.e., effect of spatial relations on recognition performance [Biederman, 1972, Cave and Kosslyn, 1993].

In document image analysis, relations are used for analysis of architectural documents and for automatically recognition of models [Vandenbrande and Requicha, 1993] and the graphical drawings understanding of scanned colour map documents [Centeno, 1997] or to define efficient retrieval methods [Gevers and Smeulders, 1992, Lee and Hsu, 1992, Heidemann, 2004, Medasani and Krishnapuram, 1997].

Before explaining the details about the impact of spatial relations for document image analysis, in the following, we first briefly outline spatial relations, their categorisations, properties as well as their appropriate usage.

Families – which family? [Winston, 1975] provides one of the first consistent studies of spatial relations and their variations according to the context. An important family of spatial relations and associated properties came from Freeman [Freeman, 1975]. Those relations are categorised as follows:

1. topological relations;
2. metric relations; and
3. directional relations.

Topological relations describe neighbourhoods and incidence such as *disconnected* and *externally connected*, directional relations provide order in space such as, *north* and *south*, and metric describes distance relations like *near* or *far*.

In what follows, we shortly provide background for all families of spatial relations including their global characteristics.

Topological Relations In connection with [Renz and Nebel, 1998], basic topological relations close to human understanding are *disconnected* (DC), *externally connected* (EC), *covers* (Cr) or *covered by* (CB), *contains* (Cn) or *inside* (I) and *equal* (EQ). Figure 2.3 shows an illustration of it. In this illustration, we can observe that topological relations satisfy affine transformation invariant properties.

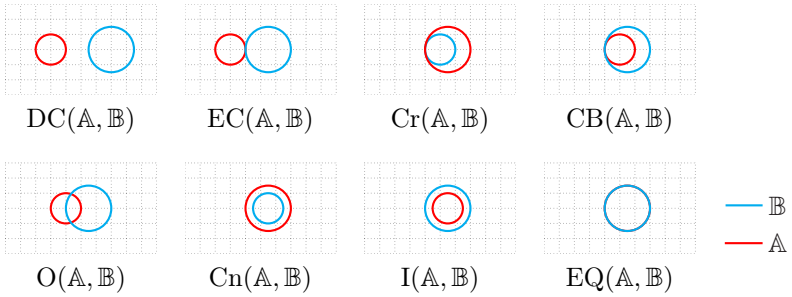


Figure 2.3: Possible topological relations between two objects $\mathbb{A}$ and $\mathbb{B}$.

Metric Relations provide an idea of distance between two spatial objects. Consider a metric on a set X is a function (called the distance function or simply distance), then

$$d : X \times X \to \mathbb{R}$$

where $\mathbb{R}$ is the set of real numbers. For all x, y and z in X, this function is required to satisfy the following conditions:

c1. $d(x, y) \geq 0$ i.e., non-negativity

c2. $d(x, y) = 0$ if and only if $x = y$

c3. $d(x, y) = d(y, x)$ i.e., symmetry

c4. $d(x, z) \leq d(x, y) + d(y, z)$ i.e., triangle inequality.

Based on it, many modifications have been made according to the applications.

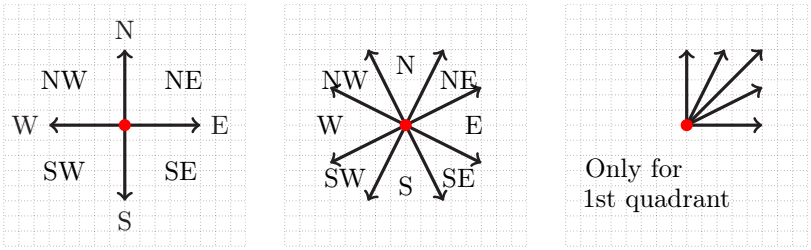
Computer representation of geo-spatial information has been motivated by proximity relations such as *nearness* and *locality*, as described in [Worboys, 1995]. For example, *nearness* is derived form relative distance i.e., $\textit{relative distance}(x, y) = \frac{d(x,y)}{\mu_c}$ where μ_c is the mean distance measured from the centre.

Directional Relations In general, they provide an idea of *orientation* of the primary spatial objects with respect to a reference. Each object is represented by one or more representative points, and the space is partitioned using these points. The relation is then determined using the partitions to find where the object representative points are.

Depending on the concept of partitioning, there are several different ways to handle directional relations between the spatial objects. In the following, we provide some of the fundamental concepts or models.

1. Angle-based model – Angle based relational models provide a true orientation of spatial objects. Two fundamental models fall under this category.

   - Cone-shaped model: The relations can be approximated by their centres based on discretised angle [Miyajima and Ralescu, 1994, Mitra, 2002] i.e., $\angle(C_{\mathbb{A}}, C_{\mathbb{B}})$ between the objects $\mathbb{A}$ and $\mathbb{B}$. It is sometimes also called the bi-centre model. It provides several different configurations based on star calculus [Renz and Mitra, 2004]. Figure 2.4 shows progressive refinement of bi-centre model from 4-directions to 16-directions. This model possesses the following shortcomings.



(a) 4-directions (b) 8-directions (c) 16-directions

Index: N = North, S = South, E = East and W = West

Figure 2.4: Star calculus via bi-centre model using angle-based theory.

  - Distance between two spatial objects is not possible. Therefore, relations remain unchanged unless there exists significant change in separation.
  - It does not take shape and size into account. Therefore, it is robust to small variations of shape and size in one sense. On the other hand, it is not certain that centroid falls within the spatial object.
  - It does not carry topological information. Therefore, for objects having *inside* or *contain* topological configurations for instance, it produces ambiguous spatial predicates.
  - In case of centroid coincidence from two studied objects (even for two different shapes), it yields no relations.

  Therefore, it is best suited in situations where the studied objects are far from each other.

- Angle histogram: Approaches like *Angle histograms* [Wang and Keller, 1999] consider all pixels where the cone-shaped model only took the centroid into account. As a consequence, their computational cost increases dramatically. Let us consider two objects $\mathbb{A}$ and $\mathbb{B}$ as the sets of their pixels:

$$\mathbb{A} = \{a_i\}_{i=1\ldots m} \text{ and } \mathbb{B} = \{b_j\}_{j=1\ldots n}.$$

  The $m \times n$ pairs of points allow for the computation of a set of angles $\theta_{i,j}$ between each (a_i, b_j). The histogram $\mathcal{H}$ representing the frequency of occurrence of each angle f_θ can then be formulated as,

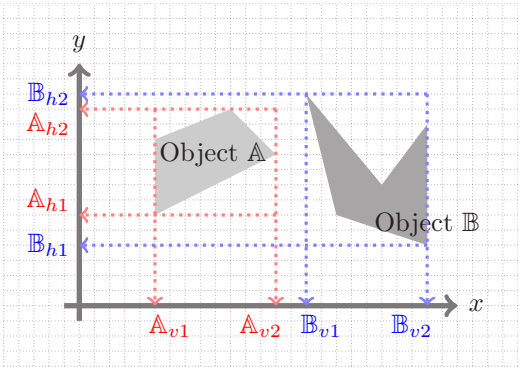
$$\mathcal{H}_\theta(\mathbb{A}, \mathbb{B}) = [\theta, f_\theta].$$

  For a simplicity, histogram values can be aggregated into a single value.

  As a remark, we note that centroid and aggregation methods mainly compute a single value for pairwise relations. The major difference with bi-centre model lies in the fact that the averaging is made on the objects' points for the centroid method while it is applied after angle computation in the aggregation method. If the objects are far from each other, this averaging converges to bi-centre model.

2. Projection model – The projection model uses the classical minimum boundary rectangle (MBR) model [E.Jungert, 1993]. Figure 2.5 shows the MBR model and its iconic vertical and horizontal projections, regardless the compactness of the objects. Compacity is defined as the percentage of the spatial object in the MBR. Such a partitioning

the space is dynamic according to the shape and size variation of the reference object. [Goyal and Egenhofer, 2001] provides an example of it. It possesses the following proper-

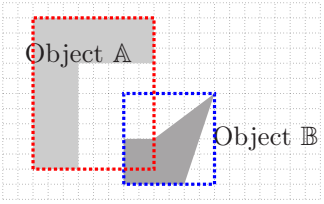


Index: $\mathbb{X}_{h1}$ and $\mathbb{X}_{h2}$ = horizontal projections, and $\mathbb{X}_{v1}$ and $\mathbb{X}_{v2}$ = vertical projections, where $\mathbb{X}$ can be $\mathbb{A}$ or $\mathbb{B}$

Figure 2.5: MBR model showing horizontal and vertical projections.

ties.

- MBR is only appropriate as long as spatial objects are regular. This means that it depends on compactness. Compacity of more than 0.80 is found to be regular.
- False overlapping is possible. It misleads results in case of no actual intersection of the spatial objects (see Figure 2.6).



Compacity of $\mathbb{A} = 0.56$.

Figure 2.6: An example showing false overlapping.

Until now, we have presented the fundamental concepts for all three basic families of spatial relations including their general properties. Now, we provide knowledge representations in doing spatial reasoning.

Qualitative or Quantitative? Another way to categorise spatial reasoning [Dutta, 1991, Dehak, 2002] is either based on

1. qualitative or
2. quantitative knowledge representation.

The former one conveys boolean spatial information i.e., 1 for the presence of spatial object and 0 otherwise. The latter one is often based on fuzzy set theory [Zadeh, 1965] allowing a better

managing of the ambiguous aspect of spatial relations. However, their usages and impacts vary equally according to the nature of applications. To make a quick difference between them with the help of an example, let us follow Figure 2.5 where object $\mathbb{A}$ is provided as the reference. Now,

$$\text{qualitative relation}(\mathbb{B},\mathbb{A})=\begin{bmatrix}0 & 0 & 1\\ 0 & 0 & 1\\ 0 & 0 & 1\end{bmatrix} \text{ and quantitative relation}(\mathbb{B},\mathbb{A})=\begin{bmatrix}0 & 0 & 0.005\\ 0 & 0 & 0.880\\ 0 & 0 & 0.115\end{bmatrix}.$$

This means that object $\mathbb{B}$ is found to be extended from *right bottom* to *right top* with respect to object $\mathbb{A}$. The choice of either qualitative or quantitative spatial representation can be summarised as follows.

- Qualitative interpretation provides spatial relations more close to natural language as used in spatial predicates like *right*, *left*. Qualitative knowledge is usually cheaper since it does not need to compute percentage value.
- On the other hand, quantitative spatial reasoning is chosen in cases where it needs natural instead of all-or-none relations [Freeman, 1975]. Consequently, fuzzy concepts have been introduced in several different applications since they are directly related to shape and size information and are comparable to human perception. Angle histograms rather than just a single angle value is one of the basic examples. Similarly, fuzzy landscape based on fuzzy morphological operators [Bloch, 1999] and force-histogram approaches have been popularly used such as [Matsakis and Wendling, 1999]. If there is uncertainty, then it is inherently suited for fuzziness [Morris, 2003].

Symbol Representation and Matching

Graph-based or graph-like representation – representing the relationships between the entities used in the image – provides an abstract concept of the studied image. Figure 2.7 provides an illustration of it. A few common examples are:

- [Moravec, 1977] introduced the notion of interest points by considering corners and junctions. Later, it has been represented by using local descriptors such as SIFT features [Lowe, 2004].
- [Rosenfeld, 1974] introduces the modelling of adjacency relations between the segmented regions.
- Skeletal graph for shape representation [Kimia et al., 1995, Torsello and Hancock, 2004] uses skeleton points, which are categorised into three families: junction, end and branch points. The first two categories represent vertices while the latter ones, represent edges.

Graph-based representation schemes are not limited however, they vary widely such as attributed relational graph (ARG), line graph for line drawing images and proximity graph for connecting neighbour relationships between segmented objects and so on. In other words, they vary from one application to another i.e., a single representation does not fit for all [Jouili and Tabbone, 2011].

In the following, we report graphics recognition applications related to graph-based representation.

In [Groen et al., 1985], skeletal graph is used to represent the symbol in electrical diagrams. For graph matching, bounded search is used to select the pose of the graph such as rotation, translation and scale for a minimum error transformation. It is entirely based on probabilistic models.

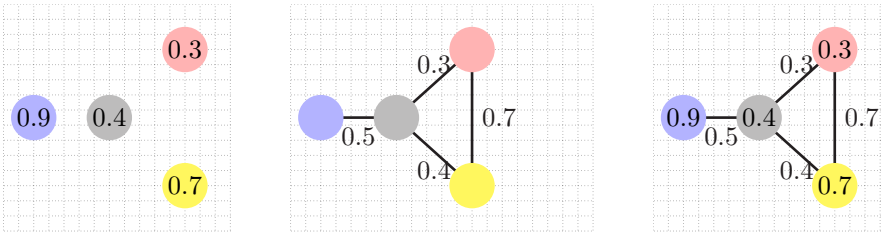
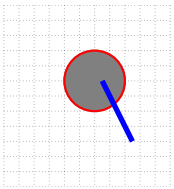


Figure 2.7: Graph representations: (a) only node features, (b) only relations and (c) integrating both. Features are labelled with numeric weighted values in order just to give an example.

In [Lee et al., 1990], graphs are used for building a model-based scheme for recognising hand-drawn symbols in schematic diagrams. To construct the graph, endpoints, junctions and crossings are represented by vertices attributed with the number of neighbours and the angles between incident edges. The edges represent connecting lines in the drawing attributed with the length and curvature of the respective line. For recognition, a small number of candidate graphs are selected by means of geometrical constraints. The distance measure is based on the correspondences found between components of the two graphs and the classification of the unknown symbol is carried out using a nearest-neighbour classifier.

For on-line symbol representations in [Changhua et al., 2000, Wenyin, 2003], ARGs are used for graphical primitives such as *lines* and *arcs*. They are labelled as vertices while their connections using topological relations like *intersection* and *inside*, define labelled edges. Similarly, another concept on using spatial relation to represent complete symbol is spatial relation graph (SRG) [Xiaogang et al., 2004]. As said before, a few topological relations like *intersection*, *parallelism*, *tangency* and *concentric* are used to connect primitive components such as *line* and *arc* segments, and *ellipses*. These are not sufficiently discriminative considering the complexity in drawing images such as ladder-like sketches. For example, $overlap(line_{\mathbb{A}}, rectangle_{\mathbb{B}})$ does not answer about 'how much'. Therefore, metrical details are necessary as explained in [Egenhofer and Shariff, 1998], for instance. Figure 2.8 shows an example of it. Overall, such precision provides more confidence in recognition.



$$\text{Inner transversal splitting} = \frac{\text{length(Line} \cap \text{Region)}}{\text{length(Line)}}$$

Figure 2.8: Inner transversal splitting example based on [Egenhofer and Shariff, 1998].

Run-length encoding (RLE) [Monagan and Roosli, 1993] is another way to build graph-like structure. It works well for sparse line drawings mainly having horizontal and vertical lines. The main weakness exists on vectorisation quality and it does not really possess topological information. In case of minimal line property preserving (MLPP) graph [Burge and Kropatsch, 1998], which takes topological features into account (in addition to geometric) such as *loops* and *holes* and it succinctly provides relationships among them.

Network based graph [Ah-Soon and Tombre, 2001] – hierarchical structure has also been studied, where constraints between segments is used. Interestingly, it can update its structure if new symbols are added to the system. In [Liu et al., 2004], symbol is described by using sub-division tree (SDT) where relational model yields connections to branching nodes. In both cases, the weak point however is that they are not unique i.e., they vary according to segment selection within the symbol.

No matter which schemes are used to make graphs, graph matching in general, finds the best match using sub-graph isomorphism between the input image and the models. The similar argument has been highlighted in [Bunke and Shearer, 1998]. Graph structure varies with the level of noise, occlusion, distortion etc. and therefore it suffers from intense computational complexity due to the general NP-hard problem of sub-graph matching. This means that computation of various graph similarity measures is exponential in the number of involved vertices.

- In such a situation, error-tolerant sub-graph isomorphism graph edit operations has been introduced. [Messmer and Bunke, 1998, Lladós et al., 2001] are two major examples in the literature. In [Messmer and Bunke, 1998], vertices represent line segments without attributes and the edges connect two line segments that have a common endpoint, attributed with the angle between the corresponding line segments. In [Lladós et al., 2001], hand-drawn diagrams are represented by RAGs, where vertices represent closed regions (attributed with a boundary string resulting from polygonal approximation of the region) and edges represent the adjacency relations between them. To handle similarity measure between the two regions, string matching techniques are applied in addition to the error-tolerant sub-graph isomorphism. Overall, it combines string matching (at the local level) with graph matching (at the global level).

- Median graph representation [Jiang et al., 1999, Jouili et al., 2010] is another alternative that reduces processing delay to a some extent.

- Tree-like graphs also save storage space. Similarly, graph probing concept [Barbu et al., 2006] also reduces matching delay.

- Techniques like integer linear programming [Nemhauser and Wolsey, 2004] are used to model the sub-graph isomorphism problem [Bodic et al., 2009] for symbol detection using RAG.

- Statistical classification based on the structural representation of symbol. As an example, we take [Luqman et al., 2009], where signatures exploit structural details of symbols. It presents quantitative features like number of vertices and edges, symbolic features like number of edges with fixed or predefined attributes, and range of features that include relative length and angle. The method however has not been applied to degraded or distorted and noisy models.

Those methods however, do not provide significant reduction in computational complexity. In [Fankhauser et al., 2011], a sub-optimal edit distance is proposed that runs in polynomial time. The reason for its sub-optimality is that *edge information* is taken into account only in a limited fashion during the process of finding the optimal vertex assignment between two graphs.

Remarks. Based on the review on graph-based representation and matching, we account the following issues: relations and computational complexity. In parallel, we explain how this thesis contributes to the problem.

– In graph-based representations, relationships like spatial, temporal and conceptual play a crucial role. Among them, spatial information within the image provide sufficient information such that recognition guidance is feasible. For instance, topological configurations such as *disconnected* in addition to their orientation relations such as '*right of*' describe the studied image clearly, instead of using separately. Such a combination of both topological and directional relations is the obvious choice to build the complete relational graph. Existing approaches are not aware of it. One of the key contributions in this thesis is to integrate them into a fixed size feature vector representation. Besides, natural relations (instead of qualitative relations) would yield accurate guidance. Therefore, our relational descriptor provides point–based distribution of the spatial objects, that relies on a fuzzy concept. We will present the detail explanation of this contribution in Chapter 4.

– As said before, computational complexity is one of the major drawbacks of graph-based representations. We avoid the general NP-hard sub-graph matching problem by having uniquely labelled vertices attributed with *circle*, *corner* etc. As a consequence, vertex and edge matching thus becomes trivial and can be done in near-constant time. This becomes comparable to [Fankhauser et al., 2011] where a sub-optimal edit distance is proposed that runs in polynomial time.

2.2.3 Hybrid Approaches

Except those aforementioned approaches, there are few methods designed for symbol recognition that preserve both statistical and structural information. Hybrid representations tend to be very application dependent. We will explain a few common approaches in the following.

- Representing symbols using vectorial signatures has been started since 90's. A relevant work can be cited as [Ventura and Schettini, 1994] where the symbol is first well skeletonised and vectorised. To represent the whole symbol in terms of signatures, features such as line segments, their lengths, acute angles between them and intersecting points are used. Within this framework, in [Dosch and Lladós, 2003], a set of 5 spatial relations such as *neighbour*, *parallelism*, *L junctions* and *V junctions* are employed. The approach is of course faster compared to graph-based representation. But on the other side, more spatial predicates need to be defined in order to accurately locate features that ultimately avoids false alarm.

- Statistical integration of histogram array (SIHA) feature [Yang, 2005], is proposed for skeleton symbols. It considers every triplet of points $\{p_i, p_j, p_k\}$ to represent two different structural features: its angle $\measuredangle p_j, p_i, p_k$ and length ratio $\min\{\frac{|p_i p_j|}{|p_i p_k|}, \frac{|p_i p_k|}{|p_i p_j|}\}$. For all points, both angle histogram (AH) and length ratio histogram (LH) are constructed for symbol representation. To make fixed size histograms (in terms of matrices), one needs to fix the number of bins.

  In [Zhang et al., 2006], symbols are represented as $2D$ Kernel densities and their similarity is measured by using the Kullback-Leibler divergence [Cover and Thomas, 2006]. Since symbols consists of a series of coordinate points in $2D$ feature space, computing density distribution is immediate compared to SIHA. To make it rotation invariant, it consumes

significant running time. To handle this, two different methods are used: gradient-based angle searching or independent component analysis.

In both approaches [Yang, 2005, Zhang et al., 2006], there exist two major drawbacks.

- As in shape descriptors, they do not distinguish the symbols appearing in composite form. However, it is well suited for segmented as well as linear line-drawing symbols i.e., GREC dataset, for instance.
- They suffer from high computational complexity.

Remarks. In general, hybrid approaches try to integrate best of the two worlds: structural and statistical. But in the literature, faster methods using vectorial signatures do not provide satisfactory results and pixel-based approaches like SIHA while providing better recognition take high computational time.

In this framework, one of our contributions concerns the use of integration of spatial relations and shape description together in our ARG framework (Chapter 5). We use the similarity between spatial organisation and shape features of the visual primitives that compose the symbol. This concept clearly addresses what [Tombre, 2010] has mentioned i.e., capable method to efficiently combine structural and statistical methods, and spatial nature of the information makes structural methods quite natural.

2.3 Application Domains

Related to graphics recognition, there exist several different varieties of applications. Some of the major domains will be enumerated with their very brief underlying inherent problems.

1. Logic circuit diagrams [Groen et al., 1985, Jiang et al., 1999] – The main aim is to describe the symbols that are characterised by *loop* structures, rectilinear connections between them including textual informations.

2. Engineering drawings – There are varieties of diagrammatic notations for engineering drawings that makes difficult to take standard classes for recognition. In general, it is used to distinguish two levels of symbols:

   - graphical elements [Ablameyko, 1997] such as *arcs*, *straight lines*, *dashed lines*, *cross-hatched areas*
   - domain-dependent knowledge [Collin and Colnet, 1994] due to combination of *arrow-heads*, *lines* and *texts*.

3. Maps – It is another challenging problem. It mainly includes conversion of GIS formats like cartographic information, satellite images and special areas (i.e., cadastral, telephone, water etc.) Within this, three different types of maps with their own conventional notations are defined.

   - Cadastral city maps [Roy et al., 2008] where polygonal shaped symbols are filled by hatched patterns, mixed with texts and annotations.
   - Utility maps [Adam et al., 1999] which mainly includes binary images consisting of lines, small symbols, network-like structure of water, telephone, gas etc.

- Geographic maps analysis [Myers et al., 1995, Samet and Soffer, 1996b] where line objects that are used to represent streets for instance including colour information.

4. Musical scores [Gordo et al., 2010, Fornés, 2009] – The interpretation has usually three phases: extraction of staff lines, individual notes recognition, and interpretation of the whole musical score.

5. Architectural drawings – One of the most active researches [Ah-Soon and Tombre, 2001, Valveny and Martí, 2000, Macé et al., 2010] in graphics recognition. It includes the recognition of higher level primitives such as doors, windows, chairs and stairs. However, segmentation has become a central problem since symbols are appeared in the embedded form within the document.

6. Logo [Cortelazzo et al., 1994, Psyllos et al., 2010, Sun and Chen, 2011] – Basically, it is based on extracting signatures from contour information in addition to the OCR integration since logo image contains text in it.

7. Others – Besides those aforementioned applications related to symbol recognition, the following problems are still worth considering.

   - Mathematical formula recognition [Fujiyoshi et al., 2010].
   - Pen-based applications: on-line symbol [Yu, 2007] and character [K.C. et al., 2010] recognition.
   - WWW based graphics indexing and querying [Paek and John, 1998].

2.4 Conclusions

Until now, we have explicitly discussed common methods used in document image analysis as well as a few major approaches designed for symbol recognition including their pros and cons. In this section, based on that, we can make two remarks.

1. ***Structural versus statistical.*** Structural approaches do not have rich set of mathematical tools. They mostly depend on k-means classification [Bunke and Riesen, 2011]. While in statistical approaches, there is a rich set of mathematical tools such as several different types of neural networks, decision theoretic methods, machine learning procedures and clustering algorithms [Jain et al., 2000]. Besides, graph structure varies with the level of noise, occlusion, distortion etc. while statistical approaches tolerate noise to a some extent, without significantly changing the global signatures. Overall, they all depend on how images are described.

2. ***Problem dependent methods.*** In case of graphics recognition problems, one must usually rely on a priori knowledge about the nature of the document to locate candidate symbols or regions of interest to which specific recognition techniques should be applied. It is therefore not surprising that different approaches have been devised for different applications. It is quite common, in the most general area of image analysis, that methods and strategies used with one image type may also prove to be effective for images of a different type. However, the methods needed to solve certain problems peculiar to a given type of images, can seldom be used for other image types. Similar arguments have been highlighted in [Lladós and Kwon, 2004].

2.5 What is Next?

Based on a comprehensive overview of the three different approaches: statistical, structural and hybrid approaches, we will explain and validate our major contributions in the following chapters. Note that in every chapter, we will revise the related work in more detail in case it requires.

End of Chapter

Keywords

Radon Features,
Orientation Estimation
Dynamic Programming,
Graphics Recognition.

3

DTW–Radon: A New Shape Descriptor for Graphics Recognition

Contents

3.1 Organisation of the Chapter 68
3.2 Introduction 68
3.2.1 Related Work 69
3.2.2 Outline of the proposed Method 69
3.3 Method 70
3.3.1 Radon Features 70
Affine Transformation Properties 70
Number of Bins 71
Pattern Orientation and Number of Bins 74
3.3.2 Matching 75
3.3.3 Recognition and Retrieval 77
3.4 Experiments 78
3.4.1 Benchmarking Methods 78
3.4.2 Graphics Recognition and Retrieval 78
Experimental Results 78
Experimental Results Analysis 83
3.4.3 Case Studies 84
3.5 Conclusions 88
3.6 What is Next? 89

Foreword. In this paper, we introduce a method based on dynamic programming for matching the Radon features, for graphics recognition. The key characteristic of the method is to use dynamic time warping (DTW) algorithm to match corresponding pairs of histograms for all possible projections. This

allows to exploit the Radon transform property to include both boundary as internal structure of shapes. This avoids avoiding compressing pattern representation into a single vector and thus miss information.

Experimental tests over several different graphical symbol datasets such as FRESH, GREC and CVC show that the method is robust to distortion, degradation and deformation including. ■

3.1 Organisation of the Chapter

The chapter is organised as follows. We start with introduction in Section 3.2 which includes a description of the underlying problem, our motivations, related work as well as brief outline of the proposed method. Section 3.3 provides details of the proposed method composed of pattern representation, orientation estimation and matching. A full experimental set up is reported in Section 3.4. It contains a comparison with state-of-the-art methods along with the extension of the work – case studies. Section 3.5 concludes the work.

3.2 Introduction

Automatic recognition, description, classification and grouping of patterns are important issues in many disciplines such as, biology, computer vision, artificial intelligence or remote sensing where shape analysis plays an important role [Loncaric, 1998, Zhang and Lu, 2004]. Patterns [Watanabe, 1985] for instance, can be graphical elements in document-like symbols of any kind including cursive characters, shoes and finger–prints.

To do pattern recognition through shape analysis, two major stages are shape representation and matching. Within the framework, the following issues will be taken into account.

1. Shape representation

   - Global shape representation is widely used due to its simplicity. Moreover it does not necessarily require extra pre-processing and segmentation contrary to local pattern representation.
   - Shape representation must be sufficiently enriched i.e., important internal content must not be missed.
   - Another common implementation problem is the inability to assume the shape distribution of the patterns in the feature space. Consequently, non-parametric methods are much more practical since they use linear functions to describe classifiers.

2. Matching

   Matching techniques are often induced by how patterns are represented. For instance, normalising pattern information into a fixed size single vector (as in global signal-based descriptors [Zhang and Lu, 2004]) provides immediate matching.

In the existing literature, feature normalisation procedure makes feature vector invariant to rotation while on the other hand, it does not provide complete shape information. Therefore, to respect the aforementioned points in *item 1*, features should not be normalised in order to exploit detail shape information including internal contents.

We take advantage of the Radon transform [Deans, 1983] to represent pattern and dynamic programming (DP) is used to match patterns of any size that avoids compressing pattern representation into a single vector unlike the use of $\mathcal{R}-$transform [Tabbone et al., 2006], for instance. The work is the extension of the previous work designed for off-line signature verification [Coetzer, 2005, Jayadevan et al., 2009] along with the orientation angle estimation as

in [J-K and S-Z, 2005]. Overall, an optimal selection of number of projections or bins to collect the Radon features will be an interesting part of the work, in addition to the appropriate integration of estimated orientation. Moreover, number of bins determines the optimal interest between the two: detail shape information and running time.

Very recently, the growing interests of shape descriptors for symbol representation has been highlighted [Valveny et al., 2007], under the purview of document analysis. This motivates the present work for graphics recognition.

3.2.1 Related Work

Shape representation [Loncaric, 1998, Zhang and Lu, 2004] has been an important issue in pattern analysis and recognition. In this context, features are often categorised as

1. region-based and
2. contour-based descriptors.

For both categories, in Section 2.2.1 of Chapter 2, we have discussed the advantages, disadvantages as well as their appropriateness by taking a major set of shape descriptors. Very briefly, contour-based descriptors do not provide the information about whole shape of the pattern while region-based ones do. However, normalisation of the shape distribution (region-based, for instance) in the feature space in order to make it rotation invariant, introduces errors.

For document image analysis, a review of shape descriptors has been made [Terrades et al., 2007a]. Among them (*cf.* Section 2.2.1), the growing interest of the Radon Transform can be observed from the following previous works such as matching of line–drawing images [Fränti et al., 2000], $2D$ shape and symbol categorisation [Leavers, 2000, Tabbone et al., 2006] and gait recognition [Boulgouris and Chi, 2007]. Those attest the robustness of the Radon transform to noisy, degraded as well as occluded samples. Radon based descriptors is able to encode contour information as well as internal structure. In the following section, we discuss a brief outline of major points of the proposed method.

3.2.2 Outline of the proposed Method

In connection with all highlighted items in Section 3.2, we use the Radon transform [Deans, 1983] to represent patterns. Pattern matching is then made through DTW [Bellman and Kalaba, 1959, Kruskall and Liberman, 1983, Keogh and Pazzani, 1999] between corresponding the Radon features at every specified projection or bin. The Radon transform is essentially a set of parametrised histograms or features. Therefore, the method addresses the optimal selection of number of bins rather than uses the straightforward discretised Radon transform. This means that there exists a trade-off between the computational running time and optimal number of bins selection.

Before matching, the primary issue is to make feature affine transformation invariant. Among three, rotation property has been carefully integrated with Radon features. Remaining translation and scaling properties are handled by using image centroid.

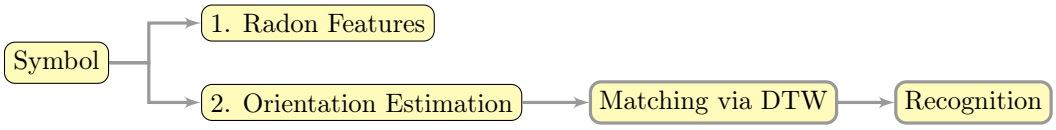


Figure 3.1: An architecture.

Figure 3.1 shows an architecture of the method. In particular, the method is able to recover the principle shortcoming appeared in classical $\mathcal{R}-$signature [Tabbone et al., 2006] that does not exploit complete shape information because Radon information at every projecting angle is compressed into a single value via $\mathcal{R}-$transform.

3.3 Method

3.3.1 Radon Features

As shown in Figure 3.2(a), a collection of projections of the pattern at different angles refers to the Radon transform [Deans, 1983]. In other words, the Radon transform for any image pattern $\mathcal{P}(x, y)$ and for a given set of angles can be thought of as computing the projection of all non-zero points. The resulting projection is the sum of the non-zero points for any pattern $\mathcal{P}(x, y)$ in each direction, which eventually form a matrix. Therefore the integral of $\mathcal{P}$ over a line $\mathcal{L}(\rho, \theta)$ defined by $\rho = x\cos\theta + y\sin\theta$ can formally be expressed as,

$$R(\rho, \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \wp(x, y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$

where $\delta(.)$ is the Dirac delta function,

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Also, $\theta \in [0, \pi[$ and $\rho \in]-\infty, \infty[$. For the Radon transform, $\mathcal{L}_i$ be in normal form (ρ_i, θ_i). For all θ_i, the Radon transform now can be described as the length of intersections of all lines $\mathcal{L}_i$. Note that the range of ρ i.e., $-\rho_{min} < \rho \leq \rho_{max}$ is entirely based on the size of pattern. The complete illustration is provided in Figure 3.2(b).

Affine Transformation Properties

Since the Radon transform itself does not satisfy invariance properties, we consider the following affine transformation properties to adapt it for recognition.

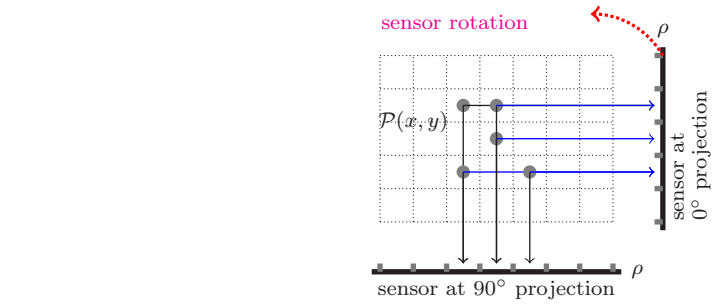
1. Translation: In case of translation, we use image centroid (x_c, y_c) such that translation vector is $\vec{u} = (x_c, y_c)$: $R(\rho - x_c\cos\theta - y_c\sin\theta, \theta)$. Therefore, translation of f results in the shift of its transform in ρ by a distance equal to the projection of translation vector of the line $\mathcal{L}$ (see Figure 3.2).

2. Scaling: For *scaling*, features are normalised into $[0, 1]$ at every projecting angle.

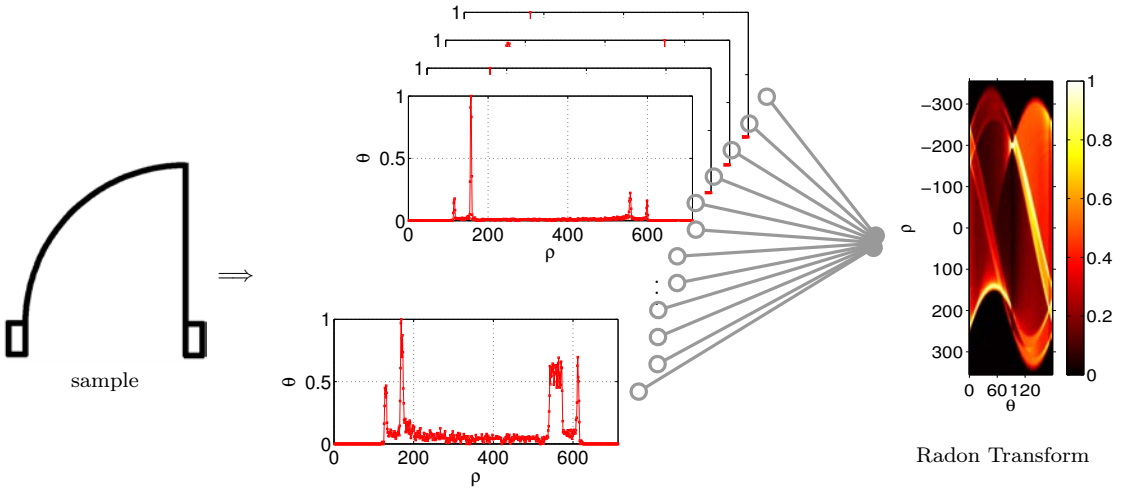
3. Rotation: For rotation, orientation angle can be estimated as in [J-K and S-Z, 2005],

$$\alpha = arg\left[\min_{\theta}\frac{d^2\sigma_\theta^2}{d\theta^2}\right]$$

where $\sigma_\theta^2 = \frac{1}{P}\sum_\rho (R(\rho, \theta) - \mu_\theta)^2$ is the variance of projection at θ, $\mu_\theta = \frac{1}{P}\sum_\rho R(\rho, \theta)$ and P, the number of samples. If angle of rotation is α, then $R^\alpha(\rho, \theta) = R(\rho, \theta + \alpha)$. This simply implies a circular shift of the histograms such that it does not require histograms duplication from $[0, \pi[$ to $[\pi, 2\pi[$ as in [Coetzer, 2005] to make rotation invariant.



(a) Projection example.



(b) Radon Features showing projections, for corresponding sample.

Figure 3.2: Radon transform – a complete illustration: (a) pattern projection using the Radon transform theory and (b) radon features for all possible projections over $[0, \pi[$ and a complete radon transform i.e., a collection of all radon histograms or features.

Figure 3.3 shows the Radon features for reference, rotation, scaling, as well as degradation instances from a known class of graphical symbol [GREC, 2003]. In this illustration, orientation angle estimation, the Radon histograms from their corresponding sample images are provided. In all cases, the Radon histograms are similar to each other except the magnitude difference. More examples are provided in Figure 3.4.

Number of Bins

As said before, the Radon transform is essentially a set of parametrised histograms or features since projecting angle extends over $[0, \pi[$. Each projecting angle represents a bin. This means that every bin yields a Radon histogram as a feature (*cf.* Figure 3.2). Therefore, in generic form,

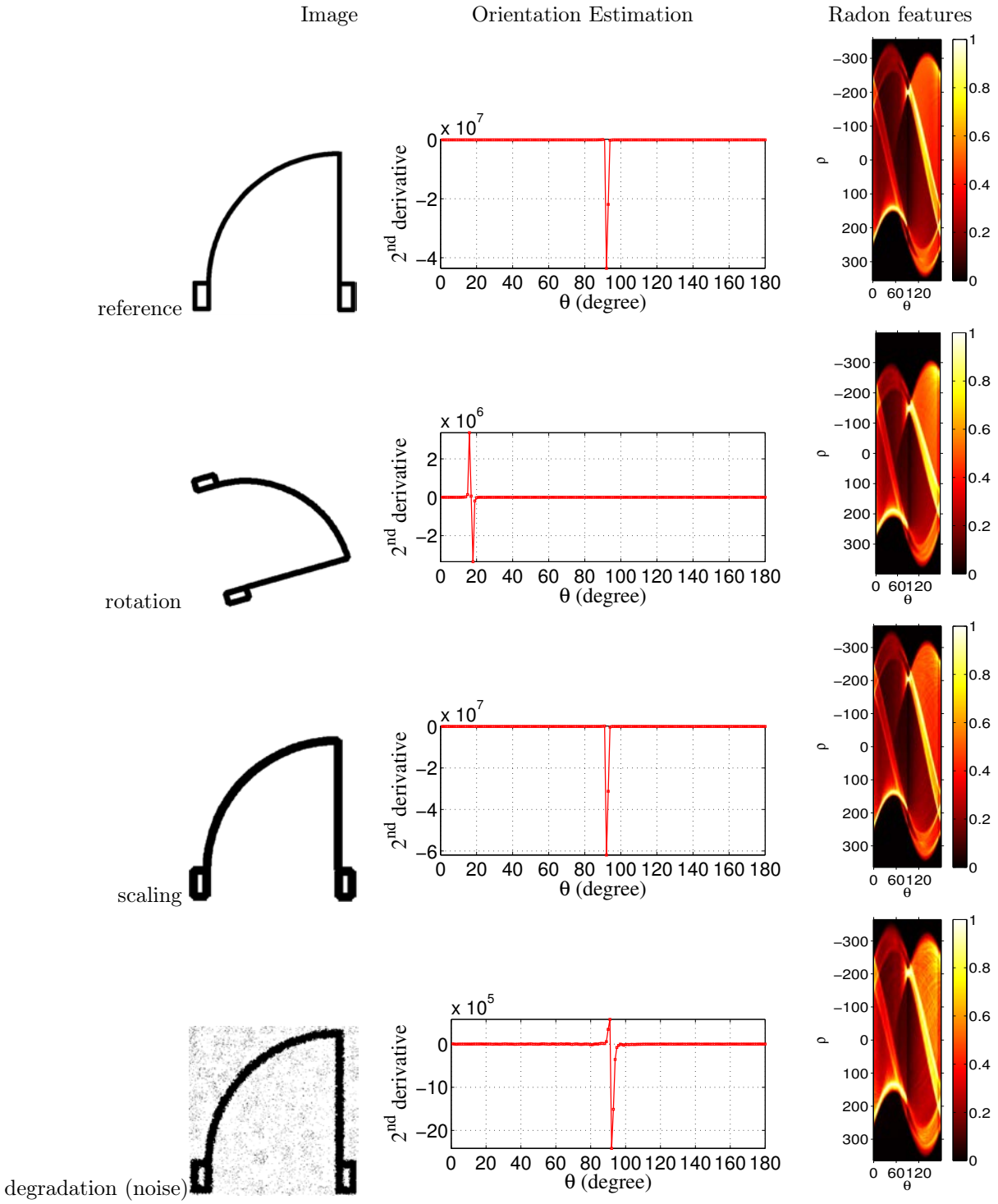


Figure 3.3: Images, their corresponding orientation estimation and the Radon features. Samples are taken from graphical symbol dataset [GREC, 2003]. Orientation angle α has been estimated as follows: 90° for reference, scale and degradation images while 17° for rotation image.

a complete set of the Radon features $R(\rho, b)$ can be expressed as,

$$\mathcal{F} \quad = \quad \{\mathbf{F}_b\}_{b=1,\ldots,B}. \tag{3.1}$$

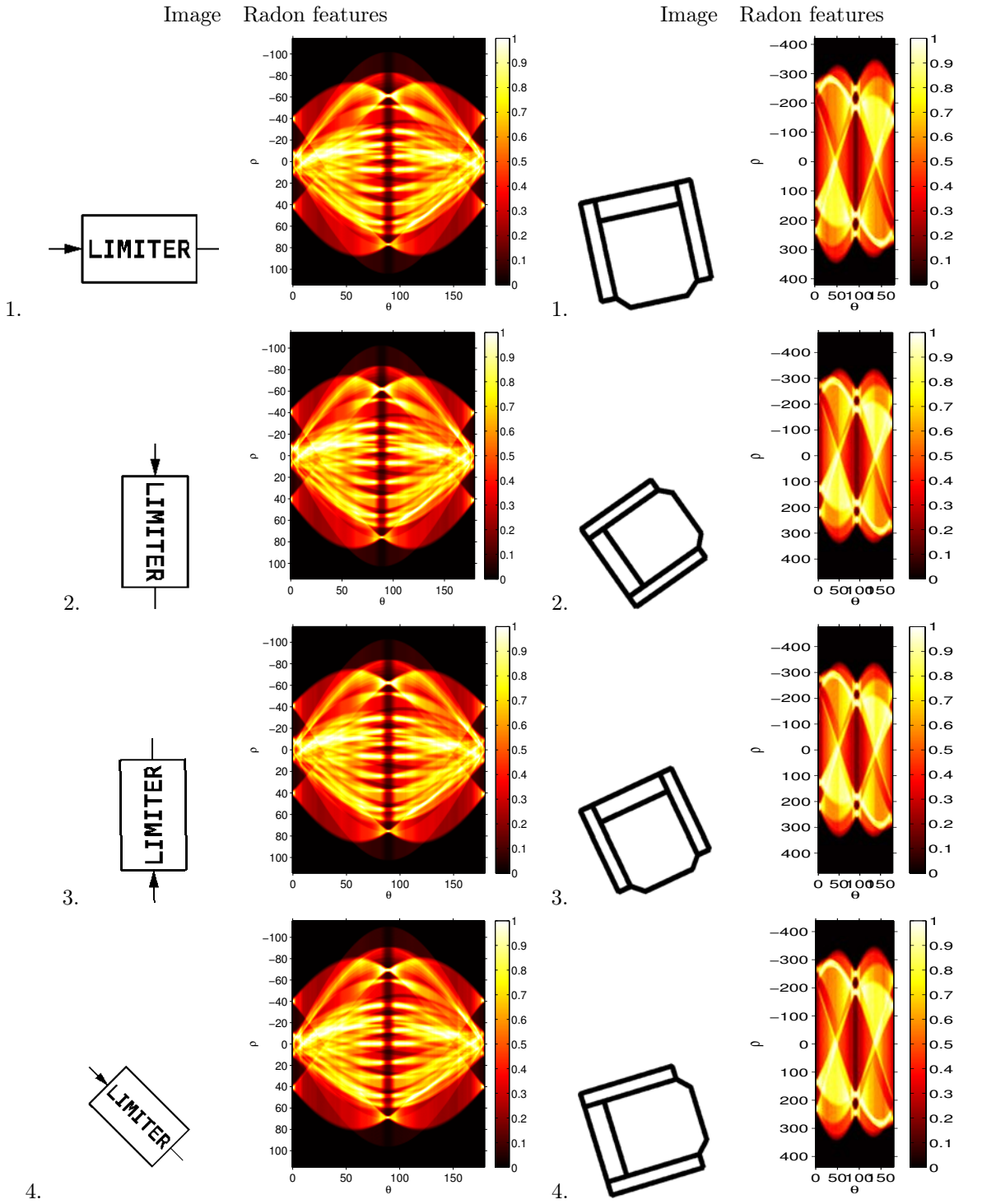


Figure 3.4: Rotational images and their Radon features. Samples are taken from graphical symbol dataset [GREC, 2003, FRESH, 2007].

The total number bins B can be formulated as,

$$B = \frac{180}{\Theta} = \begin{cases} 180 & \text{when } \Theta = 1^\circ \\ 90 & \text{when } \Theta = 2^\circ \\ 60 & \text{when } \Theta = 3^\circ \\ 36 & \text{when } \Theta = 5^\circ \\ \dots & \text{so on,} \end{cases}$$

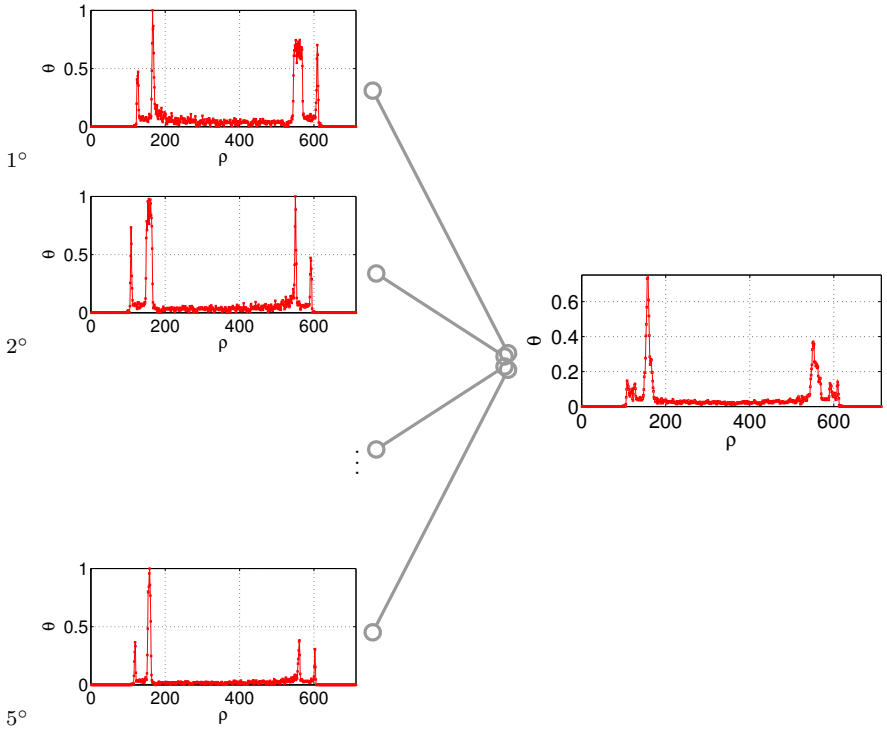


Figure 3.5: A single Radon feature over projection angle-range Θ by averaging all discretised histograms within it. It goes over the range $[0, \pi[$.

where Θ is the projection angle-range. A single Radon feature at bin b is $\mathbf{F}_b$ is the collection of histograms $\mathcal{H}$ at every discrete projection angle. If $\Theta = 1°$, it is called discrete Radon transform where there are 180 bins. While, let us say if $\Theta = 5°$, all histograms within it are averaged to form a single Radon feature and overall, there are 36 bins. Figure 3.5 shows an example. Overall, our Radon feature is the collections of histograms over the provided projecting angle-range over $[0, \pi[$. Performances of using several different number of bins will be reported in Section 3.4 in order to provide generic behaviour of the method. Besides, it provides an optimal selection of number of bins (projections) that are needed to supersede state-of-the-art of shape descriptors.

Pattern Orientation and Number of Bins

As mentioned before, to make the Radon feature $\mathcal{F}$ rotation invariant, we make a circular shift of the histograms from where it is found to be skewed. The adopted method mentioned before, is well suited even if there exists two or more than two spikes appeared. Its because of the second derivative of the projection made by the Radon Transform.

Since the orientation of the studied pattern is just an estimated value, it may not accurately provide circular shift of the histograms in case when the discrete Radon transform i.e., $B = 180$ is used. On the other hand, we use several different number of bins in order to collect the features at every projecting angle-range instead of a single projecting angle. In such a situation, if the estimated value lies within the projecting angle-range, it provides accurate circular shift. This means that the bigger the projecting angle-range, the accurate the circular shift of the histograms to make it rotation invariant. On the other side, it misses the actual information while providing

a single feature for a large value of projecting angle-range. Consequently, optimal selection of number of bins is required. The influence of the number of bins B will be reported in Section 3.4.

3.3.2 Matching

As explained in Section 3.3.1, we have a collection of set of features $\mathcal{F}$ in a specified number of bins B, to represent any pattern $\mathcal{P}$.

Given two patterns: query $\mathcal{P}^q$ and database $\mathcal{P}^d$, matching can be obtained between corresponding features from the complete set of $\mathcal{F}^q$ and $\mathcal{F}^d$. The Radon transform generates different ρ sizes depending on the image contents' size. In order to be able to adapt to these differences in size, DTW algorithm is employed. In what follows, matching computation will be explained first and then derived the matching score between the whole patterns.

Dynamic Time Warping It allows us to find the dissimilarity between two non-linear sequences potentially having different lengths [Kruskall and Liberman, 1983, Keogh and Pazzani, 1999]. Let us consider two feature sequences $\mathbf{X} = \{\mathbf{x}_k\}_{k=1,\dots,\mathcal{K}}$ and $\mathbf{Y} = \{\mathbf{y}_l\}_{l=1,\dots,\mathcal{L}}$ of size K and L, respectively. The aim of the algorithm is to provide the optimal alignment between both sequences. At first, a matrix $\mathcal{M}$ of size $\mathcal{K} \times \mathcal{L}$ is constructed. Then for each element in matrix M, local distance metric $\delta(k,l)$ between the events e_k and e_l is computed. Let $D(k,l)$ be the global distance up to (k,l),

$$D(k,l) \quad = \quad \min\left[D(k-1,l-1), D(k-1,l), D(k,l-1)\right] + \delta(k,l)$$

with an initial condition $D(1,1) = \delta(1,1)$ such that it allows warping path going diagonally from starting node $(1,1)$ to end $(\mathcal{K},\mathcal{L})$. The main aim is to find the path for which the least cost is associated. The warping path therefore provides the difference cost between the compared signatures. Formally, the warping path is,

$$\mathcal{W} = \{w_t\}_{t=1\dots\mathcal{T}},$$

where $\max(k,l) \leq \mathcal{T} < k+l-1$ and t^{th} element of $\mathcal{W}$ is $w(k,l)_t \in [1:\mathcal{K}] \times [1:\mathcal{L}]$ for $t \in [1:\mathcal{T}]$. The optimised warping path $\mathcal{W}$ satisfies the following three conditions.

c1. boundary condition: $w_1 = (1,1)$ and $w_{\mathcal{T}} = (\mathcal{K},\mathcal{L})$

c2. monotonicity condition: $k_1 \leq k_2 \leq \cdots \leq k_{\mathcal{K}}$ and $l_1 \leq l_2 \leq \cdots \leq l_{\mathcal{L}}$

c3. continuity condition: $w_{t+1} - w_t \in \{(1,1)(0,1),(1,0)\}$ for $t \in [1:\mathcal{T}-1]$

c1 conveys that the path starts from $(1,1)$ to $(\mathcal{K},\mathcal{L})$, aligning all elements to each other. **c2** forces the path advances one step at a time. **c3** restricts allowable steps in the warping path to adjacent cells, never be back. Note that **c3** implies **c2**.

We then define the global distance between $\mathbf{X}$ and $\mathbf{Y}$ as,

$$\Delta(\mathbf{X},\mathbf{Y}) = \frac{D(\mathcal{K},\mathcal{L})}{\mathcal{T}}.$$

The last element of the $\mathcal{K} \times \mathcal{L}$ matrix, normalised by the T provides the DTW-distance between two sequences where T is the number of discrete warping steps along the diagonal DTW-matrix. The warping path can be computed with the help of back-tracking along the minimum cost index

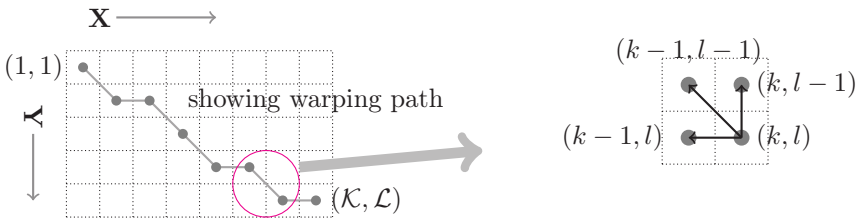


Figure 3.6: DTW algorithm – illustration.

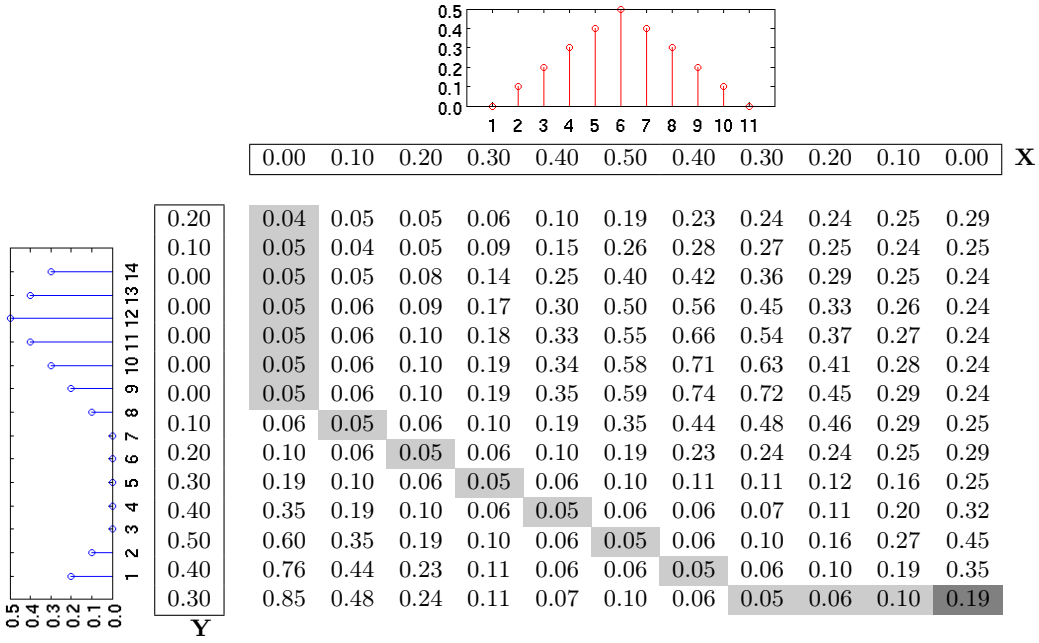


Figure 3.7: Computing DTW distance between two sequences having different lengths. The last element of the matrix, normalised by the number of DTW steps along DTW diagonal matrix provides global distance between them i.e., $\Delta(\mathbf{X}, \mathbf{Y}) = \frac{0.19}{17} = 0.0112$.

pairs (k, l) starting from $(\mathcal{K}, \mathcal{L})$. As shown in Figure 3.6, the back-tracking procedure following the optimal warping path is handled with the help of dynamic programming (DP).

$$w_{t-1} \quad = \quad \begin{cases} (1, l-1) & \text{if } k = 1 \\ (k-1, 1) & \text{if } l = 1 \\ \text{argmin}\{D(k-1, l-1), D(k-1, l), D(k, l-1)\} & \text{otherwise,} \end{cases}$$

where we take the lexicographically smallest pair in case 'argmin' is not unique. The overall process is illustrated with the help of an example in Figure 3.7.

Matching Score Aggregating distances between the Radon features in all corresponding bins $b \in B$ between $\mathcal{P}^q$ and $\mathcal{P}^d$ yields a global pattern matching score,

$$\text{Dist.}(\mathcal{P}^q, \mathcal{P}^d) \quad = \quad \sum_{b=1}^{B} \Delta(\mathbf{F}_b^q, \mathbf{F}_b^d).$$

Overall, since we have employed estimated orientation α, Radon feature alignment can go one-to-one basis. In case where α is not integrated, it is also possible to treat the method as rotation invariant via the use of the Radon feature alignments for all i.e., one-to-all. Formally, it can be expressed as,

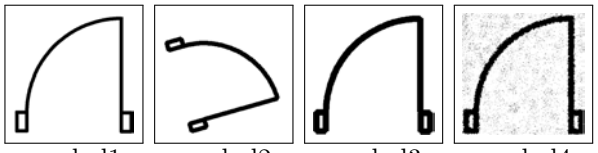
$$\text{Dist.}(\mathcal{P}^q, \mathcal{P}^d) \quad = \quad \sum_{b,b'=1}^{B} \min_b \left(\Delta(F_b^q, F_{b'}^d) \right).$$

The difference between the two different ways of aligning the Radon features lies in computational complexity i.e.,

$$\text{Computational Complexity of Dist.}(\mathcal{P}^q, \mathcal{P}^d) = \begin{cases} O(B) & \text{one-to-one alignment, and} \\ O(B^2) & \text{one-to-all alignment.} \end{cases}$$

Therefore, overall running time is depend on how many bins will be employed.

In what follows, let us take a few examples to see the difference between the two DTW matching scores. Figure 3.8 and 3.9 provide matching scores matrices between rotated, scaled, degraded, distorted as well as deformed images. In both cases, we use the Radon features with 180 number of bins. Those illustrations provide similar behaviour between the two distance computation methods: 'with' and 'without α'. Therefore, in our experiments, considering time complexity issue, the Radon features alignment will be made with α.



symbol1 symbol2 symbol3 symbol4

	symbol1	symbol2	symbol3	symbol4
symbol1	0.0000	0.0002	0.0004	0.0011
symbol2		0.0000	0.0008	0.0014
symbol3			0.0000	0.0019
symbol4				0.0000

(a) with α.

	symbol1	symbol2	symbol3	symbol4
symbol1	0.0000	0.0002	0.0004	0.0010
symbol2		0.0000	0.0008	0.0017
symbol3			0.0000	0.0020
symbol4				0.0000

(b) without α.

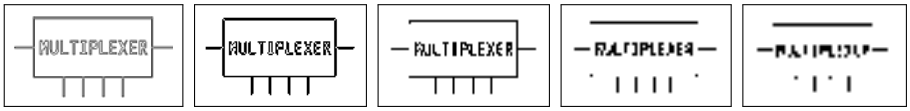
Figure 3.8: Matching scores between reference, rotation, scaling and degradation sample images. A reference sample image is taken from GREC dataset [GREC, 2003]. It follows Figure 3.3.

3.3.3 Recognition and Retrieval

We can now use the previously described approach as a global pattern matching score. This score expresses the similarity between database patterns and query. Our problem is: given a set of points S in a metric space M_s and a query point $q \in M_s$, find the closest point in S to q. Now, we express similarity as,

$$\text{Similarity}(\mathcal{P}^q, \mathcal{P}^d) = 1 - \overline{\text{Dist.}}(\mathcal{P}^q, \mathcal{P}^d) = \begin{cases} 1 & \text{for the closest candidate} \\ 0 & \text{for the farthest candidate.} \end{cases}$$

where matching scores are normalised into $[0, 1]$ by, $\overline{\text{Dist.}}(\centerdot) = \frac{\text{Dist.}(\centerdot) - \text{Dist.}^{min.}(\centerdot)}{\text{Dist.}^{max.}(\centerdot) - \text{Dist.}^{min.}(\centerdot)}$. Ranking can therefore be expressed on the decreasing order of similarity. In our experiments, we will distinguish "recognition" (search for the closest candidate) from "retrieval" (where closest candidates are retrieved for a given short-list).



symbol1 symbol2 symbol3 symbol4 symbol5

	symbol1	symbol2	symbol3	symbol4	symbol5
symbol1	0.0000	0.0015	0.0025	0.0044	0.0055
symbol2		0.0000	0.0026	0.0039	0.0049
symbol3			0.0000	0.0038	0.0052
symbol4				0.0000	0.0027
symbol5					0.0000

(a) with α.

	symbol1	symbol2	symbol3	symbol4	symbol5
symbol1	0.0000	0.0012	0.0024	0.0043	0.0061
symbol2		0.0000	0.0025	0.0034	0.0048
symbol3			0.0000	0.0036	0.0051
symbol4				0.0000	0.0030
symbol5					0.0000

(b) without α.

Figure 3.9: Matching scores between distortion as well as deformed images. A reference sample image is taken from FRESH dataset [FRESH, 2007].

3.4 Experiments

3.4.1 Benchmarking Methods

We confront $\mathcal{D}-$Radon[2] for several different number of bins B, with well-known descriptors:

1. $\mathcal{R}-$signature [Tabbone et al., 2006],
2. GFD [Zhang and Lu, 2002],
3. SC [Belongie et al., 2002] and
4. ZM [Kim and Kim, 2000].

For those descriptors, it is important to fit the best parameters. For the Radon transform, projecting range is $[0, \pi[$. In case of GFD, we have tuned the parameters, radial $(4 : 12)$ and angular $(6 : 20)$ frequencies to get the best combinations. For SC, we use 100 sample points. In case of ZM, we have used 36 *zernike* functions of order less than or equal to 7.

3.4.2 Graphics Recognition and Retrieval

We work on several different datasets in different contexts. However, we primarily focus on distorted, degraded and deformed symbols in document analysis – graphics recognition and retrieval. Depending on the nature of the datasets, evaluation metrics vary. For more understanding about evaluation metrics and their proper usages, we refer to Appendix A. In the following, we shortly mention the evaluation metric once dataset is introduced.

In order to test the robustness of the Radon features, we work on raw data, no pre-filtering (de-noising, for instance) has been applied.

Experimental Results

We have used both computer-printed and hand-drawn symbols. It is composed of symbols from architectural and electrical wiring diagrams in separate datasets. Throughout the experiments,

[2]For simplicity, DTW–Radon is named as $\mathcal{D}-$Radon.

we keep on focussing how many number of bins for $\mathcal{D}-$Radon will be appropriate to do recognition. This means that comparison is first made among the state-of-the-art of shape descriptors before confronted 'the best' performer with $\mathcal{D}-$Radon for several different number of bins. To make easy comparison, the compared results are highlighted. For benchmarking methods, the best performer is highlighted first. Then in case of $\mathcal{D}-$Radon, the highlighted numeric figures from different number of bins will then be identified as optimal selection of B.

FRESH dataset and Ground-truth Formation

Dataset We work on a real world industrial problem to identify a set of different known symbols in aircraft electrical wiring diagrams as in [FRESH, 2007, Tombre and Lamiroy, 2008]. Figure 3.10 gives some examples of symbols in the database. Symbols may either be very similar in shape – and only differ by slight details – or either be completely different from a visual point of view. Symbols may also be composed of other known and significant symbols and need not necessary be connected. It is composed of roughly 500 different known symbols. Our dataset is completely unlabelled and imbalanced i.e., neither ground truth is given nor identical number of similar symbols exist for all queries.

Ground-truth Formation Since there is no absolute ground-truth associated to our dataset, we have proceeded by using human validation, but by taking care of eliminating subjective bias. In order to achieve this we have asked 6 volunteers to manually select what they consider as "similar" symbols, for all queries executed in this section. Human evaluators have chosen the candidates which have similar visual overall appearance or which have significantly similar parts with respect to the chosen query. In our testing protocol, we consider that a result returned from an algorithm is correct if at least one human evaluator has selected the same result among the similar items. In more formal terms, for each query the "ground-truth" is considered to be the set of symbols formed by the union of all human selected sets. Figure 3.10 provides a few examples. For instance, for query *a1*, evaluators have provided a list of symbols which they consider visually close, or containing parts that are visually close. The evaluators were not required to provide any ranking order nor degree of visual resemblance.

Evaluation Metric Our aim is not only limited to distinguish symbols but also extended to rank symbols in the provided lists. Ranking is related to similarity based on distance measure as described in Section 4.5.2. It is important to notice that the number of ground-truths varied from one query to another. Therefore traditional precision and recall cannot be used. We use retrieval efficiency [Kankanhalli et al., 1995] as the evaluation metric. For every chosen query, retrieval efficiency for a given short-list of size K is expressed as,

$$\eta_K = \begin{cases} n/N & \text{if } N \leq K \\ n/K & \text{otherwise,} \end{cases}$$

where n is the number of returned relevant symbols, N the total number of relevant symbols and K the number of ranked symbols requested. Note that η_K computes the traditional recall if $N \leq K$ and computes precision otherwise. The main advantage of this is that the average retrieval efficiency curve is not biased even with different ground-truths for different queries while it happens for precision measure when $N < K$. The more explanation will be discussed in Appendix A.

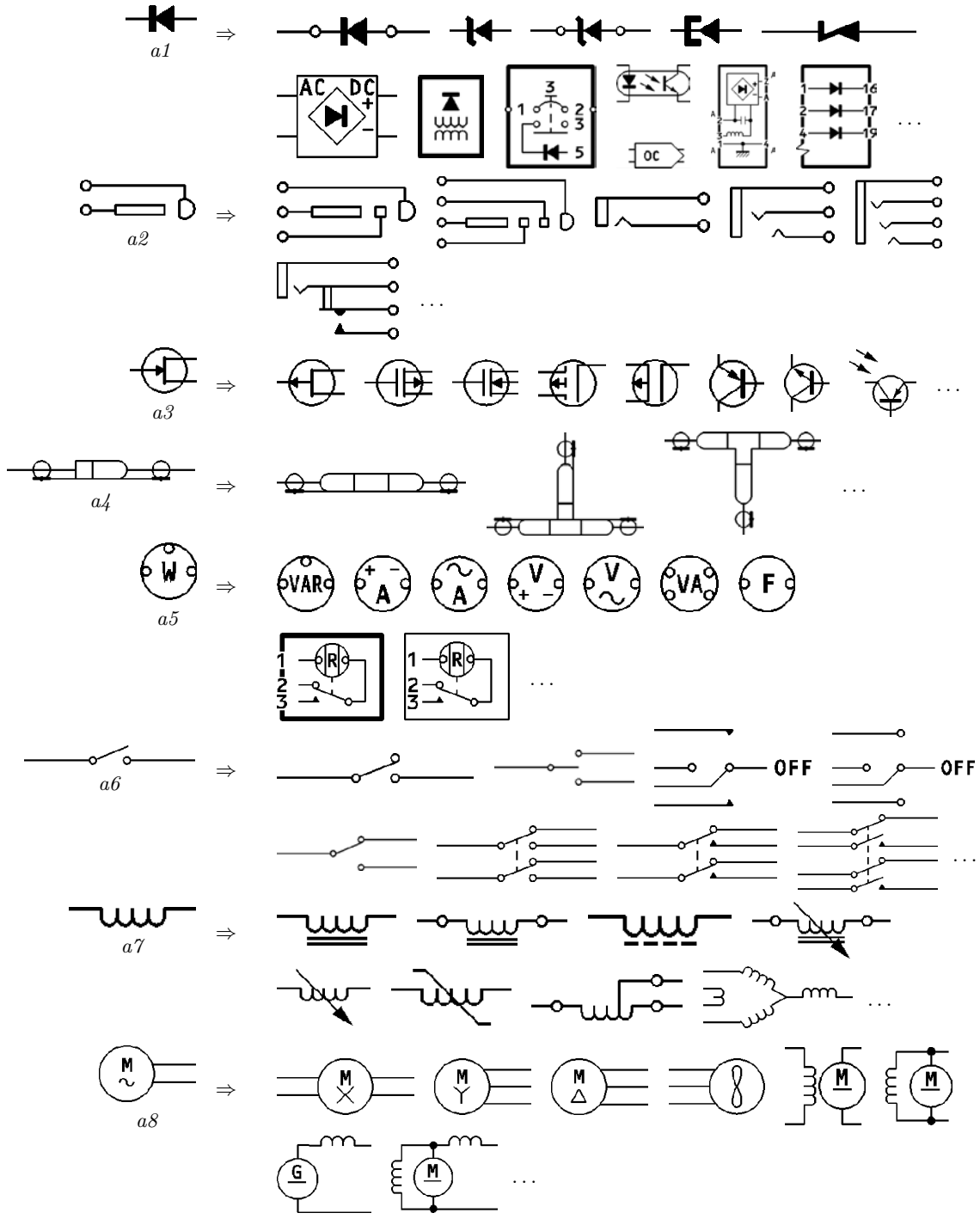


Figure 3.10: A sample of few electrical symbols and their similar sample images. For every test symbol: *a1* to *a8*, a few relevant symbols are enlisted based on human evaluation. It consists of both linear as well as symbols in the composite form.

Observations Average retrieval efficiency, over requested retrieval list – 1 to 10 is shown in Table 3.1. From state-of-the-art of shape descriptors, GFD performs better, which is followed by SC and ZM one after another. In contrast, $\mathcal{D}-$Radon outperforms for different values of bins like

$B = 180, 90, 60$. The performance of $\mathcal{D}-$Radon using 36 number of bins can also be compared since it provides marginal difference with 'the best' performer i.e., GFD from state-of-the-art.

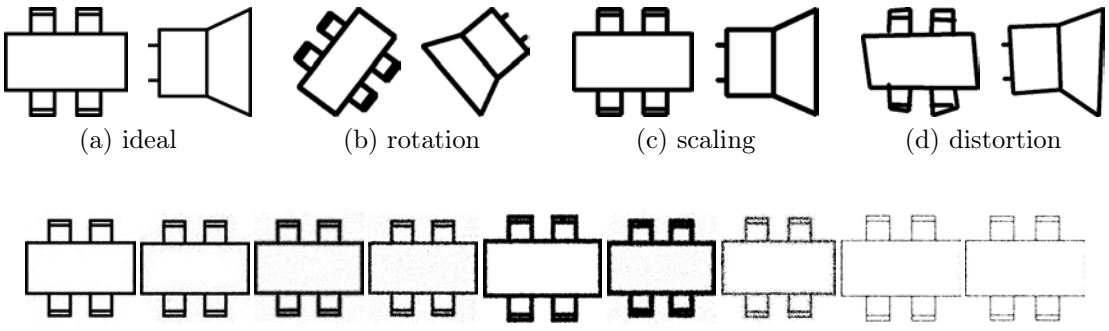
Requested List	GFD	ZM	SC	$\mathcal{R}-$sign.	$\mathcal{D}-$Radon for different values of B 180	90	60	36	18	09	02
top-2	91	88	87	84	**92**	**92**	91	87	78	77	75
top-4	80	72	72	71	**83**	82	81	76	66	62	60
top-6	74	65	63	60	**77**	75	74	69	57	53	50
top-8	71	60	59	51	**76**	75	73	64	49	45	45
top-10	69	56	54	49	**73**	71	69	58	44	42	41

Table 3.1: Retrieval efficiency (in %) over 50 queries for FRESH dataset.

GREC dataset

In this dataset[3] [GREC, 2003], we have used the following different categories: ideal, rotation, scaling, distortion as well as degradation.

Altogether, there are 50 different model symbols. Those symbols are grouped into 3 sets, containing 5, 20 and 50 model symbols. Each model symbol has 5 test images in every category except the ideal one. Ideal test images are directly taken from the set of model symbols and therefore the test is to evaluate the ability of simple shape discrimination, as the number of symbols increases. Since vectorial distortion works only with symbols with straight lines, and not arcs, it is applied to a subset of 15 model symbols. Besides, there are 9 models of degradation, aiming to evaluate the robustness to the scalability with degradation. Figure 3.11 shows a few samples of GREC dataset.



(a) ideal (b) rotation (c) scaling (d) distortion

m1 m2 m3 m4 m5 m6 m7 m8 m9

(e) 9 different degradation models (m1 to m9)

Figure 3.11: GREC2003 samples – graphical symbol.

To evaluate the method, each test image is matched with the model symbols to get the closest model. Experimental results for all types of aforementioned categories of datasets are shown in Table 3.2. More precisely, it provides results for ideal, rotation, scaling as well as combination of rotation and scaling. In addition, results for two different sets of vectorial distortions including for degradation samples having nine different models are provided.

Observations Following Tables 3.2, we observe the following.

[3] International symbol recognition contest, 2003

Images Set	GFD	ZM	SC	$\mathcal{R}-$sign.	$\mathcal{D}-$Radon for different values of B 180	90	60	36	18	09	02
ideal	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
rotate	**98**	94	97	94	97	94	88	73	82	77	71
scale	99	98	99	96	**100**	**100**	**100**	**100**	84	74	57
rotate+scale	**98**	93	**98**	92	**98**	97	94	82	79	73	62
distort	**100**	94	**100**	92	**100**	**100**	**100**	**100**	85	72	47
degrade	91	79	78	76	**99**	98	95	84	67	47	35

Table 3.2: Average recognition rates (in %) for four different categories: ideal, rotation, scaling and rotation + scaling (from 50 models) of GREC dataset.

- Based on the results from ideal test images, every shape descriptor provides similar recognition performance. In case of $\mathcal{D}-$Radon, it is interesting to notice that we obtain 100% recognition rate from all provided number of bins. For rotated images, GFD performs better, proving a marginal difference with SC and $\mathcal{D}-$Radon for 180 bins. For scaled images, $\mathcal{D}-$Radon outperforms all, where $B = 180, 90, 60$ and 36 provide 100% recognition rates while not offering substantial difference with the state-of-the-art. For those test images where rotation and scale are combined, $\mathcal{D}-$Radon for 180, GFD and SC provide similar results. Overall, not a single 'the best' performer from the state-of-the-art has been observed. Besides, one cannot judge the superiority of the methods. Only running time comparison would be an alternative.

- Results from test images with vectorial distortions shows the worst case scenario for $\mathcal{R}-$signature while for others, there exists similar behaviour. In case of $\mathcal{D}-$Radon, 36 bins can also be compared with 'the best' performer from the state-of-the-art.

- However, we receive notable differences between $\mathcal{D}-$Radon and GFD, in case of binary degradations. Others follow it. In case of $\mathcal{D}-$Radon, 60 bins can provide competitive results.

Overall, 'the best' performer from the state-of-the-art can be changed from time to time (depending on the dataset) while $\mathcal{D}-$Radon provides consistent results. $\mathcal{D}-$Radon in particular, shows an interesting behaviour for different values of bins.

CVC dataset

As in [Wendling et al., 2008, Fornés et al., 2010], we have tested 10×300 sample images i.e., 10 different known classes of hand-drawn architectural symbols with 300 instances in each. Symbols have been drawn by 10 different people using 'anoto' concept. Figure 3.12 shows a few samples of the dataset. Samples provide distortions, gaps, overlapping as well as missing parts within the shapes.

To validate the methods, each test image is matched with all images and number of correct matches is over the requested list. In this test, since we have 300 samples per class, the size of the requested list is 300 in order to retrieve all similar images of every class including itself. To handle in-depth experimental results analysis, we provide retrieval rate in every 20 increasing step. It is important to notice that the last value of retrieval rate is equivalent to the recognition rate. Table 3.3 shows the average retrieval rate for all requested short-lists (e.g., top-20, top-40 and so on).

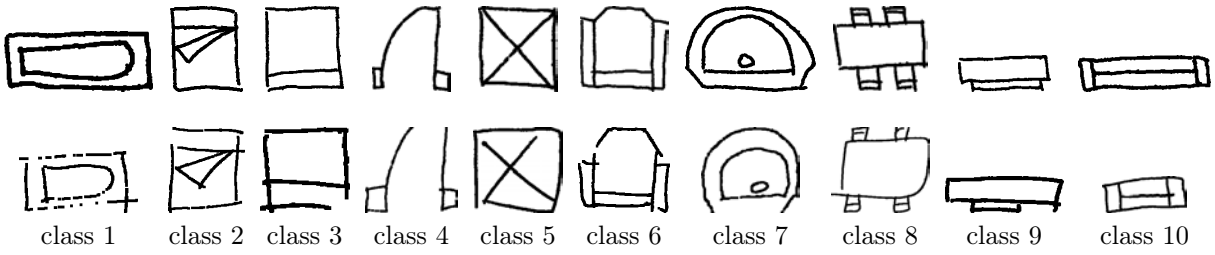


Figure 3.12: 2 hand-drawn samples from 10 different known classes.

Observations Unlike the previous dataset, SC provides the best performance from state-of-the-art. In contrast, $\mathcal{D}-$Radon outperforms 'the best' performer.

Up to top-60, one cannot decide which method performs well since there exist no notable retrieval rate differences among them. It is only determined after top-60. The aim of the test is to evaluate the retrieval stability of the methods. $\mathcal{D}-$Radon provides the rate of more than 16% difference with SC. SC lags GFD by approximately 9%. $\mathcal{R}-$signature provides an average results compared to ZM.

Overall, recognition rates from $\mathcal{D}-$Radon for $B = 180, 90, 60$ and 36 provide interesting results compared to GFD, SC, $\mathcal{R}-$signature and ZM.

Requested List	GFD	ZM	SC	$\mathcal{R}-$sign.	$\mathcal{D}-$Radon for different values of B 180	90	60	36	18	09	02
top-20	96	68	98	82	**99**	**99**	**99**	98	94	92	83
top-40	93	62	95	75	**99**	**99**	98	97	92	88	79
top-60	90	59	95	69	**97**	**97**	**97**	96	90	86	67
top-80	88	57	92	65	**97**	**97**	**97**	94	88	81	64
top-100	85	55	91	62	**97**	**97**	96	94	84	80	61
top-120	83	54	88	59	**97**	96	96	93	80	74	60
top-140	81	52	87	56	**95**	**95**	**95**	92	76	67	58
top-160	78	50	85	54	**95**	**95**	94	91	73	64	57
top-180	76	50	83	51	**93**	**93**	94	89	67	61	54
top-200	73	48	81	49	**93**	92	91	86	62	56	53
top-220	71	44	78	48	**93**	92	91	86	59	53	50
top-240	68	41	78	46	**92**	90	88	84	55	51	49
top-260	66	39	75	45	**91**	89	86	83	52	44	46
top-280	63	37	73	43	**88**	87	85	81	48	42	45
top-300	61	36	70	42	**86**	**86**	84	78	47	39	44

Table 3.3: Average retrieval rate (in %) for CVC dataset.

Experimental Results Analysis

We analyse the behaviour of the methods based on the key characteristics as well as major challenges found in datasets. In general, we focus on those samples which are distorted, embedded with different levels of noise and even degraded. We also take into account those samples with missing parts including severe vectorial distortions in hand-drawn symbol dataset in addition to a significant size variation as well as multi-class similarity between the classes. Within this framework, based on the observations in all experimental tests, we discuss the methods in two

different issues:

1. recognition rate and
2. running time complexity.

Recognition performance is related to how discriminative the feature is. Matching on the other hand, provides running time.

All descriptors perform well except for distorted and degraded samples. In particular, $\mathcal{R}-$signature has been severely disturbed. It is due to the square effect (via $\mathcal{R}-$transform). SC shows almost similar behaviour where the level of noise is high since it takes those pixels into account while sampling. For ZM, we have observed the effect in case of degradation models. GFD provides fairly satisfying results. However, a single shape descriptor cannot be always 'the best' performer for all types of sample images. $\mathcal{D}-$Radon in contrast, shows discriminative power.

$\mathcal{D}-$Radon is constrained by how many number of bins are used. As said previously, the larger the number of bins, the higher the discrimination power and vice versa. The statement is validated with the help of a series of tests. In all experimental test results, $\mathcal{D}-$Radon provides competitive recognition and retrieval performances even when the decrement of number of bins upto 60. 36 number of bins can also be compared with state-of-the-art of shape descriptors. It is on the other hand, depend on how complex is the dataset. For instance, only 2 bins (i.e., vertical and horizontal projections) provides 100% recognition rate for ideal sample set in GREC dataset (*cf.* Table 3.2). In the later situation, since execution time is related according to how many number of bins are used, we can possibly reduced the delay.

Running time processing is usually high since it uses DTW for matching. However it also largely depends on how big the image is. As far as computational cost is concerned, the observed average running time for all methods is given in Table 3.4.

Shape descriptors	Time (in sec)
1. $\mathcal{R}-$signature [Tabbone et al., 2006]	02
2. ZM [Kim and Kim, 2000]	17
3. GFD [Zhang and Lu, 2002]	13
4. Shape Context [Belongie et al., 2002]	38
5. $\mathcal{D}-$Radon	71

Table 3.4: Average running time (in sec) for a single pair.

Overall, it is important to notice that state-of-the-art of shape descriptors provide different results according to the behaviour of the dataset i.e., not a single shape descriptor can be 'the best' performer. As a consequence, for comparison, execution time will be an alternative. In contrast, $\mathcal{D}-$Radon outperforms of all 'the best' performers in all types of dataset while paying more execution time. Besides, since experimental tests have been made without any pre-processing, there exists a room to improve further when suitable pre-processing is integrated.

3.4.3 Case Studies

In order to establish the interest as well as extensibility of the method, several different datasets are also employed. More specifically, we mainly discuss in two different contexts.

1. Isolated character recognition and
2. Foot-wear impression evidence verification.

Character Recognition

Several different datasets from different scripts[4] are tested. They are Roman, Bangla, Devanagari and Oriya [Bhowmik et al., 2006, Bhattacharya and Chaudhuri, 2009]. Figure 3.13 shows a couple of sample images in each class.

Figure 3.13: 2 numeral samples from 4 different known scripts for 10 $(0-9)$ classes.

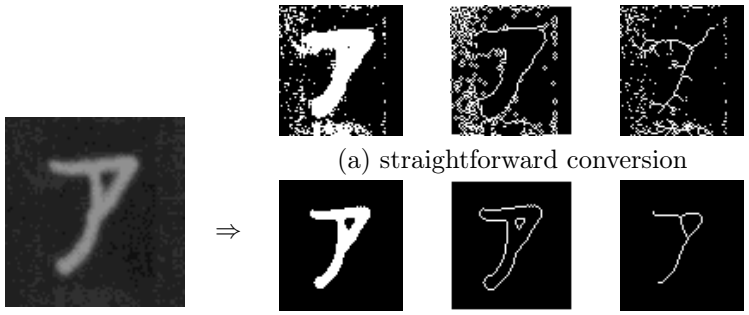


Figure 3.14: An example of binarisation sample from the ETL dataset.

The preliminary task is to do pre-processing since characters tend to be highly degraded as they are taken from newspaper, postal cards etc. under varying different lighting conditions, for instance. It mainly considers stroke synthesis, thresholding, gray-scale to binary conversion; noise removal, foreground textual information extraction by removing background [Alginahi, 2010].

[4]ISI character datasets for Indian scripts, CVPR unit, India
ETL3 Roman dataset, AIST, Japan

However, this introduces many ad-hoc techniques. This work does not aim to develop pre-processing technique. In this work, isolated character images are simply converted to binary. Otsu method has been found to be prominent to handle gray-scale images [Otsu, 1979]. However, it does not suited for all types of sample images used in the paper. In such a case, while converting, an average gray-scale pixel intensity value (in the range of 70–100) is used to make a threshold that goes differently from one dataset to another. Fig. 3.14 shows a few examples of it. In this illustration, binary conversion is followed by contour detection and thinning using basic image processing tools.

While experimenting, every test sample is matched with training candidates and the closest one is reported. The closest candidate corresponds to the labelled class, which we call 'character recognition'.

To evaluate the performance of the methods, $\mathbb{K}-$fold Cross-Validation (CV) has been implemented unlike traditional dichotomous classification. In $\mathbb{K}-$fold CV, the original sample for every class is randomly partitioned into $\mathbb{K}$ sub-samples. Of the $\mathbb{K}$ sub-samples, a single sub-sample is used for validation, and the remaining $\mathbb{K}-1$ sub-samples are used for training. This process is then repeated for $\mathbb{K}$ folds, with each of the $\mathbb{K}$ sub-samples used exactly once. Finally, a single value results from averaging all. The opposite process holds for inverse $\mathbb{K}-$fold CV i.e., $\mathbb{K}-1$ sub-samples are used for testing.

While experimenting, tests move from normal $\mathbb{K}-$fold CV to inverse. This means that every test goes from $\mathbb{K}-1$ to $\mathbb{K}-4$ training sub-samples when $\mathbb{K}=5$. The aim of the use of such a series of rigorous tests is to avoid the biasing of the samples that can be possible in conventional dichotomous classification.

						$\mathcal{D}-$Radon for different values of B						
Dataset	Training	GFD	ZM	SC	$\mathcal{R}-$sign.	180	90	60	36	18	09	02
Roman	$\mathbb{K}-1$	97	83	98	78	**100**	**100**	**100**	88	79	76	71
10×100	$\mathbb{K}-2$	95	78	97	75	**100**	**100**	99	84	75	74	68
	$\mathbb{K}-3$	94	74	96	70	**98**	96	95	79	74	71	66
	$\mathbb{K}-4$	91	67	96	66	**98**	96	94	77	74	68	67
Oriya	$\mathbb{K}-1$	98	44	98	58	**100**	98	92	81	67	60	51
10×100	$\mathbb{K}-2$	96	38	96	52	**98**	97	94	77	64	55	48
	$\mathbb{K}-3$	85	43	93	46	**98**	96	88	74	62	50	47
	$\mathbb{K}-4$	72	32	92	43	**97**	92	90	71	68	49	45
Devanagari	$\mathbb{K}-1$	86	40	96	55	**99**	97	89	79	71	62	52
10×300	$\mathbb{K}-2$	84	40	94	54	**98**	96	87	75	68	59	48
	$\mathbb{K}-3$	81	38	93	50	**98**	94	84	74	66	58	47
	$\mathbb{K}-4$	69	34	87	46	**96**	**95**	83	73	65	55	46
Bangla	$\mathbb{K}-1$	73	47	**95**	48	**95**	**95**	84	72	60	51	41
10×400	$\mathbb{K}-2$	69	44	**94**	48	**94**	92	79	68	55	48	38
	$\mathbb{K}-3$	68	43	91	46	**93**	86	76	61	54	47	37
	$\mathbb{K}-4$	64	40	89	44	**92**	81	75	59	53	45	36

Table 3.5: Average recognition rate in % using $\mathbb{K}-$fold CV (where $\mathbb{K}=5$).

Observations Table 3.5 shows the average recognition rates for all datasets using $\mathbb{K}-$fold CV. In Roman dataset, shape descriptors provide encouraging recognition performances, having almost similar results. The significant differences between them exist in case of Devanagari and Bangla datasets. SC yields consistent recognition rates for all while others do not follow such

a characteristic. GFD however, comes closer to SC. $\mathcal{D}-$Radon is now confronted with SC and sometimes with GFD. In this category, there exists no surprising differences between them i.e., approximately 1–2%. However, it is found that number of bins $B = 90$ for $\mathcal{D}-$Radon, provides better results and $B = 60$ can be compared with.

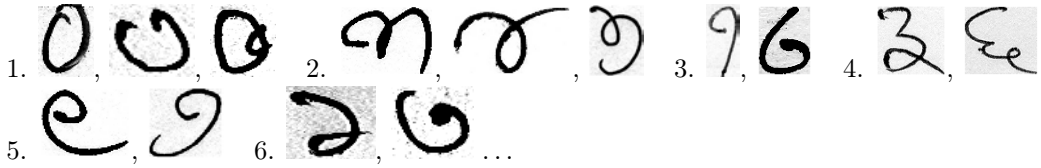


Figure 3.15: Difficulties in character recognition – a few examples.

Let us highlight a few major challenges such as

- multi-class similarity,
- symmetric shape similarity,
- missing parts, and
- stroke length or size variation.

Multi-class similarity is one of the major problems in character recognition. In addition, shape descriptors with rotation invariance properties have been affected from those samples as shown in Figure 3.15. These are the major reasons affecting those existing methods. To avoid this, $\mathcal{D}-$Radon has been tested without integrating estimated orientation angle since isolated characters are almost vertically aligned. In case of samples with missing parts as well as with different stroke length variation, state-of-the-art shape descriptors do not provide optimal performance since feature vector can be changed accordingly. In contrast, $\mathcal{D}-$Radon is not affected, thanks to DTW.

Foot-wear Impression Evidence Verification

It is another case study to see the behaviour of the methods when pattern complexity increases. For example foot-wear evidence verification, where internal structure is very important.

In this dataset[5], we have 64 foot-wear evidence impression or foot-wear print classes, each with height instances. Foot-wear impression images are categorised into full print and partial print. Figure 3.16 shows sample images for a couple of classes. In this illustration, we put a border to every sample in order to clearly determine full prints as well as partial prints in addition to shape and size. As shown in Fig. 3.16, there are four images in the full print category including rotation while the remaining ones are partial prints. Partial prints also contain toe and heel portions.

For immediate analysis, we use straightforward verification for all test images. Every test image is matched with 64 templates to find the best match which corresponds to the target foot-wear print.

[5]Thanks to Signal Processing Laboratory, Kinki University, Japan for providing this dataset.

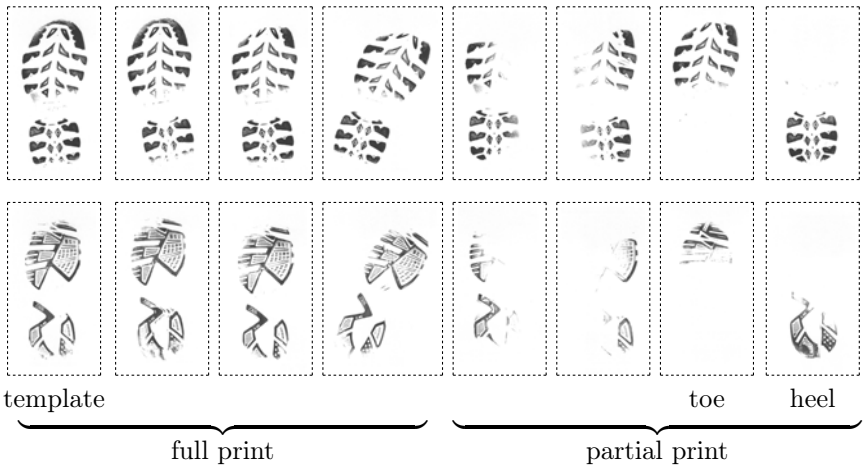


Figure 3.16: 8 samples of 2 different known footwear print classes.

Observations Global signal-based (state-of-the-art shape) descriptors for foot-wear impression evidence verification do not provide interesting results. This is due to insufficient extraction of local information. However, we employ this dataset as a case study. An average recognition rate is provided in Table 3.6. In all categories of test samples, state-of-the-art shape descriptors do not provide satisfactory recognition performance. In contrast, $\mathcal{D}-$Radon yields the highest recognition rates – all provided number of bins can be compared with.

Images Set			GFD	ZM	SC	$\mathcal{R}-$sign.	$\mathcal{D}-$Radon for different values of B 180	90	60	36	18	09	02
Full print		64×4	65	45	55	46	**81**	78	74	73	61	55	54
Partial print	heel	64×1	18	19	17	03	**44**	43	42	35	29	27	24
	toe	64×1	09	08	03	03	**35**	34	33	32	22	17	12
	other	64×2	11	17	04	07	**56**	52	52	41	31	29	22

Table 3.6: Average recognition rate (in %) for full print as well as partial print.

3.5 Conclusions

In this paper, we have introduced a method based on dynamic programming for matching the Radon features, for graphics recognition. The key characteristic of the method is to use DTW algorithm to match corresponding pairs of the Radon features for all possible projections. Since features are not normalised into a fixed length vectors, dynamic programming has been used to absorb the varying Radon features sizes resulting from image signal variations from shape and sizes. In order to make correct matching between the Radon features, we have integrated the estimated orientation value such that it makes rotation invariant.

We have attested the method for graphics recognition using several different datasets and compared with the state-of-the-art of shape descriptors. Furthermore, two different case studies have been made: character recognition and foot-wear impression evidence verification. Based on this framework, we plan to proceed the following: one-to-all feature matching – aim for highly degraded, distorted as well as deformed patterns (*cf.* Section 3.3.2). To handle this, we will

integrate the optimised DTW as presented in [Lemire, 2009]. Besides, the latter work is aimed to extend upto crime scene investigation in addition to the use of local descriptors as well as bag-of-features (BOFs) approach using shape descriptors.

3.6 What is Next?

In this chapter, graphics recognition has been made via global signal–based shape descriptors. The methods are found to be appropriate for isolated pattern of any kind. However, these are not always suitable for solving real–world problems where both isolated as well as composite symbols are found to be occurred. This is happened in FRESH dataset. In such a case, we can take advantage of structural approaches. In the next chapter, we will describe the symbol by using spatial relations between the extracted visual primitives from the symbol.

End of Chapter

Keywords

Vocabulary,
Spatial Relations,
Attributed Relational Graph,
Symbol Recognition.

4

Symbol Recognition using Spatial Relations

Contents

4.1 Organisation of the Chapter 92
4.2 Introduction 92
4.2.1 Related Work 92
4.2.2 Outline of the Proposed Recognition Method 94
4.3 Spatial Relations 94
4.3.1 Unique Reference Point Set 95
4.3.2 Directional Relations 95
Projection Model 96
Radial line Model 99
4.4 Symbol Description 103
4.4.1 Visual Vocabulary 104
4.4.2 Pairwise Spatial Relations 104
4.4.3 Attributed Relational Graph 106
4.5 Symbol Recognition 107
4.5.1 Matching 107
4.5.2 Ranking 109
4.6 Experiments 109
4.6.1 Dataset and Evaluation Metric 110
4.6.2 Matching Scope 110
4.6.3 Experimental Results 110
Our Method 111
Other Spatial Relation Models 112
Global Signal-based Descriptors 113
Pixel-based Approaches 113
Comparison 114
4.6.4 Experimental Results Analysis 115

4.6.5 Advantages and Limitations 117
4.7 Conclusions 118
4.8 What is Next? 118

Foreword. In the previous contribution, we have introduced global signal-based shape descriptor and validated over several different datasets in order to see how discriminate it is, in comparison to a major set of state-of-the-art of shape descriptors. The method has been applied to isolated symbol recognition. Such a method fails to handle real-world applications where symbols are connected with other graphical elements and texts. Such a situation will be handled by extracting visual primitives that compose the symbol so that visual relations between them can be used for recognition.

In this chapter, we present a method for symbol recognition based on the spatio-structural description of a 'vocabulary' of extracted visual elementary parts. The method consists of first identifying vocabulary elements and placing them into different groups based on their types (e.g., *circle*, *corner*). We then compute spatial relations between the possible pairs of labelled vocabulary types which are further used as a basis for building an attributed relational graph (ARG) that fully describes the symbol. To validate the method, it is applied to symbols in electrical wiring diagrams. The experiments reported in this chapter show that this approach, used for recognition, significantly outperforms both structural and statistical state-of-the-art methods. ■

4.1 Organisation of the Chapter

The chapter is structured as follows. We start with an introduction in Section 4.2 which mainly includes our motivation, describes the underlying problem, reviews pertinent literature and give an outline of our proposed method. Since our recognition method is mainly based on spatial relations, we develop a method that describes spatial relations in Section 4.3 and the derive symbol representation method form it in Section 4.4. We explain a symbol matching method from it and explain it in Section 4.5. Full experiments are reported in Section 4.6 and confront our method with current state-of-the-art algorithms. It includes a comprehensive experimental result analysis. We point out the summary of the work in Section 4.7.

4.2 Introduction

As already presented in Chapter 2, research on graphics recognition has an extremely rich state-of-the-art literature, aimed to localise/recognise symbols depending on the applications. According to [Cordella and Vento, 2000b, Lladós et al., 2002], these methods are particularly suited for isolated line symbols, not for composed symbols connected to a complex environment. In order to exploit the information embedded in those documents, one needs to be able to extract visual primitives and formalise the possible links that exist between them. This combination of symbol localisation based on extracted visual primitives is going to be the core of this chapter.

4.2.1 Related Work

We pointed out in Chapter 2 that existing approaches, can roughly be sorted into three approaches: statistical, structural and hybrid. As respective examples, among others, one can cite [Cordella and Vento, 2000a, Lladós et al., 2001, Yang, 2005, Zhang et al., 2006]. The chapter is mainly focussed on structural approaches. In this section, we put a brief idea from the comprehensive explanation of structural approaches mentioned in section 2.2.2.

In structural approaches, the principle idea is to decompose the symbols into either vector based primitives like points, lines and arcs or into meaningful parts like *circles*, *triangles* and *rectangles*. These primitives are then used for representing symbols in terms of attributed relational graphs (ARG) [Bunke and Messmer, 1995, Conte et al., 2004], region adjacency graphs (RAG) [Lladós et al., 2001], constraint networks [Ah-Soon and Tombre, 2001] as well as deformable templates [Valveny and Martí, 2003]. Their common drawback comes from error-prone raster-to-vector conversion. Those errors can increase confusions among different symbols. Furthermore, variability in the size of the graphs leads to computational complexity in matching. However, structural approaches provide a powerful representation, conveying how parts are connected to each other, while also preserving generality and extensibility.

To describe the symbols, it is necessary to handle relations between the decomposed parts. The following paragraph gives an overview of existing work on spatial relations and their proper usages.

As explained in Section 2.2.2, effect of spatial relations on recognition performance have been examined comprehensively for scene understanding [Bar and Ullman, 1993], document analysis and recognition tasks [Garnesson and Giraudon, 1990, Pham and Smeulders, 2006]. Spatial relations can be either topological [Egenhofer and Herring, 1991, Renz and Nebel, 1998], directional [Matsakis and Wendling, 1999, Mitra, 2002] and metric in nature, their choice depending on the kind of application. For example, in [Xiaogang et al., 2004], topological configurations are handled with a few predicates like *intersection*, *interconnection*, *tangency*, *parallelism* and *concentricity* expressed with standard topological relations as described in [Egenhofer and Herring, 1991].

In a similar way, various directional relation models have been developed for a wide range of different situations.

- If the objects are far enough from each other, their relations can be approximated by their centres based on the discretised angle [Miyajima and Ralescu, 1994, Mitra, 2002]. This approach is robust to small variations of shape and size of the studied objects.
- If they are neither too far nor too close, relations can be approximated by their minimum bounding rectangle (MBR) as long as they are regular [Peuquet and CI-Xiang, 1987, E.Jungert, 1993, Papadias and Theodoridis, 1997]. In other words, the quality of the MBR depends on compactness[6] of the tile.
- Approaches like angle histograms [Wang and Keller, 1999] tend to be more capable of dealing with overlapping, something the previous approaches have difficulties with. However, since they consider all pixels, their computational cost increases dramatically.
- Other methods, like for instance, F-Histograms [Matsakis and Wendling, 1999] use pairs of longitudinal sections instead of pairs of points, also at the cost of high time complexity.
- Another well-known approach uses fuzzy landscapes [Bloch, 1999], and is based on fuzzy morphological operators.

Previously mentioned approaches address only either topological or directional relations. Managing both comes at high computational costs. Even then, no existing model fully integrates topology. They rather have various degrees of sensitivity to or awareness of topological relations. While methods like [Xiaogang et al., 2004] focus on topological information only, the approach we develop in this thesis unifies both topological and directional information into one descriptor [K.C. et al., 2009c] without any additional running time cost.

[6] Compactness $= \frac{Area(\mathbb{A})}{Area(MBR(\mathbb{A}))}$

Placing spatial relations in the context of recognition and symbol description, one should note that spatial relations also have a language-based component (related to human understanding e.g., to the *right* of) that can be formalised in a mathematical way (e.g., the 512 relations of the 9−intersection model [Egenhofer and Herring, 1991]). Therefore, qualitative and quantitative relations are another way to do categorisation of spatial relations. As an example, consider an object $\mathbb{A}$ extending from *right* (98%) to *top* (2%) with respect to $\mathbb{B}$ is expressed as *right* – $top(\mathbb{A}, \mathbb{B})$. This spatial predicate remains unchanged upto a reasonable change of the objects' shape and position. Taking this into account, our work uses more natural relations than the all-or-none nature of standard relations [Freeman, 1975].

In the following section, we explain our proposed method by focusing on using spatial relations for describing and matching symbols.

4.2.2 Outline of the Proposed Recognition Method

Our recognition method is based on the spatial relations between the extracted visual primitives that compose a symbol. The proposed spatial relations are explained in Section 4.3.

We compute the spatial relations (see Section 4.4.2) between the distinct labelled attributes for building an ARG (see Section 4.4.3), achieving at the same time integration of both topological and directional information. Figure 4.1 shows a block-diagram to illustrate the overall proposed idea for symbol description and recognition.

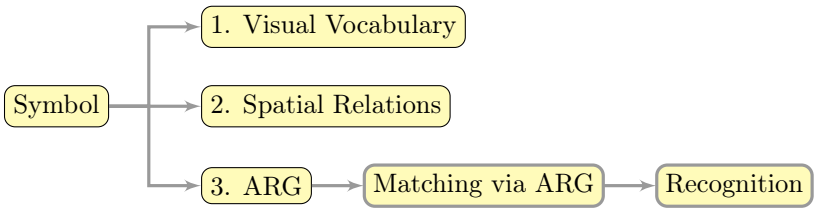


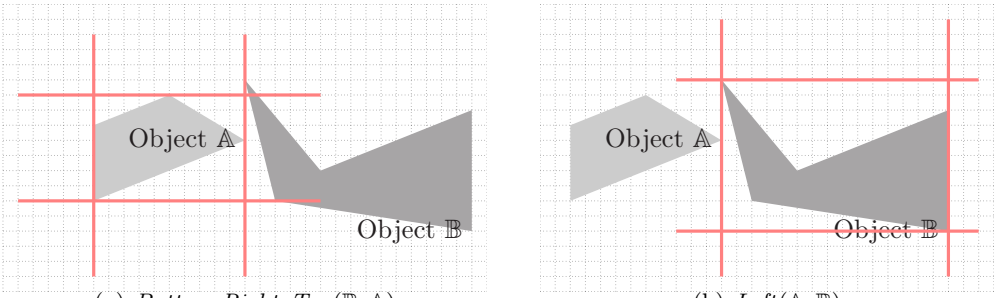
Figure 4.1: An architecture for Symbol Description and Recognition using spatial relations. It uses ARG based symbol description using visual vocabulary and their possible pairwise relations. Those relations are used for matching.

In our proposed method, what is interesting to notice can be summarised as follows. Without any other consideration, it is obvious that the size of the resulting relational graph is potentially very large and variable from one symbol to another. However, when grouping visual primitives together by their types (e.g., *circle*, *corner*) and by labelling them accordingly (see Section 4.4.1), we can eliminate all the combinatorial problems inherent to graph matching, without sacrificing recognition quality or expressive power. We avoid the general NP-hard sub-graph matching problem by having uniquely labelled nodes. Our ARG matching is therefore done in near-constant time. Unlike the existing methods, our method is therefore able to retrieve isolated, as well as known or significant parts of the line symbols embedded in others. Before explaining the aforementioned claims and supporting them by using experiments, in what follows, we first introduce spatial relation models.

4.3 Spatial Relations

Pairwise spatial relations are often expressed by using one of the objects as reference. For example, $\mathbb{A}$ is to the *right* of $\mathbb{B}$: $right(\mathbb{A}, \mathbb{B})$, where $\mathbb{B}$ is referenced. Let us take an example as shown in Figure 4.2, to illustrate the possible ambiguity whenever a correct reference is not

provided. In this illustration, we have taken a classical MBR theory as a spatial relation model in order just to demonstrate that the proper reference can be either $\mathbb{A}$ or $\mathbb{B}$.



(a) *Bottom:Right–Top*$(\mathbb{B}, \mathbb{A})$ (b) *Left*$(\mathbb{A}, \mathbb{B})$

Figure 4.2: Asymmetry of spatial predicates in the case of MBR.

To avoid such a difficulty, we first set up a unique reference point set from each pair. Then, we compute directional relations with respect to the reference point set.

4.3.1 Unique Reference Point Set

We consider a unique reference set $\mathbb{R}$, defined by the topology of the MBR of $\mathbb{A}$ and $\mathbb{B}$ and with the help of the 9−intersection model [Egenhofer and Herring, 1991]. Figure 4.3 shows the different configurations of $\mathbb{R}$ that can occur.

To do this, we simply check topological relations between them in a 9−dimensional binary space via the use of intersections of the boundaries ($\partial\mathbb{X}$), interiors ($\mathbb{X}^o$) and exteriors ($\mathbb{X}^-$) of two sets $\mathbb{A}$ and $\mathbb{B}$. The topological configuration Topology$(\mathbb{A}, \mathbb{B})$ is a vector in this space in which components equal 0 if the corresponding intersection is empty, and 1 otherwise, as shown here:

$$\text{Topology}(\mathbb{A}, \mathbb{B}) = \begin{bmatrix} A^o \cap B^o & \mathbb{A}^o \cap \partial B & A^o \cap B^- \\ \partial A \cap B^o & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^o & A^- \cap \partial B & A^- \cap B- \end{bmatrix}.$$

Therefore 3×3 binary signatures are:

$$\begin{aligned} &\textit{Disconnected}(A, B) = \left[\begin{smallmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{smallmatrix}\right], \\ &\textit{Externally Connected}(A, B) = \left[\begin{smallmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{smallmatrix}\right], \\ &\vdots \\ &\textit{Equal}(A, B) = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right]. \end{aligned}$$

Then, in connection with [Renz and Nebel, 1998], and as shown in Figure 4.3, $\mathbb{R}$ is either the common portion of two neighbouring sides in the case of *disconnected* MBRs or the intersection in the case of *overlapping*, *equal* or otherwise *connected* MBRs. Depending on the obtained topological configurations, $\mathbb{R}$ can range from a point to a rectangular $2D$ area.

4.3.2 Directional Relations

Using $\mathbb{R}$, we propose two different ideas to generate directional relations. They are

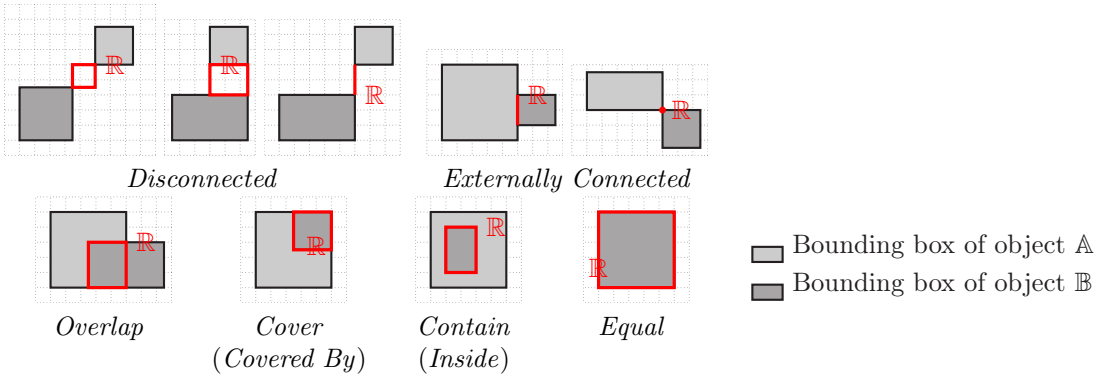


Figure 4.3: Reference point set $\mathbb{R}$ via topological relations from a pair.

1. Projection model and
2. Radial line model.

These two methods differ in the way we partition the image space. In case of the projection model, $\mathbb{R}$ is used to partition the image space bys using orthogonal projection of the boundaries. In radial line model, on the other hand, we obtain several orientation spaces by rotating a radial line using predetermined rotation step, form a pivotal point within $\mathbb{R}$. Each spatial object is represented by one or more representative points, and the relations are determined with the help of the aforementioned partitioned spaces where the object representative points are found.

Projection Model

Given a unique reference point set $\mathbb{R}$, cardinal Relations, based on projection are designed for developing 9−directional relations (predicates):

$$\begin{bmatrix} \text{Left Top (LT)} & \text{Top (T)} & \text{Right Top (RT)} \\ \text{Left (L)} & \text{Middle (M)} & \text{Right (R)} \\ \text{Left Bottom (LB)} & \text{Bottom (B)} & \text{Right Bottom (RB)} \end{bmatrix}.$$

Computing relational matrix $\mathcal{M}$ is just to verify whether the part of the object $\mathbb{X}$ is found to be present in one of the directional spaces made by $\mathbb{R}$. Such a projection concept is completely related to conventional MBR model.

Compared to conventional MBR, our model uses $\mathbb{R}$ and its centroid. Figure 4.4 shows an illustration where the use of centroid point provides directional space splitting. In this illustration, various models are presented in order just to provide relational matrices, without taking spatial objects into account.

- $2D$ projection: Compared to the use of MBR, 'Middle' is sub–divided into four regions i.e., Middle $= \begin{bmatrix} \text{M1 M2} \\ \text{M3 M4} \end{bmatrix}$. Similarly, 'Right', 'Left', 'Top' and 'Bottom' are sub–divided into two: Left $= [\text{L1 L2}]$, Right $= [\text{R1 R2}]$, Top $= [\text{T1 T2}]$ and Bottom $= [\text{B1 B2}]$.
- $1D$ projection: Since it is just a line projection, a few directional spaces are omitted compared to $2D$ projection. However, it brings sub-division of 'Top' and 'Bottom' in case of horizontal line while 'Left' and 'Right' with vertical line.
- $0D$ projection: A single point projection provides the standard four directional spaces.

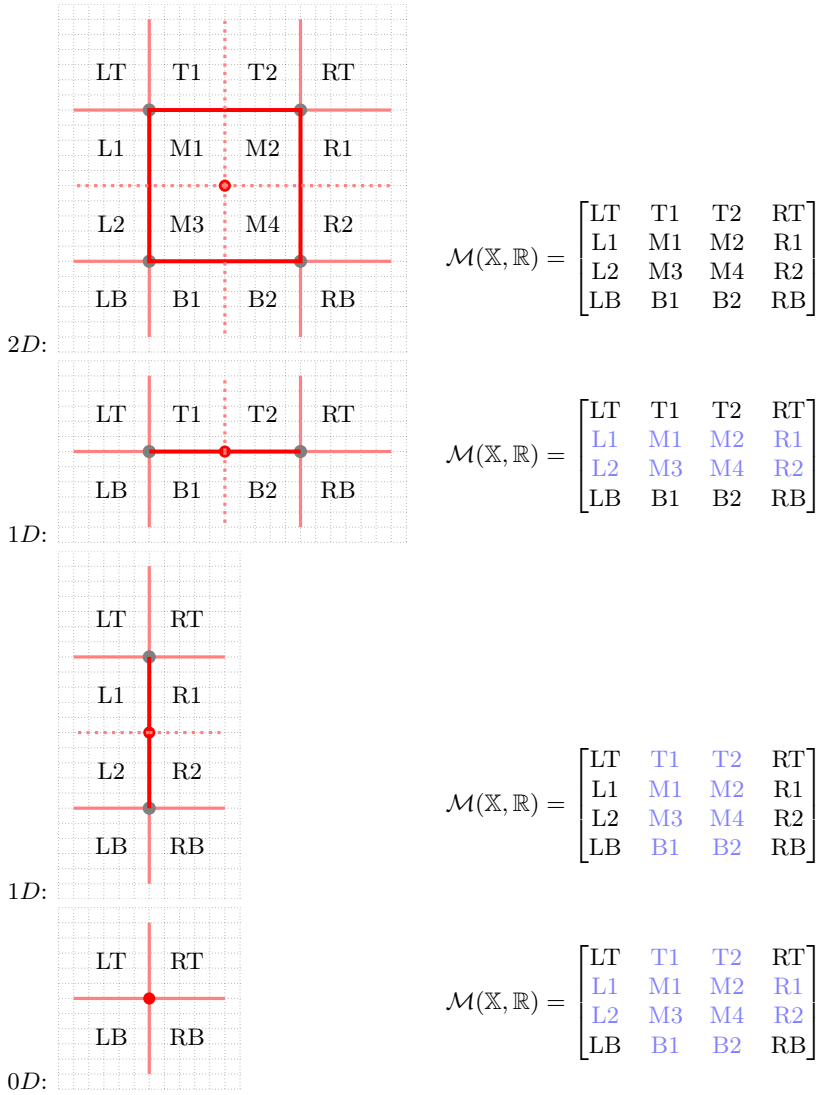


Figure 4.4: Projection models for all possible types of $\mathbb{R}$ developed in Figure 4.3. Relational matrices are expressed following the corresponding models. The dimmed elements of the relational matrix $\mathcal{M}$ refer to values equal to 0.

Spatial relation $\Re$ between $\mathbb{A}$ and $\mathbb{B}$ is

$$\Re(\mathbb{A},\mathbb{B}) = \{\mathcal{M}(\mathbb{A},\mathbb{R}), \mathcal{M}(\mathbb{B},\mathbb{R})\}.$$

Parameter, Precision and Computational Complexity The previously described projection model does not require any parameters. The model takes running time less than two seconds for generating relational matrices as well as matching. To provide behaviour of the computational complexity, we have taken the toy example of image size 288×512 pixels, from Figure 4.2 in page 95.

Figure 4.5 shows the execution time of the model in the function of objects size. It provides for both refined (actual proportion) and boolean (presence or absence i.e., 1 or 0) relations. In this illustration, we have scaled images with the step of +2 i.e., reference image is scaled by $\{2, 4, \ldots, 10\}$.

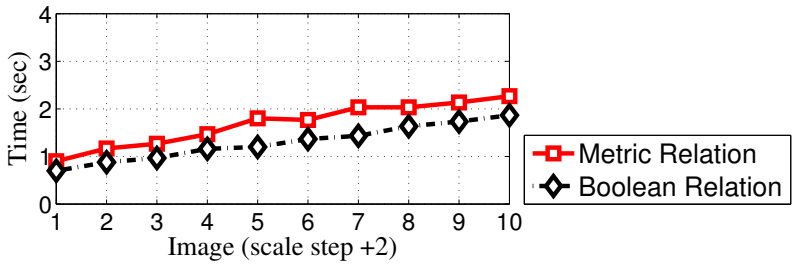
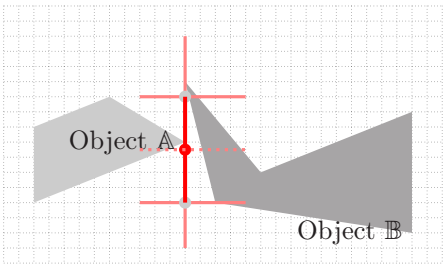


Figure 4.5: Behaviour of computational time for boolean and metric relation matrices. Image is scaled by $\{2, 4, \ldots, 10\}$, with the step of $+2$.

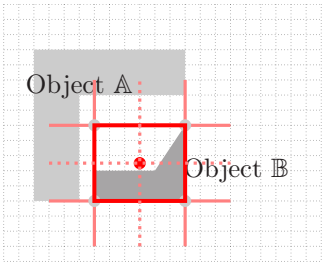
(a) from Figure 4.2



Conventional Model: $\mathcal{M}(\mathbb{A}, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and $\mathcal{M}(\mathbb{B}, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 0.005 \\ 0 & 0 & 0.880 \\ 0 & 0 & 0.115 \end{bmatrix}$

Our Model: $\mathcal{M}(\mathbb{A}, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.45 & 0 & 0 & 0 \\ 0.55 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ and $\mathcal{M}(\mathbb{B}, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 0 & 0.005 \\ 0 & 0 & 0 & 0.310 \\ 0 & 0 & 0 & 0.570 \\ 0 & 0 & 0 & 0.115 \end{bmatrix}$

(b)



Conventional Model: $\mathcal{M}(\mathbb{A}, \mathbb{R}) = \begin{bmatrix} 0.35 & 0.35 & 0 \\ 0.30 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and $\mathcal{M}(\mathbb{B}, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Our Model: $\mathcal{M}(\mathbb{A}, \mathbb{R}) = \begin{bmatrix} 0.35 & 0.175 & 0.175 & 0 \\ 0.150 & 0 & 0 & 0 \\ 0.150 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ and $\mathcal{M}(\mathbb{B}, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0.220 & 0 \\ 0 & 0.315 & 0.465 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Figure 4.6: Illustration of relational matrices using $\mathbb{R}$ (see Figure 4.3) for both objects $\mathbb{A}$ and $\mathbb{B}$. Both conventional and extended projection model are used to notify their differences in terms of precision.

Illustrations Figure 4.6 provides an illustration using the reference point set $\mathbb{R}$. This illustration is based on the toy examples in order just to provide an intuitive feeling on how they behave. It provides comparison between conventional and our projection model.

In Figure 4.6, the examples are provided with proportion of the spatial objects in the prede-

termined partitioning space, instead of not just saying its presence and absence. These matrices are refined relational matrices. As said before, our model provides 4×4 relational matrix compared to 3×3 from conventional MBR. Let us consider Figure 4.6 (b), where $\mathcal{M}(\mathbb{B}, \mathbb{R})$ provides notable difference in terms of precision: conventional model conveys that object $\mathbb{B}$ is completely in the middle, while our model conveys actual precision by providing its positioning only in three partitioned spaces.

Radial line Model

Following the unique reference point set $\mathbb{R}$ in Figure 4.3 (in page 96), we take its centroid point $\mathbb{R}_{p_c}$ as our reference point for computing spatial relation $\Re$ between $\mathbb{A}$ and $\mathbb{B}$ i.e., $\Re(\mathbb{A}, \mathbb{B})$. For a given reference point $\mathbb{R}_{p_c}$, we cover the surrounding space at m regular angular intervals of $\Theta = 2\pi/m$. As shown in Figure 4.7, a radial line rotates over a cycle, and when intersecting with object $\mathbb{X}$, generates a boolean histogram $\mathcal{H}$,

$$\mathcal{H}(\mathbb{X}, \mathbb{R}_{p_c}) = [I(\mathbb{R}_{p_c}, j\Theta)]_{j=0,\ldots,m-1}$$

where

$$I(\mathbb{R}_{p_c}, \theta_i) = \begin{cases} 1 & \text{if } line(\mathbb{R}_{p_c}, \theta_i) \cap \mathbb{X} \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

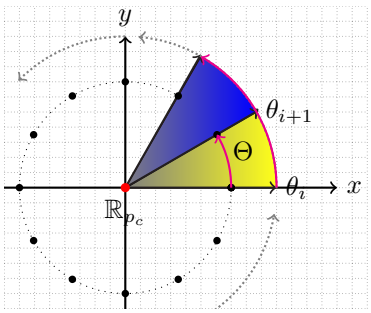


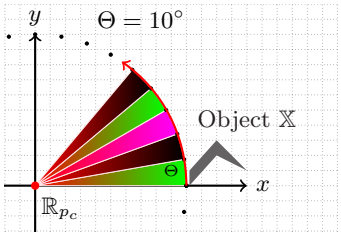
Figure 4.7: Radial line rotation ↻ from a given reference point $\mathbb{R}_{p_c}$.

This boolean histogram expresses whether there are any black pixels in direction θ_i. We extend this binary direction histogram to a histogram covering sectors defined by two successive angle values: θ_i and θ_{i+1} and is normalised with respect to the total area of the studied object such that $\sum \mathcal{H}(\cdot) = 1$. As in the case of the relational matrices described before (*cf.* projection model in Section 4.3.2), rather than using just boolean values, we can account for the percentage of pixels of the whole object lying in the sector defined by θ_i and θ_{i+1}. Figure 4.8 gives an example for both types of histogram, boolean and percentage.

Our relational signature $\Re(\mathbb{X}, \mathbb{R}_{p_c})$ is the set of both histograms:

$$\Re(\mathbb{X}, \mathbb{R}_{p_c}) = \{\mathcal{H}(\mathbb{A}, \mathbb{R}_{p_c}), \mathcal{H}(\mathbb{B}, \mathbb{R}_{p_c})\}.$$

Overall, our method captures the spatial information by the angular positions in the histogram. The magnitude of the histogram contains the structural information.



Histograms:
boolean $\mathcal{H}(\mathbb{X}, \mathbb{R}_{p_c})$ = [1 1 0 0 0 ... 0]$_{1\times 36}$
metric $\mathcal{H}(\mathbb{X}, \mathbb{R}_{p_c})$ = [0.79 0.21 0 0 0 ... 0]$_{1\times 36}$

Figure 4.8: Relational histogram using radial line rotation.

Parameter, Precision and Computational Complexity Note that our method does not only consider the visibility (as defined in [Bloch, 1999]) angle from a reference point $\mathbb{R}_{p_c}$ but also the percentage of pixels in every sector. Figure 4.9 shows the visibility angle α from a given reference point $\mathbb{R}_{p_c}$. In this illustration, although the shape and size of $\mathbb{X}$ change, visibility angle values do not change. In contrast, our method generates different histograms if the structure of $\mathbb{X}$ changes. To explain this in more detail, let us first discuss the rotation step parameter for the radial line.

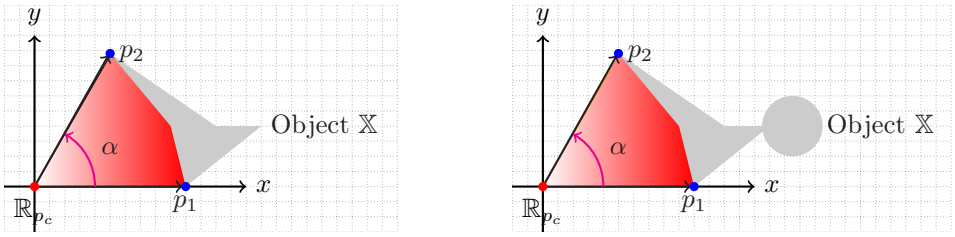


Figure 4.9: Visibility angle values do not change even when objects shape changes, given a reference point $\mathbb{R}_{p_c}$.

While rotating the radial line, one has to select the parameter i.e., *rotation step* Θ. The *rotation step* is simply the resolution at which angular histogram is computed. Figure 4.10 provides an example. In this illustration, we aim to provide the effect of resolution on to the precision. It clearly illustrates that smaller resolution is the appropriate choice even when there exists no difference in visibility angle α (see Figure 4.9) nor does the boolean $\mathcal{H}$. Its value is a trade-off between precision and execution time.

As in projection model (*cf.* Section 4.3.2), to provide behaviour of the computational complexity, we have taken the toy example of image size 288×512 pixels, from Figure 4.2 in page 95. To handle this, a few sets of resolutions: $\{1°, 2°, 3°, \ldots, 10°\}$ are taken and shown in Figure 4.11 for both boolean as well as metrical models. In both cases, it is found that the running time changes with a small increment (but not linearly) when objects size increases. In contrast, it does not increases linearly with the size of the spatial objects as in the angle histogram approaches [Miyajima and Ralescu, 1994], for instance.

Illustrations Until now, we have explained the working principle of the radial line model. In the following, we provide illustrations using toy examples. Those illustrations are not intended to give a full and formal evaluation of our approach, but rather to provide the user with an intuitive feeling on how they behave. In Section 4.4 we take real-world examples and full experiments on real-images are reported in Section 4.6.3.

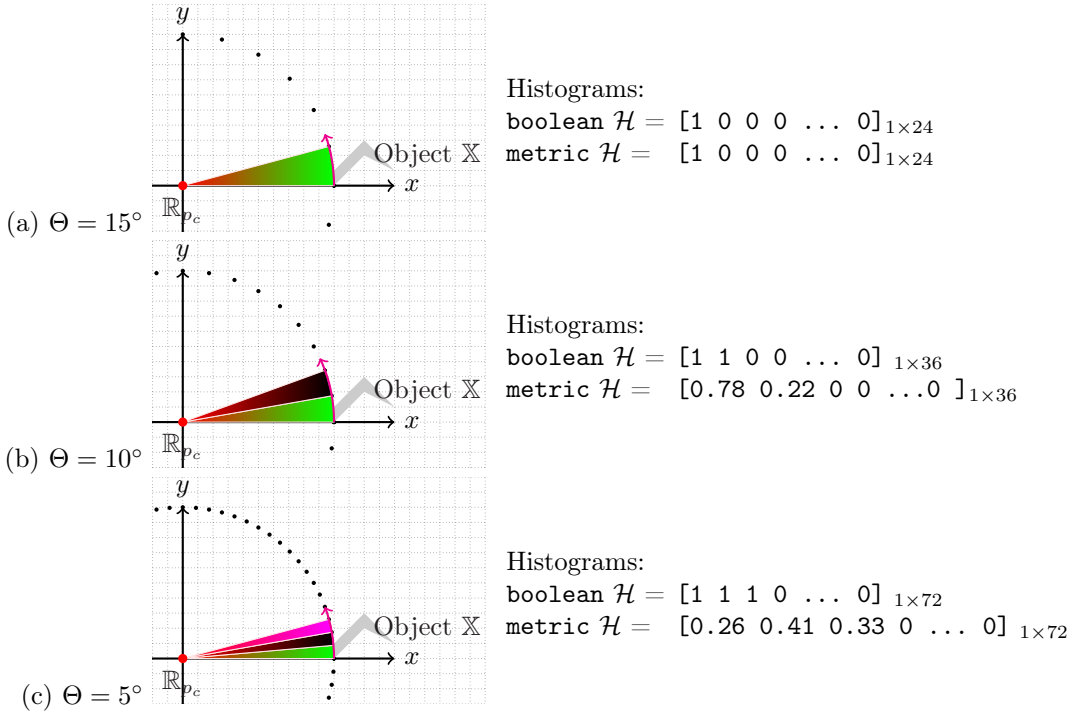
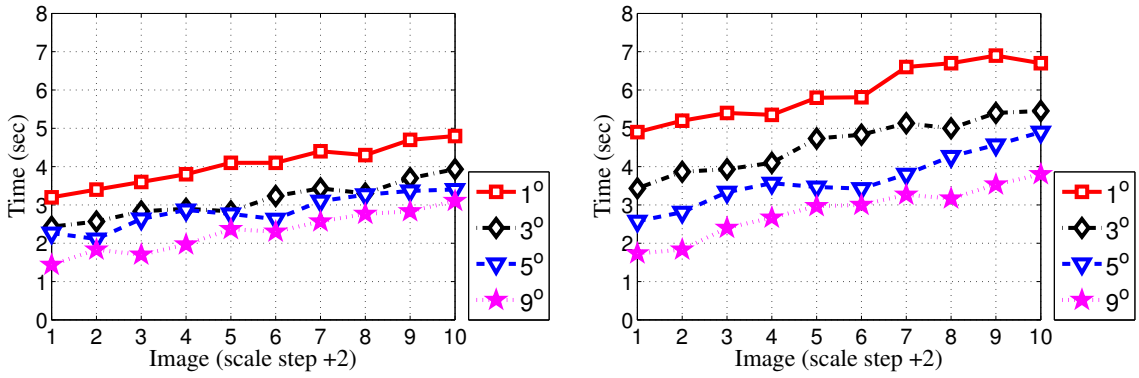


Figure 4.10: Different radial line resolutions effect precision. The smaller the Θ, the higher the precision. In boolean signature, 1*s* represent the presence of 𝕏. Metrical information provides the percentage.

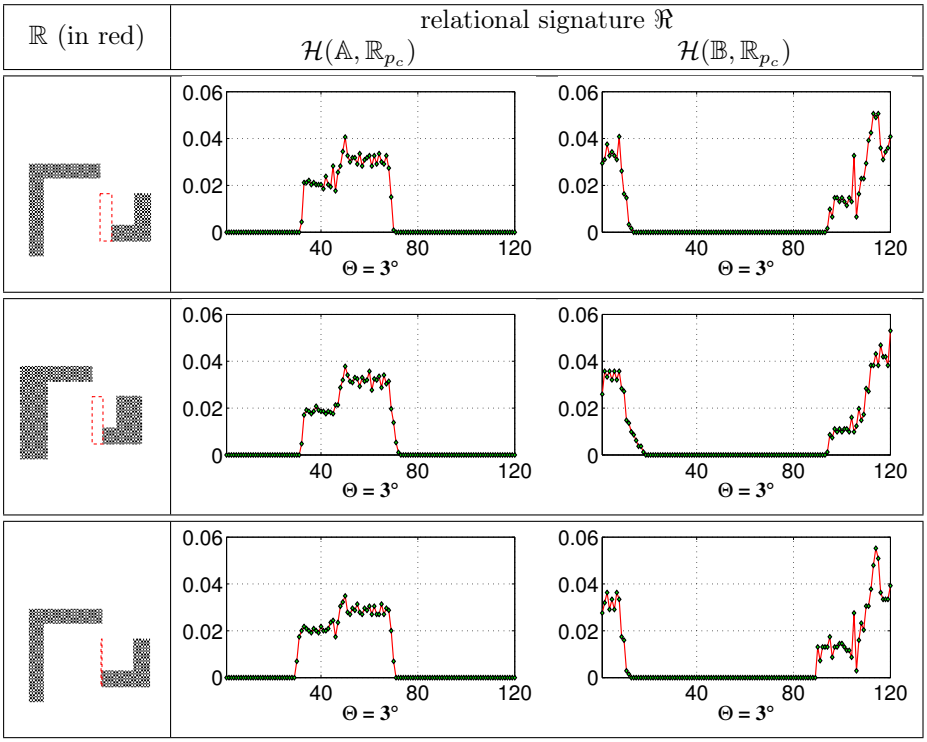


(a) boolean (b) metric

Figure 4.11: Behaviour of computational time for boolean and metric histograms, using a few sets of resolutions: 1°, 3°,5° and 9°. Image is scaled with the step of +2.

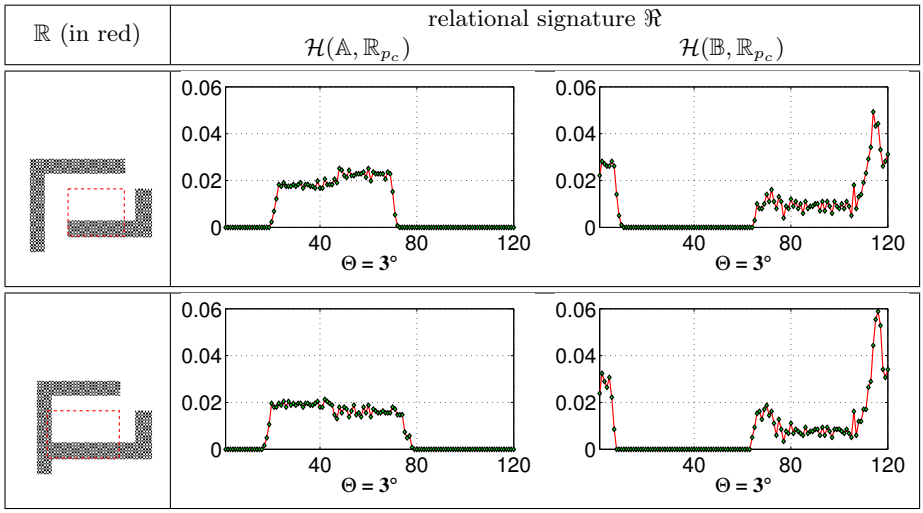
Several different topological configurations between two objects one many encounter are included. We start from a *disconnected* topology including stroke thickness variation as well as false *overlapping* and *inclusion*.

- For *disconnected* topological configuration as shown in Figure 4.12: keeping (a) as a reference image, we have changed a stroke thickness without changing relative positioning in (b) and moved objects closer while keeping an identical overall topological i.e., *disconnected* configuration in (c).



Left Object: $\mathbb{A}$ and Right Object: $\mathbb{B}$

Figure 4.12: Histograms at 3° resolution for *disconnected* objects including scaling.



Left Object: $\mathbb{A}$ and Right Object: $\mathbb{B}$

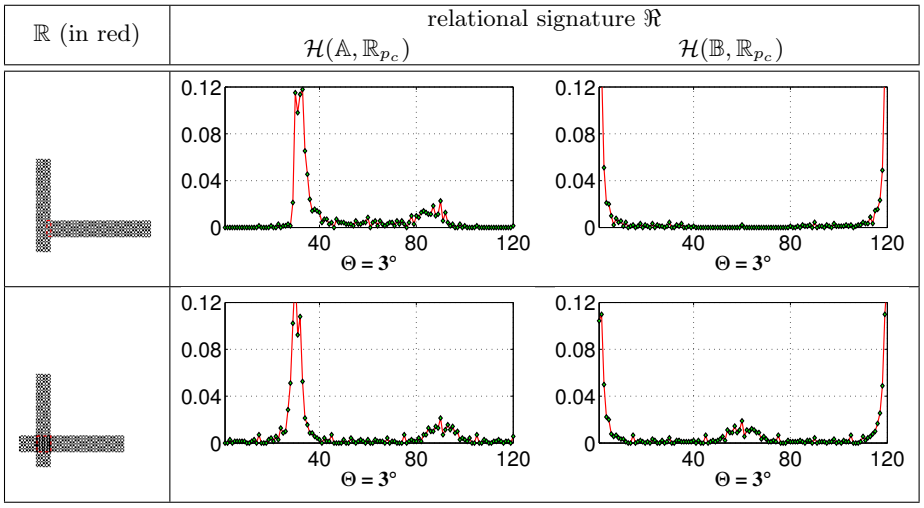
Figure 4.13: Histograms at 3° resolution for (a) false as well as (b) real *overlapping* objects.

For all configurations, the histograms do not show any significant difference. Scaling does not affect our method since $\mathcal{H}$ is normalised. In addition, the line rotation does not consider distances as long as they do not change the angular positioning.

- For false as well as real *overlapping* configurations, as shown in Figure 4.13:

  The coverage angle of $\mathcal{H}$ changes due to the change in structure (elongating horizontal limb in both objects). Our method, as shown in Figure 4.13, generates angular histograms according to the change in structure within the shape, even though angular position does not vary.

- *Overlapping* cases as shown in Figure 4.14:



Left Object: $\mathbb{A}$ and Right Object: $\mathbb{B}$

Figure 4.14: Histograms at 3° resolution for *overlapping* objects.

It is found that difference lies in the middle of $\mathcal{H}(\mathbb{B}, \mathbb{R}_{p_c})$ (between $40° - 80°$). The difference in histograms supports the fact that the method is able to discriminate slight changes in the object configurations even when identical topologies exist.

4.4 Symbol Description

Until now, we have developed spatial relation models and illustrated them with a few toy examples in order to provide an intuitive feeling on how they behave. In this section, we implement those spatial relations in a complete symbol recognition framework using real-world images. The section mainly focusses on how to describe the symbol that uses spatial relations between visual primitives that compose the symbol. Therefore, in what follows, we explain three major issues.

1. How to define and extract visual vocabulary?

2. How to compute pairwise spatial relations? and

3. How to integrate them in ARG framework?

As mentioned in Section 4.2.2, we first define our visual vocabulary (Section 4.4.1). In Section 4.4.2 we explain the way we compute pairwise spatial relations and finally use both in Section 4.4.3 to build an ARG and completely describe the symbol.

4.4.1 Visual Vocabulary

We define a set of well controlled visual elementary parts as a *vocabulary* [K.C. et al., 2009a]. While, in the general case, this vocabulary can be of any kind from any type of spatially significant features, related to what is visually pertinent in the application context under consideration, our current vocabulary is related to electrical symbols. It can be easily extended to adapt to other domains. Such visual elementary parts are extracted with the help of image treatment analysis operations as described in [Rendek et al., 2004]. They are briefly described below.

- *thick primitive:* We employ straightforward thin/thick separation by counting all *thick* connected components within the image. It uses standard skeletonisation based on chamfer distance and computes the histogram of line thickness. An optimal cut value is computed from the histogram to distinguish between thick zones and thin zones.
- *circle primitive:* We use the algorithm as described in [Lamiroy and Guebbas, 2010] which is based on Random Sample Consensus minimization.
- *corner primitive:* We mainly consider four types of corners such as North–East, North–West, South–East and South-West. It uses template matching process i.e., if the ratio of black and white pixels is greater than or equal to the template threshold, then the presence of corner is assessed.
- *extremity primitive:* We detect loose end coordinates from a given skeleton pixel where there is only one unique neighbouring pixel connecting to the main skeleton, itself connected by another unique pixel.

Figure 4.15 shows an illustration of those visual elementary parts, extracted from two different symbols. In what follows, rather than using every detected element as a basis for expressing and computing spatial relations, we group them together by type as shown in Figure 4.15. We denote the set of these generated groups as, $\sum_{\mathbb{T}} = \{\mathbb{T}_{thick}, \mathbb{T}_{circle}, \mathbb{T}_{corner}, \mathbb{T}_{extremity}\}$.

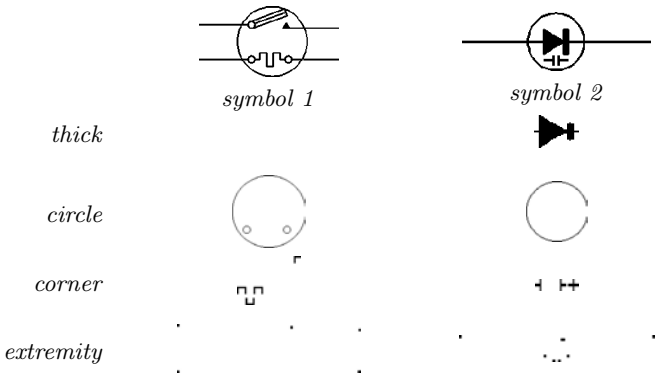


Figure 4.15: Illustration of vocabulary type.

4.4.2 Pairwise Spatial Relations

In order to express the spatial distribution of the previously formed groups, we compute a signature $\Re$ (explained in Section 4.3), expressing the spatial relations between two sets of pixels $\mathbb{A}$ and $\mathbb{B}$.

(a) *circle* and *corner*

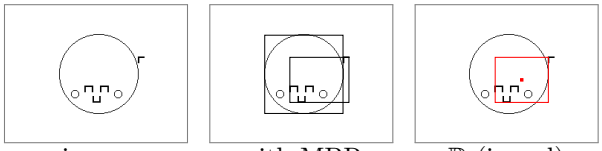
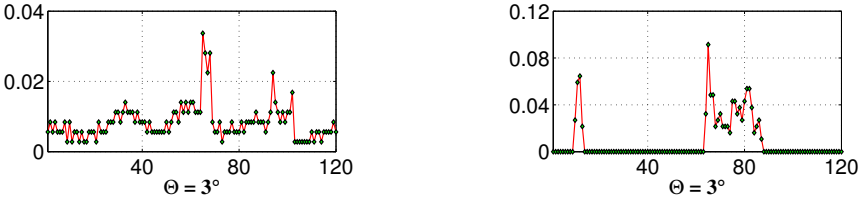


image with MBR $\mathbb{R}$ (in red)

1. Projection Model: $\mathcal{M}(circle, \mathbb{R}) = \begin{bmatrix} 0.094 & 0.112 & 0.098 & 0 \\ 0.088 & 0 & 0.044 & 0 \\ 0.165 & 0 & 0.165 & 0 \\ 0.055 & 0.123 & 0.056 & 0 \end{bmatrix}$ $\mathcal{M}(corner, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 0 & 0.143 \\ 0 & 0 & 0 & 0 \\ 0 & 0.857 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

2. Radial line Model:



$\mathcal{H}(circle, \mathbb{R}_{p_c})$ $\mathcal{H}(corner, \mathbb{R}_{p_c})$

(b) *circle* and *extremity*

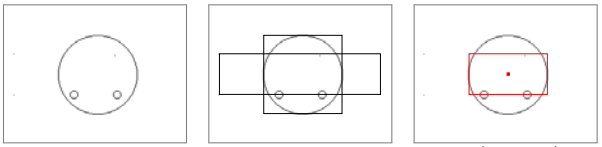
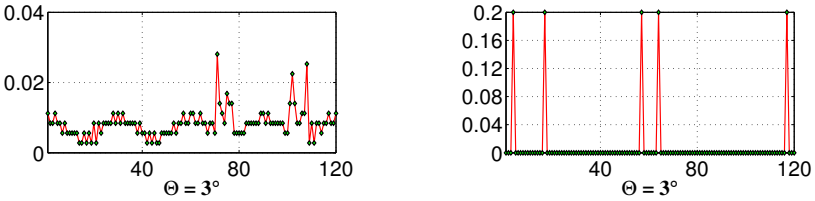


image with MBR $\mathbb{R}$ (in red)

1. Projection Model: $\mathcal{M}(circle, \mathbb{R}) = \begin{bmatrix} 0 & 0.140 & 0.140 & 0 \\ 0 & 0.061 & 0.061 & 0 \\ 0 & 0.092 & 0.092 & 0 \\ 0 & 0.181 & 0.181 & 0 \end{bmatrix}$ $\mathcal{M}(extremity, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.20 & 0 & 0.20 & 0.20 \\ 0.20 & 0 & 0 & 0.20 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

2. Radial line Model:



$\mathcal{H}(circle, \mathbb{R}_{p_c})$ $\mathcal{H}(extremity, \mathbb{R}_{p_c})$

(c) *corner* and *extremity*

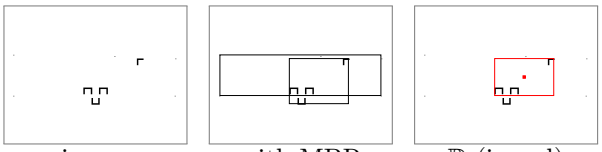
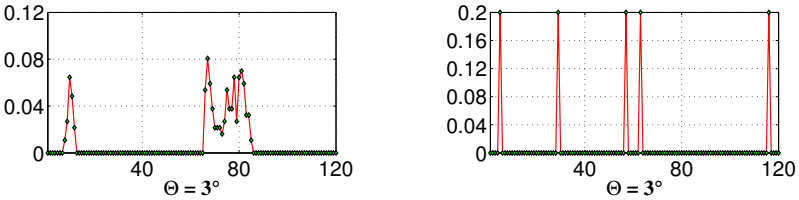


image with MBR $\mathbb{R}$ (in red)

1. Projection Model: $\mathcal{M}(corner, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0.14 & 0 \\ 0 & 0.58 & 0 & 0 \\ 0 & 0.28 & 0 & 0 \end{bmatrix}$ $\mathcal{M}(extremity, \mathbb{R}) = \begin{bmatrix} 0.20 & 0 & 0.20 & 0.20 \\ 0 & 0 & 0 & 0 \\ 0.20 & 0 & 0 & 0.20 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

2. Radial line Model:



$\mathcal{H}(corner, \mathbb{R}_{p_c})$ $\mathcal{H}(extremity, \mathbb{R}_{p_c})$

Figure 4.16: Relational matrices $\mathcal{M}$ and histograms $\mathcal{H}$ for all possible pairs of vocabulary types from *symbol 1* in Figure 4.15.

As mentioned in the previous section, each set contains all occurrences of a given vocabulary type. This implies that we know the visual vocabulary types to which $\mathbb{A}$ and $\mathbb{B}$ belong (*cf.* Section 4.4.1). Defining a fixed arbitrary order on the set of types $\sum_{\mathbb{T}}$ solves the potential ordering problem when comparing two corresponding relations.

For those vocabulary types from *symbol 1* in Figure 4.15, we have provided relational matrices and angular histograms, respectively from both projection and radial line models. The projection model is used to generate relation matrix and radial line model, angular histogram. Angular histogram at 3° resolution is chosen as an example. Those relations are used to label edges in ARG as shown in Figure 4.17.

4.4.3 Attributed Relational Graph

The vocabulary developed in Section 4.4.1 consists of a set of fixed label attributes, while the spatial relations between the attributes are the histograms described in Section 4.4.2. This gives us all the elements to express symbols as a complete ARG in which each vertex represents a distinct attribute type and the edges are labelled with a numerical expression of the spatial relations $\Re$.

More formally, we express the ARG as a 4-tuple $G = (V, E, F_A, F_E)$ where

V is the set of vertices;

$E \subseteq V \times V$ is the set of graph edges;

$F_A : V \rightarrow A_V$ is a function assigning labelled attributes to the vertices where A_V is the set of attributes type set $\sum_{\mathbb{T}}$ (*cf.* Section 4.4.1) and

$F_E : E \rightarrow \Re_E$ is a function assigning labels to the edges where $\Re$ represents the spatial relation of the edge E (*cf.* Section 4.4.2). Note that $\Re$ does not provide symmetry, $\Re(\mathbb{A}, \mathbb{B}) \neq \Re(\mathbb{B}, \mathbb{A})$. But, this can be solved by fixed ordering of V and $\mathbb{R}$ is not affected.

For instance, using *symbol 1* in Figure 4.15 as an example, and its corresponding spatial relations in Figure 4.16 we obtain the following ARG representation: $G = \{$

$$\begin{aligned} V &= \{\mathbb{T}_1, \mathbb{T}_2, \mathbb{T}_3\}, \\ E &= \{(\mathbb{T}_1, \mathbb{T}_2), (\mathbb{T}_1, \mathbb{T}_3), (\mathbb{T}_2, \mathbb{T}_3)\}, \\ F_A &= \{(\mathbb{T}_1, \mathbb{T}_{circle}), (\mathbb{T}_2, \mathbb{T}_{corner}), (\mathbb{T}_3, \mathbb{T}_{extremity})\} \\ F_E &= \{((\mathbb{T}_1, \mathbb{T}_2), \Re(\mathbb{T}_1, \mathbb{T}_2)), ((\mathbb{T}_1, \mathbb{T}_3), \Re(\mathbb{T}_1, \mathbb{T}_3)), \\ &\quad ((\mathbb{T}_2, \mathbb{T}_3), \Re(\mathbb{T}_2, \mathbb{T}_3))\}\} \end{aligned}$$

This forms a complete graph, and therefore has $\frac{t(t-1)}{2}$ edges for t attribute types.

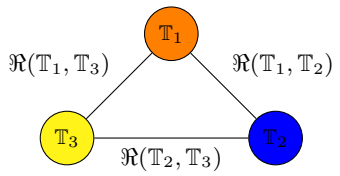


Figure 4.17: Attributed Relational Graph for a *symbol 1* in Fig. 4.15 – an example.

4.5 Symbol Recognition

Now that we have set up our ARG for symbol representation, we can define our recognition process. Recognition is based on maximal similarity, and is measured by a matching score. This score is based on matching the corresponding relational signatures between two given ARGs.

We then further extend the recognition to ranking database symbols based on the order of similarity, both of which will be explained in this section.

4.5.1 Matching

In this section, we first describe the global process to compute a matching cost between two graphs and then move to our problem: symbol recognition. In Figure 4.18, we provide a simple but complete process which consists of both vertex and edge substitution as well as insertion costs. In the example taken from [Aksoy, 2009] and represented in Figure 4.18. the vertex 'Tree' is missing in graph G^B. This example is provided to give general idea which can be regarded as one of the situations in our case. In what follows, we are going to fit the example to our situation.

In order just to fit the example into our problem, labelled edges can be considered as relations. The only difference is that we do not have any numeric label for vertices, rather known and labelled vocabulary types (this condition will change in Chapter 5).

Following the ARG description in Section 4.4.3, let us consider two graphs:

$$\begin{aligned} G^q &= (V^q, E^q, F_A^q, F_E^q) \text{ for the query symbol and} \\ G^d &= (V^d, E^d, F_A^d, F_E^d) \text{ for the database symbol.} \end{aligned}$$

Let us remind that the set of vertices $V = \{\mathbb{T}_1, \ldots, \mathbb{T}_t\}$, and the set of edges $E = \{E_1, \ldots, E_r\}$.

In order to explain our matching strategy, we are first taking the simplifying assumption that V^q and V^d are identical. In other words, both symbols contain items corresponding to identical vocabulary elements, but not necessarily sharing the same spatial arrangement. Since in our ARG every single vertex bears one distinct and unique attribute type, there is no cost in matching the vertices between G^q and G^d. As a consequence, matching edges is equally straightforward.

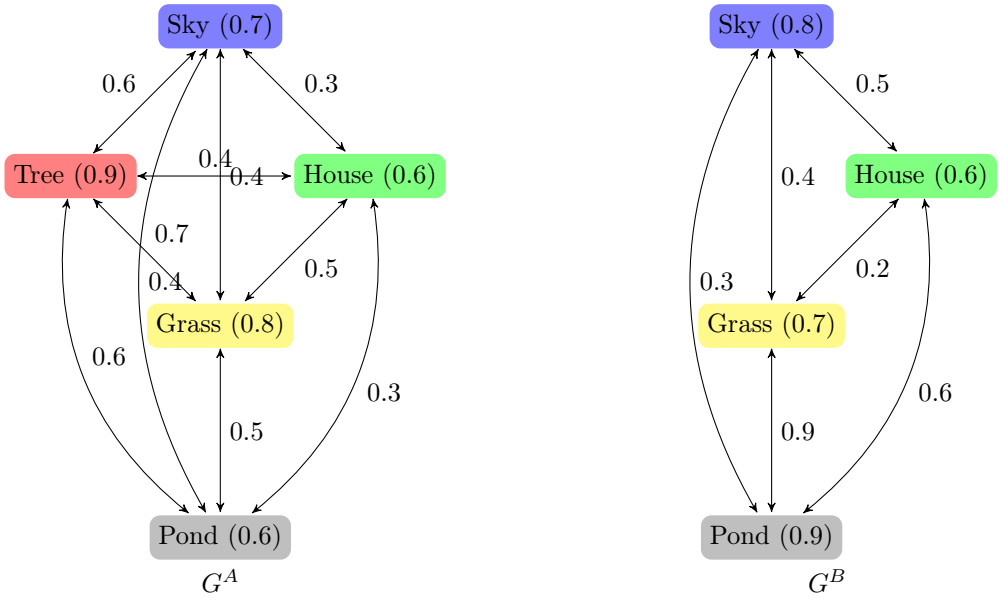
Since we have temporarily taken the assumption that V^q and V^d contain the same vocabulary elements, we can set up a bijective matching functions $\varphi : V^q \to V^d$ and $\sigma : E^q \to E^d$. This bijection exists such that uv is an edge in graph G^q if and only if $\varphi(u)\varphi(v)$ is an edge in graph G^d. Also we consider that ordering is preserved over the vertices sets V^q and V^d. *I.e.* $v_1 < v_2 \Rightarrow \varphi(v_1) < \varphi(v_2)$.

Thanks to our fixed labelling of attribute types, corresponding $\Re$ alignment is possible between the two given graphs and we can provide a matching score between the two given graphs G^q and G^d,

$$\text{Dist.}(G^q, G^d) = \sum_{r \in E} \delta\left(F_E^q(r), F_E^d(\sigma(r))\right)$$

where $\delta(a, b) = ||a - b||_2$. This is actually a very simple and straightforward metric. Given the performances of our method reported in Section 4.6 there is no real need to have a more complex one, unless rotational invariance is needed.

Of course, the assumption that V^q and V^d share the exact same vocabulary is too strong. To generalise the previously described approach to any situation, we define a binary (indicator)



Matching cost computation between G^A and G^B:

1.	Vertex substitution cost:	'Sky'	$\|0.7 - 0.8\| = 0.1$
		'House'	$\|0.6 - 0.6\| = 0.0$
		'Grass'	$\|0.8 - 0.7\| = 0.1$
		'Pond'	$\|0.6 - 0.9\| = 0.3$
2.	Edge substitution cost:	'Sky'↔'Pond'	$\|0.4 - 0.3\| = 0.1$
		'Sky'↔'Grass'	$\|0.4 - 0.4\| = 0.0$
		'Sky'↔'House'	$\|0.3 - 0.5\| = 0.2$
		'Grass'↔'House'	$\|0.5 - 0.2\| = 0.3$
		'Grass'↔'Pond'	$\|0.5 - 0.9\| = 0.4$
		'House'↔'Pond'	$\|0.3 - 0.6\| = 0.3$
3.	Vertex insertion cost:	'Tree' in G^A	0.9
4.	Edge insertion cost:	'Tree'↔'Pond' in G^A	0.6
		'Tree'↔'Sky' in G^A	0.6
		'Tree'↔'House' in G^A	0.4
		'Tree'↔'Grass' in G^A	0.7

Therefore, total matching cost $D(G^A, G^B)$ = substitution + insertion costs = 5.0.

Figure 4.18: Computing matching cost between two graphs G^A and G^B – an example [Aksoy, 2009].

function $\tau_A^V : \Sigma_{\mathbb{T}} \rightarrow \{0, 1\}$ to check the presence of vertices in the ARG, where the value of $\tau_A^V(\mathbb{T})$ is 1 if $\mathbb{T}$ is present in V and 0, otherwise. For instance, for the *symbol 1* shown in Figure 4.15, $\tau_A^V = [0, 1, 1, 1]$: which refers to the absence of *thick* components and the presence of *circle*, *corner* and *extremity* components. We can then use τ_A^{Vq} and τ_A^{Vd} for vertex insertion or deletion. Figure 4.19 shows an example. In this example, an empty vertex $\mathbb{T}_2^q$ in G^q is inserted and therefore corresponding adjacent empty relations, in order to provide one-to-one relation matching.

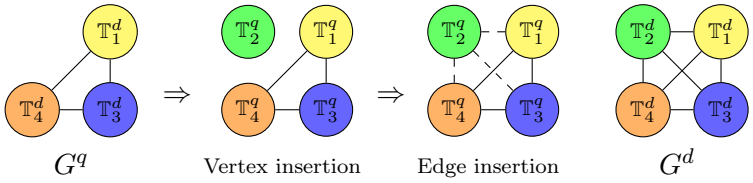


Figure 4.19: A graph transformation: $G^q \rightarrow G^d$.

4.5.2 Ranking

The previously defined matching score conveys how similar/dissimilar a database symbol is with respect to a query. In order for similarity to be ranging from 1 to 0, we normalise Dist.(.) to $[0, 1]$ by taking all database symbols: $\overline{\text{Dist.}}(.) = \frac{\text{Dist.}(.) - \text{Dist.}^{min.}(.)}{\text{Dist.}^{max.}(.) - \text{Dist.}^{min.}(.)}$. Now, the similarity is,

$$\text{Similarity}(G^q, G^d) \quad = \quad 1 - \overline{\text{Dist.}}(G^q, G^d).$$

Ranking can therefore be based on the decreasing order of similarity.

4.6 Experiments

In this section, we first give an overview of the symbols in our dataset. Then we explain the evaluation metric, clarifying its proper usage for this application. Based on the metric, we perform a series of experiments and confront our method with existing ones.

Our spatial relation is compared with other spatial relation models:

1. cone-shaped [Miyajima and Ralescu, 1994],
2. angle histogram [Wang and Keller, 1999] and
3. MBR [Papadias and Theodoridis, 1997].

Then we perform another assessment in order to establish a comparison of the complete recognition method with the state-of-the-art approaches. For this, we take a few representative global signal-based descriptors (*cf.* Chapter 3), applied directly to the symbol:

1. Zernike moments (ZM) [Kim and Kim, 2000],
2. generic fourier descriptors (GFD) [Zhang and Lu, 2002],
3. shape context (SC) [Belongie et al., 2002],
4. $\mathcal{R}-$signature [Tabbone et al., 2006] and
5. $\mathcal{D}-$Radon.

Finally, recent pixel-based approaches dedicated to symbol recognition are compared to our method. They are

1. statistical integration of histogram array (SIHA) [Yang, 2005] and
2. $2D$ kernel density [Zhang et al., 2006] based symbol representation.

To validate the method, a set of query images is compared with all images in the dataset. The database images are then ranked according to the similarity to the chosen query, using the similarity measure described in Section 4.5.2. To evaluate the methods, we first explain ground-truth formation in our dataset and then experimental protocol in the following section.

4.6.1 Dataset and Evaluation Metric

As said in Chapter 3, we work on a real world industrial problem to identify a set of different known symbols in aircraft electrical wiring diagrams: FRESH dataset [FRESH, 2007]. For evaluation, we use the same metric: *retrieval efficiency*.

4.6.2 Matching Scope

Since in our case, there are four types of vocabulary extracted from the symbol, the possible number of combinations of vocabulary types will be 2^4. However, a few combinations cannot be handled by our method because it requires at least two vocabulary types to develop relation between them.

Because of the fact we have a fixed set of labelled vertices (i.e., *vocabulary* types) in our symbol description, we are able to control the matching scope for every chosen query by using a parameter s. Using the notation introduced in Section 4.5.1, we define s as $\delta(\tau_A^{V_q}, \tau_A^{V_d})$. Depending on the value of s different matching strategies can be applied:

$s \geq 0$: all candidates in the dataset are taken into consideration for matching.

$s \leq 2$: matching is only done between candidates differing by at most two vertices (*i.e.*, two vertices can be absent or supplementary).

$s \leq 1$: matching is only done between candidates differing by at most one vertex (*i.e.*, one vertex can be absent or supplementary).

$s = 0$: matching is done by candidates only having the exact same set of vertices (*i.e.*, $V^q = V^d$).

We have applied the four different matching strategies to evaluate the behaviour of different methods with scopes ranging from $s \geq 0$ to $s = 0$. Our assumption is that candidates having the same set of vertices as well as exact labels are similar either for their whole structure or part of it when in composite forms. This assumption has been experimentally validated.

Example In Figure 4.20, the colour of the tokens represent the label of the vertex as reported in Section 4.4.3. For instance, the yellow coloured token represents the $\mathbb{T}_1$ vocabulary type in the ARG.

In this example shown in Figure 4.20, we provide a concept of matching scope by applying two different cases: $s = 0$: no difference in vertices allowed and $s \leq 1$: one difference in vertices is allowed. For $s = 0$, the example shows how matching scope is reduced to 5 symbols. This means that relation matching happens only with 5 symbols for a particular query.

4.6.3 Experimental Results

First, we test our method using all matching scope strategies. As mentioned in Section 4.4.2, projection model does not require any parameters, while for the radial line model, we need to fix or select optimal resolution Θ. We consider the influence of different resolutions Θ for relational signatures in radial line model. Once an optimal resolution is chosen, we present another series of experiments establishing the performances of our approach. We address two specific issues:

1. How does our spatial relation model compare to other spatial relation models?

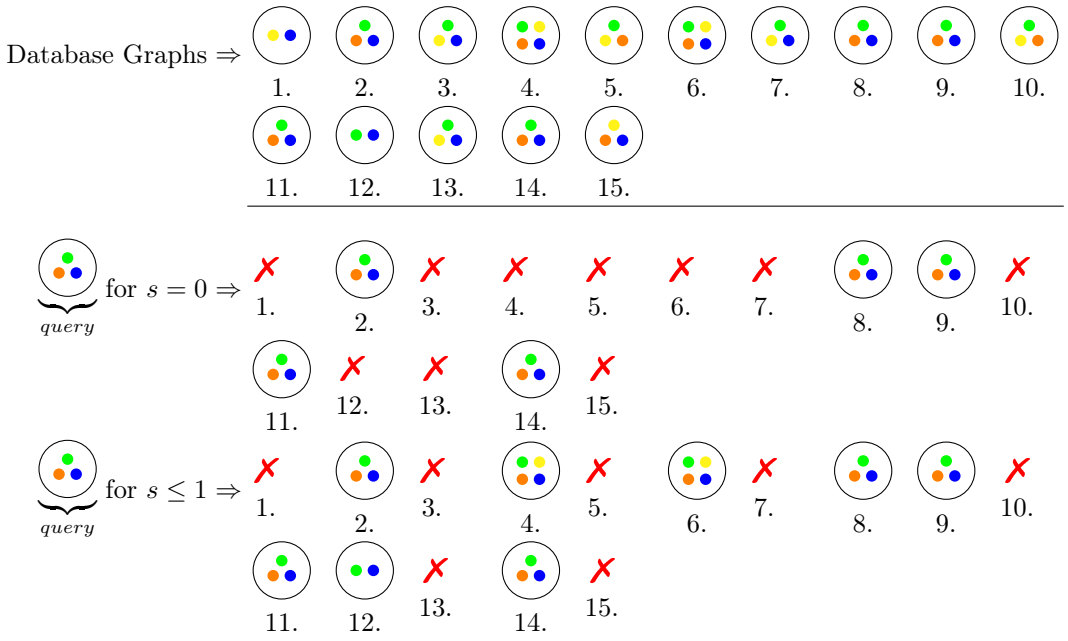


Figure 4.20: An example illustrating the matching scope.

2. How well does our recognition model do with respect to state-of-the-art recognition methods?

In all experiments, we have used *retrieval efficiency*, as described in Section 4.6.1. We compare the average retrieval efficiencies over the same 50 queries for all presented cases. These efficiency values have been computed for values of $K = 1$ to 10.

Our Method

Projection Model Figure 4.21 shows the retrieval efficiency of our projection model for all matching strategies. Matching $s = 0$ provides the best results.

Radial line Model This method, besides depending on the choice of the vocabulary, uses one main parameter: the resolution at which the angular histogram is computed[7]. Its value represents the trade-off between the optimal choice of resolution – and thus precision of spatio-structural information capture – and time/space requirements. Figure 4.22 shows the result of a series of experiments with Θ varying over $\{1°, 3°, 5°, 7°, 9°\}$. For each of its values we have measured the retrieval efficiency on the same set of queries. Without surprise, the lower Θ, the better the results, independently of the matching strategies used. Based on these results, and given the relatively low gain of efficiency between 3° and 1°, we adopt the former for the rest of our experiments.

In the following, it is important to notice that, before going to make a comparison with the state-of-the-art algorithms, we first test them and find the best of it to be compared later with our models.

[7]The matching scope s, as introduced in Section 4.6.2 should not really be considered as a parameter, but as a measure of our method's robustness.

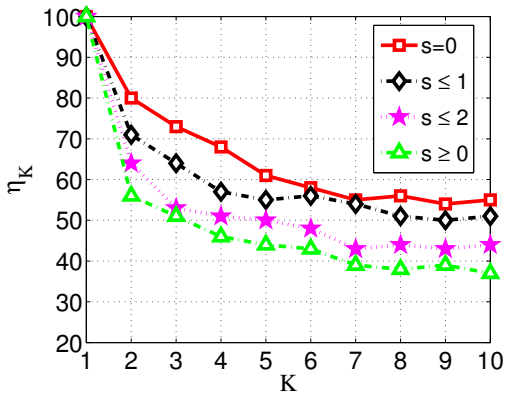


Figure 4.21: Average retrieval efficiency over requested list: 1 to 10 using projection model.

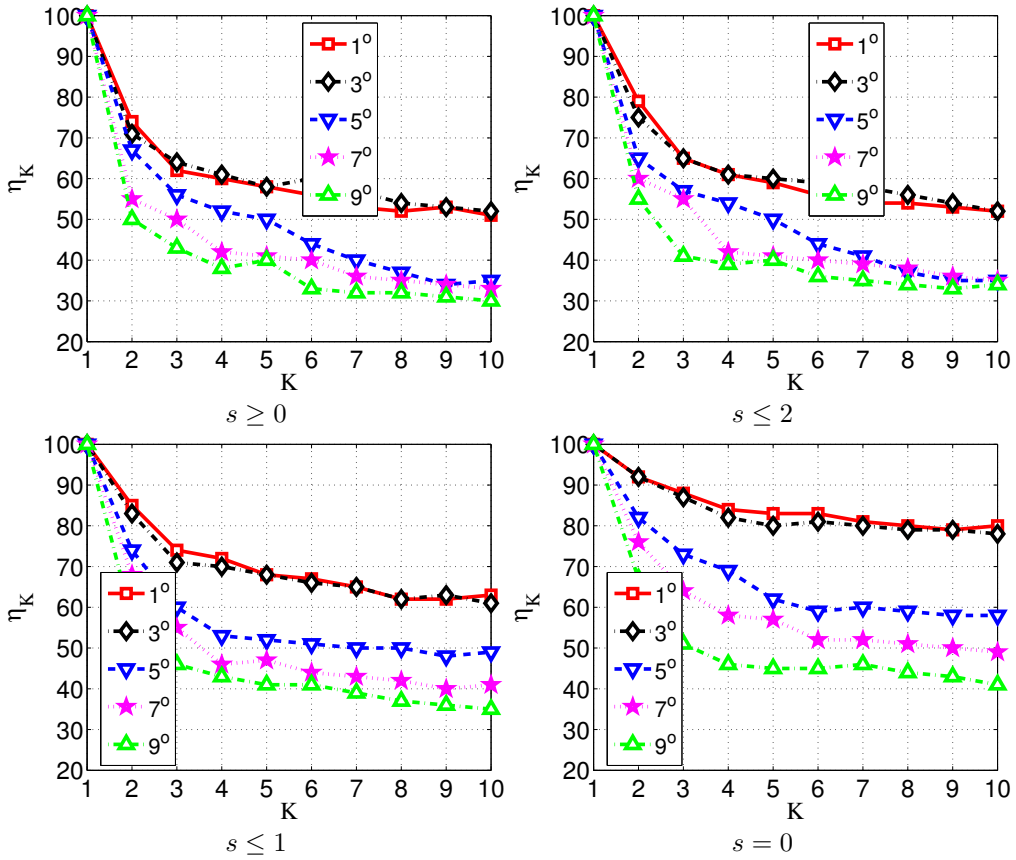


Figure 4.22: Average retrieval efficiency over requested list: 1 to 10 using radial line model. It uses different resolutions: 1°, 3°, 5°, 7° and 9°.

Other Spatial Relation Models

In order to compare our spatial relation model with others, we have adapted our ARG with cone-shaped [Miyajima and Ralescu, 1994], MBR [Papadias and Theodoridis, 1997] and angle histogram [Wang and Keller, 1999]. In order not to bias overall performance, we have submitted them to the same testing protocol as described before. Figure 4.23 shows their average retrieval

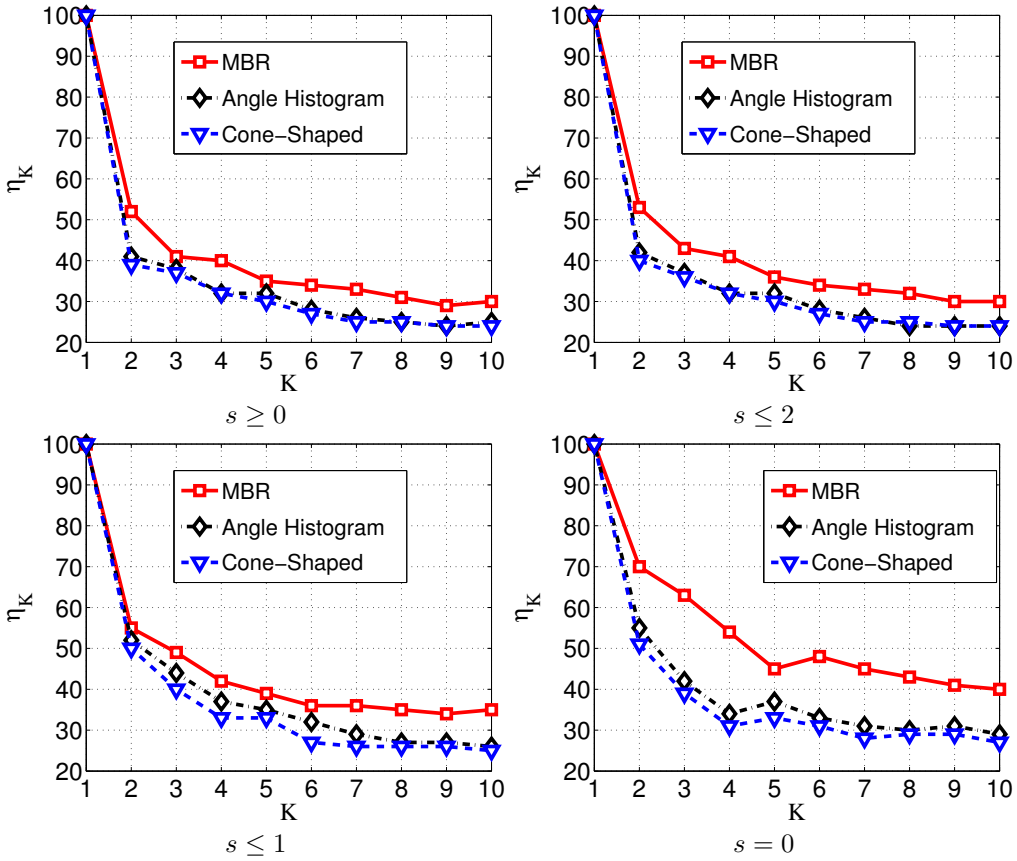


Figure 4.23: Average retrieval efficiency over requested list: 1 to 10 using fundamental spatial relation models: cone-shaped, angle histogram and MBR.

efficiency. MBR outperforms all others in all situations. We shall further compare it to our method at the end of this section.

Global Signal-based Descriptors

In order to compare our method to other recognition methods, we have selected a set of major global signal-based shape descriptors (*cf.* Chapter 3): Zernike moments [Kim and Kim, 2000], GFD [Zhang and Lu, 2002], shape context [Belongie et al., 2002], $\mathcal{R}-$signature [Tabbone et al., 2006] and $\mathcal{D}-$Radon. The same parameters will be used. Unlike the methods based on spatial relations, we cannot establish different matching scopes, based on 's' as presented in Section 4.6.2 and used previously. The same queries are presented and average retrieval efficiency is shown in Figure 4.24. $\mathcal{D}-$Radon and GFD (with marginal difference) seems to be performing the best among all tested global signal-based descriptors in our setup.

Pixel-based Approaches

We have also compared our method with two pixel-based approaches especially designed for symbol recognition: statistical integration of histogram array (SIHA) [Yang, 2005] and kernel density matching (KDM) [Zhang et al., 2006]. In SIHA, two different length-ratio and angle-ratio histograms are taken from every two pixels in reference to a third pixel from the skeleton

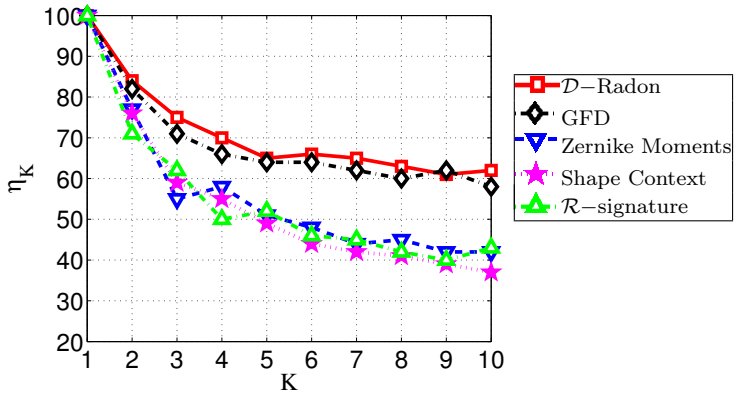


Figure 4.24: Average retrieval efficiency over requested list: 1 to 10 using global signal-based descriptors: Zernike moments, shape context, GFD, $\mathcal{R}-$signature and $\mathcal{D}-$Radon.

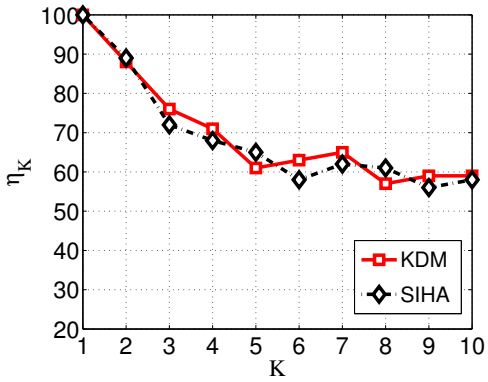


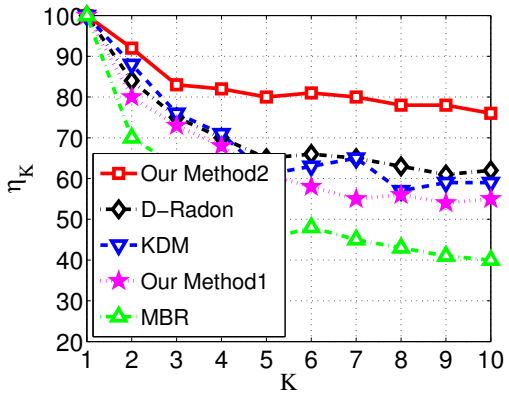
Figure 4.25: Average retrieval efficiency over requested list: 1 to 10 using pixel-based approaches: statistical integration of histogram array (SIHA) and kernel density matching (KDM)

image. In KDM, skeleton symbols are represented as $2D$ kernel densities and their similarity is measured by the *Kullback–Leibler divergence*. In Figure 4.25, results are shown for both. In this test, we observe almost similar behaviour from the two. However, KDM performs slightly better especially when also taking time complexity into account (see Table 4.1 in page 117).

Comparison

We can then compare our methods i.e., projection and radial line models with those presented before. Figure 4.26 provides a comparison by taking the best of each category: MBR from the basic spatial relation models, $\mathcal{D}-$Radon from the global signal-based descriptors and KDM from pixel-based approaches.

In Figure 4.26, projection model outperforms the conventional MBR model by approximately 10% retrieval efficiency. On the other hand, the model is lagging behind the global signal-based descriptors by approximately 8%. Our method using radial line model outperforms all spatial relation models as well as state-of-the-art methods. Our method outperforms both with a significant difference in retrieval efficiency. The difference with $\mathcal{D}-$Radon is more than 15% and 30% with MBR model. KDM and $\mathcal{D}-$Radon show similar behaviour.



Index: Our Method1 = projection model and Our Method2 = radial line model.

Figure 4.26: Comparison. Best of the breed – MBR from spatial relation model, $\mathcal{D}-$Radon from global signal-based descriptors and KDM from pixel-based approaches are taken for comparison with our proposed methods: projection model (method1) and radial line model (method2).

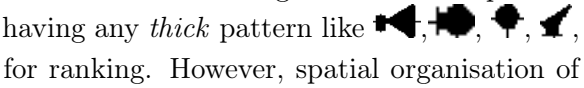
4.6.4 Experimental Results Analysis

In this section, the performance of the methods in response to the experimental results are analysed. Performance accounts for

1. retrieval efficiency and
2. execution time.

In parallel, we discuss matching scope and its effect in ranking retrieved symbols.

To visually compare the results of our method with the best of breed solutions reported in Figure 4.26, we show a selection of queries in Figure 4.27. They demonstrate the use of isolated as well as composed symbols as query. The first symbol on the top is always the chosen query and symbols are ranked from top to bottom (1 to 10) based on decreasing order of similarity. For query $Q1$, $\mathcal{D}-$Radon and KDM come close to our method while MBR presents a notable difference. In case of query $Q2$, our method outperforms all others significantly. A similar situation happens for $Q3$.

Our method exploits spatio-structural description of the visual primitives. The choice of the vocabulary types (i.e., collection of particular visual primitives) is of course an important factor to its success. However, symbols like  and  are retrieved for the query  due to the presence of *thick* patterns. This shows that our relational signatures do not provide or use any shape information. Therefore, symbols having any *thick* pattern like  and  are selected for ranking. However, spatial organisation of *thick* patterns with respect to other primitives helps to rank the best one first. Such variations of shape and size of the *thick* vocabulary types will be addressed in Chapter 5.

Running time has been measured in all experiments. An average running time (in sec.) for all methods is provided in Table 4.1.

Our method has benefited from the way we describe the matching strategy (*cf.* Section 4.6.2). Symbol matching between the candidates which share the same sets of vertices with exact labels (i.e., $s = 0$), is found to be the best among all. It sufficiently reduces time of matching to symbols which are obviously irrelevant. Similarly, this happens in those tests using basic spatial relations

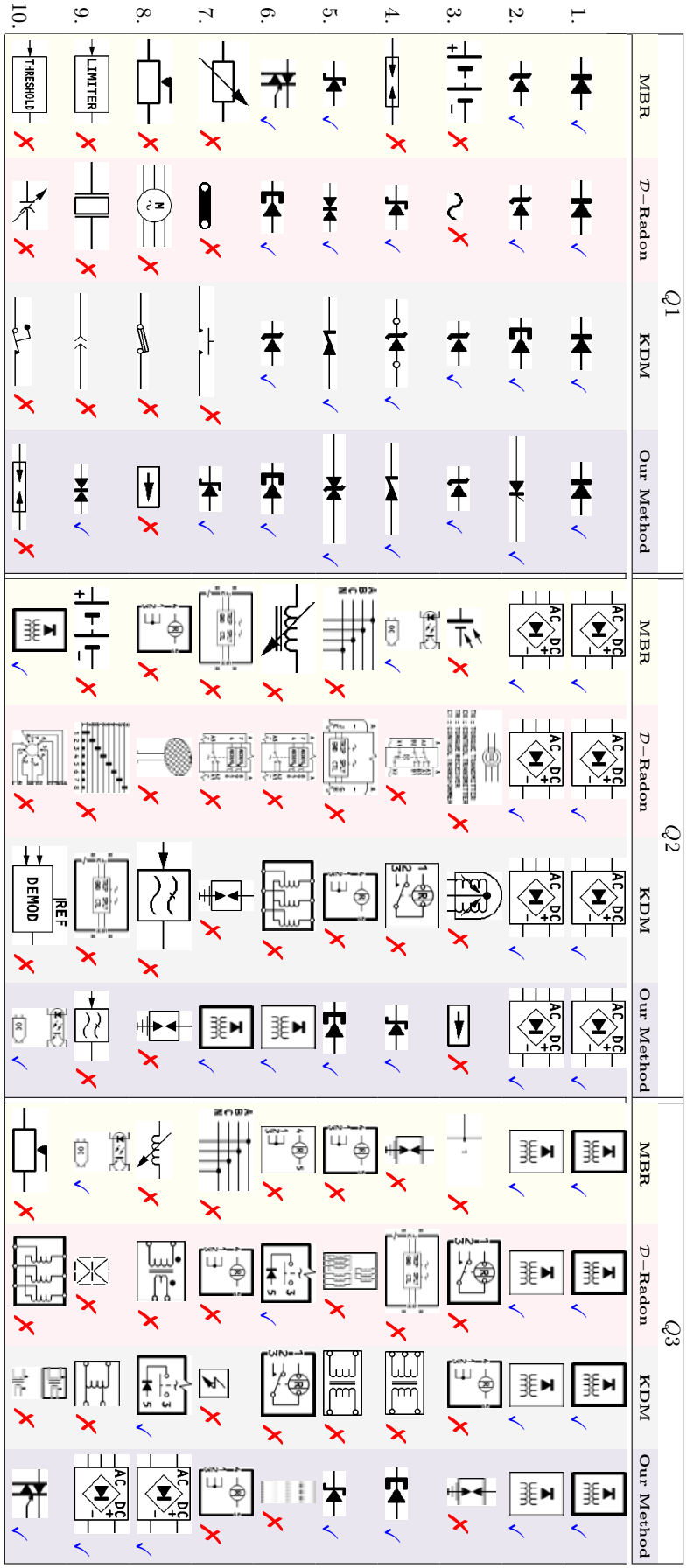


Figure 4.27: Visual illustration of symbol retrieval and ranking for a few queries, showing the true (✓) and false (✗) retrieval.

Methods	Time (in sec)
1. Basic Relation Models	
1.1. Cone-shaped [Miyajima and Ralescu, 1994]	$\leq$ 01
1.2. MBR [Papadias and Theodoridis, 1997]	02
1.3. Angle histogram [Wang and Keller, 1999]	44
2. Global Shape Descriptors (*cf.* Chapter 3)	
2.1. $\mathcal{R}-$signature [Tabbone et al., 2006]	02
2.2. ZM [Kim and Kim, 2000]	13
2.3. GFD [Zhang and Lu, 2002]	09
2.4. SC [Belongie et al., 2002]	32
2.5. $\mathcal{D}-$Radon	69
3. Pixel-based Approaches	
3.1. SIHA [Yang, 2005]	64
3.2. 2*D* KDM [Zhang et al., 2006]	24
4. Our Method	
4.1. Projection model	02
4.2. Radial line model	04

Table 4.1: Average running time (in sec).

models. But for global signal-based descriptors as well as pixel-based approaches, running time increases with number of symbols in the dataset since matching scope does not exist.

4.6.5 Advantages and Limitations

In this section, we discuss advantages and disadvantages of the proposed recognition method by highlighting three points:

1. Symbol Description – ARG based symbol description is interesting in the sense that any number of visual primitives can be represented in a fixed number of vertices. As a reminder, each vertex is labelled with distinct vocabulary type. The idea is simple and easy to implement as well as to extend. Note that spatial relations do not reveal complete shape information of the visual primitives.

2. Matching – Due to labelled vertices in our ARG, our matching avoids the NP-hard sub-graph isomorphism problem. Also, we reduce running time through the use of matching scope (*cf.* Section 4.6.2).

3. Recognition – Our method aims at retrieving linear symbols as well as in composite form. In case of composite symbols, it attempts to take significant parts of the image for recognition. Pre-filtering as described in Section 4.6.2 provides support in order to retrieve symbols in the composite form, due to labelled vocabulary types.

4. Affine Transformations

   - *translation.* It does not effect the spatial relations between the vocabulary since all visual primitives are also translated.
   - *scaling.* Since our method takes well controlled visual primitives from the symbol for computing spatial relations, image scaling does not really affect the method except a change in *thick* shape and size information.

- *rotation.* Spatial relations do not satisfy the rotation invariant property. We stick on it because of the characteristic of the dataset. We do not really have symbols with rotation. However, in Appendix B, we provide an extension of the work to show the behaviour of the proposed method in case of rotation, translation, scale (uniform as well as stretching), noisy as well as their combinations.

4.7 Conclusions

In this chapter, we have presented a method for describing symbols using a specific Attributed Relational Graph via the use of spatial relations between the visual elementary parts. Each vertex represents all visual primitives of a particular vocabulary type within the symbol. The edges represent the spatial relations between them. The proposed method is simple, and has the ability to express spatial relations between any number of visual primitives. We have validated that such a description can be used for symbol recognition. Our method has proven to significantly outperform state-of-the-art basic spatial relation models, global signal-based descriptors and pixel-based approaches for symbol recognition.

4.8 What is Next?

As mentioned in Section 4.6.4 and 4.6.5, relational histograms do not exploit shape information of the visual primitives. Therefore, in the next Chapter, we address the use shape description features for vocabulary primitives in addition to relations that exist between them.

End of Chapter

Keywords

Vocabulary,
Shape Signatures,
Clustering,
Spatial Relations,
Attributed Relational Graph,
Symbol Recognition.

5

Integrating Shape Signatures in ARG Representation for Symbol Recognition

Contents

5.1 Organisation of the Chapter 120
5.2 Introduction 120
5.2.1 Related Work 120
5.2.2 Outline of the Proposed Method 121
5.3 Vertex Signature through Shape Descriptions in ARG 122
5.3.1 Symbol Description 122
5.3.2 Symbol Recognition 123
5.4 Vocabulary Clustering through Shape Analysis in ARG 124
5.4.1 Symbol Description 124
Clustering of patterns 126
Cluster Verification and Validation 128
5.4.2 Symbol Recognition 134
5.5 Experiments 137
5.5.1 Experimental Results 138
Vertex Signature and its Integration with Edge Signature 138
Clustering of *thick* patterns 140
5.5.2 Experimental Results Analysis 140
5.5.3 Advantages and Limitations 142
5.6 Conclusions 142
5.7 What is Next? 144

Foreword. In the previous contribution, we have developed new spatial relation models and validated in the framework of symbol recognition. Spatial relations are used to provide visual links between the extracted visual primitives. These are then used as a basis for building an ARG that fully describe the symbol.

In this chapter, we present a method for symbol recognition based on the use of spatial organisation and shape features of the visual primitives that compose a symbol. Keeping the ARG based symbol description presented in Chapter 4, we address the use of shape signatures (*cf.* Chapter 3) in two different ways. First, shape signatures are used to label vertices. Second, we consider applying shape features only to the vocabulary which shows significant shape variation. Overall, both are integrated with relational signatures in the framework of ARG based symbol description. Consequently, the description is used to globally recognise structure by comparing the signatures. The method is experimentally validated in the context of electrical symbol recognition from wiring diagrams using the same dataset as in Chapter 4. ■

5.1 Organisation of the Chapter

The chapter is organised as follows. We start with an introduction in Section 5.2 which mainly includes our motivation, describes the underlying problem, reviews related work and gives an outline of our proposed method. We explain the proposed method in Section 5.3 and Section 5.4. In Section 5.3, we explain the way how we integrate vertex signature via shape analysis while in Section 5.4, we discuss pattern (extracted visual part) clustering through shape analysis. Both are integrated with relations, keeping the same ARG framework for symbol description as in Chapter 4. Full experiments are reported in Section 5.5. We point out the summary of the work in Section 5.6.

5.2 Introduction

Structural approaches provide powerful representations, conveying how parts are connected to each other. However, relations (e.g., relational histograms) do not exploit shape information as shape descriptors do. On the other hand, considering our application, global signal-based shape descriptors cannot provide optimal retrieval performance since the symbols are not only in linear and isolated form, they are composed with many elements instead. This has been clearly explained in Chapter 2 and attested in Chapter 3. Therefore, optimal selection of shape features and where to apply are the two primary tasks in the ARG framework discussed in Chapter 4.

Considering the problem of symbol localisation in real documents, composed of individual parts constrained by spatial relations, one needs to be able to formalise the relations that exist between the extracted visual primitives in addition to the description of individual shapes. This integration of shape description (*cf.* Chapter 3) of the extracted visual primitives and spatial relations (*cf.* Chapter 4) between them is going to be the core of this chapter. This means that we efficiently integrate statistical features with structural in order to fit to our symbol recognition problem.

5.2.1 Related Work

Along with the use of spatial relations for symbol recognition, we have discussed symbol representations, matching techniques as well as performances of several different structural approaches in Chapter 4. Once again, a few major references can be recalled as [Cordella and Vento, 2000a, Lladós et al., 2001]. In particular, we have kept our eyes on shortcomings considering our un-

derlying problem. In this section, we discuss shape analysis for symbol recognition and address the possible integration of both approaches: structural and statistical.

In [Cordella and Vento, 2000a], shape analysis for symbol recognition has been comprehensively addressed. It mainly refers to isolated shapes i.e., a simple $2D$ binary shapes. Under statistical approaches, global signal-based descriptors [Yuen et al., 1998, Kim and Kim, 2000, Tabbone et al., 2006, Belongie et al., 2002, Zhang and Lu, 2002, Zhang and Lu, 2004] are usually quite fault tolerant to image distortions, since they tend to filter out small detail changes. This is unfortunately an inconvenient in our context. Symbols may either be very similar in shape – and only differ by slight details – or either be completely different from a visual point of view. Symbols may also be composed of other known and significant symbols and need not necessary be connected. Moreover, they difficultly accommodate with connected or composite symbols. More precisely, the major drawbacks are due to deformation, composition with other symbols (which, in [Yuen et al., 1998] leads to unstable centroid detection, and thus errors in the ring projection) and occlusion over the boundary (leading to unstable tangents in Shape Context [Belongie et al., 2002]). Overall, they are generally not well adapted for capturing small detail changes, since they are specifically conceived to filter those out. Despite those shortcomings, researchers have been integrating descriptors [Salmon et al., 2007, Terrades et al., 2007b, Barrat and Tabbone, 2010] as well as combining several classifiers [Terrades et al., 2009] to increase their performance, partially based on the idea presented in [Tombre et al., 1998] that off-the-shelf methods are primarily designed for applications where line symbols are isolated. In these statistical approaches, signatures are simple with low computational cost. However, discrimination power and robustness strongly depend on the selection of optimal set of features for each specific application. In Chapter 3, we have studied a major set of shape descriptors and validated over several different datasets, in order to justify aforementioned points.

Therefore, an appropriate image description is required to integrate them where it accounts the generality and extensibility properties of structural approaches. It has also been clearly mentioned in [Tombre, 2010]:

> *"... the very structural and spatial nature of the information we work with makes structural methods quite natural in the community. Their efficient integration into methods which also take full advantage of statistical learning and classification is certainly the right path to take."*

An interesting example that uses shape descriptions and relations to form Region Adjacency Graph (RAG) is found in [Bodic et al., 2009]. The vector-based RAG is based on segmented regions which are labelled as vertices and geometric properties of an adjacency relations are used to label edges. The approach is limited once segmented regions change with image transformations and hence it uses few model classes to localise symbols in the technical documents. In contrast, our integration is different from the way we apply shape descriptors and the use of spatial relations. The proposed method will be discussed in the following section.

5.2.2 Outline of the Proposed Method

Global signal-based descriptors are only applied to isolated patterns (*cf.* Chapter 3). The spatial relations developed in Chapter 4 are able to exploit spatio-structural information of visual primitives. However, it does not really carry shape information in the same way shape descriptors do. Therefore, in this chapter, we aim to combine the best of both worlds: statistical and structural, and try to avoid the shortcomings of each of them. To do so, we continue to

decompose symbols by expressing their various parts in a fixed visual vocabulary, using spatial relations, graphs and signal-based descriptors to describe the whole shape.

Overall, we concentrate on two major contributions, keeping ARG framework developed in Chapter 4. They are

1. vertex signatures through shape descriptions in ARG (*cf.* Section 5.3); and
2. vocabulary clustering through shape analysis (*cf.* Section 5.4).

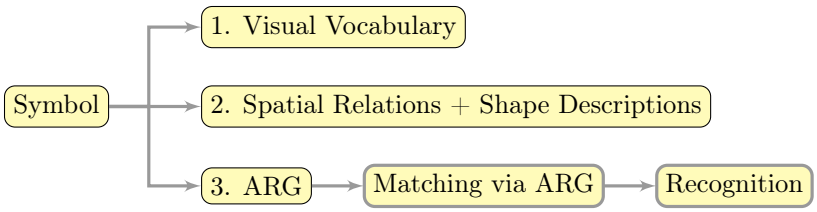


Figure 5.1: An architecture for symbol description and recognition using shape features and spatial relations. It uses ARG based symbol description using visual vocabulary and their possible pairwise relations. Shape signatures are used to discriminate visual parts.

In both cases, we study the pertinence of the use of following major set of shape descriptors, into our method:

1. Zernike moments (ZM) [Kim and Kim, 2000],
2. generic fourier descriptors (GFD) [Zhang and Lu, 2002],
3. shape context (SC) [Belongie et al., 2002],
4. $\mathcal{R}-$signature [Tabbone et al., 2006] and
5. $\mathcal{D}-$Radon.

In the following, we highlight both symbol description and symbol recognition, for both cases.

5.3 Vertex Signature through Shape Descriptions in ARG

In our ARG based symbol description mentioned in Section 4.4.3 of Chapter 4, each vertex has a distinct vocabulary type containing different shape and size information. Since spatial relations only encode relative positioning and point distributions, and do not completely exploit global shape information, we employ a number of well-known descriptors to label vertices (i.e., visual primitives), aiming to increase overall recognition performance. The descriptors are listed in Section 5.2.2. In what follows, we explain how we describe the symbol based on our ARG framework and recognition process.

5.3.1 Symbol Description

In our ARG based symbol description, edges are labelled with a numerical expression of the spatial relations (*cf.* section 4.4.2) between the connected vertices that are labelled both with their distinct vocabulary types and their global shape descriptors.

In our formal ARG expression, it consists of a 4-tuple $G = (V, E, F_A, F_E)$: V is the set of vertices, $E \subseteq V \times V$ is the set of graph edges, $F_A : V \rightarrow A_V$ is a function assigning attributes to the vertices and A_V representing a set of vocabulary type set $\sum_{\mathbb{T}}$ as well as their global shape signatures $\mathcal{S}$, and $F_E : E \rightarrow \Re_E$ is a function assigning labels to the edges where $\Re$ represents spatial relations of the edge E. Figure 5.2 shows the visual description of it.

For any symbol which has attribute type set $\{\mathbb{T}_1, \mathbb{T}_2, \mathbb{T}_3\}$, its ARG representation can be expressed as $G = \{$

$$
\begin{aligned}
V &= \{\mathbb{T}_1, \mathbb{T}_2, \mathbb{T}_3\}, \\
E &= \{(\mathbb{T}_1, \mathbb{T}_2), (\mathbb{T}_1, \mathbb{T}_3), (\mathbb{T}_2, \mathbb{T}_3)\}, \\
F_A &= \{((\mathbb{T}_1, \mathbb{T}_{circle}), \mathcal{S}(\mathbb{T}_1)), ((\mathbb{T}_2, \mathbb{T}_{corner}), \mathcal{S}(\mathbb{T}_2)), ((\mathbb{T}_3, \mathbb{T}_{extremity}), \mathcal{S}(\mathbb{T}_3))\}^{\star} \\
F_E &= \{((\mathbb{T}_1, \mathbb{T}_2), \Re(\mathbb{T}_1, \mathbb{T}_2)), ((\mathbb{T}_1, \mathbb{T}_3), \Re(\mathbb{T}_1, \mathbb{T}_3)), ((\mathbb{T}_2, \mathbb{T}_3), \Re(\mathbb{T}_2, \mathbb{T}_3))\}\}.
\end{aligned}
$$

In contrast to ARG based symbol description presented in Section 4.4, it integrates shape description of vocabulary in vertices otherwise, the complete framework is identical.

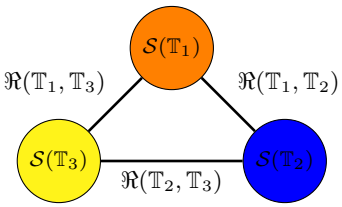


Figure 5.2: A complete ARG description for a symbol. Both vertex and relation signatures are used. The illustration can be referred to the *symbol 1* presented in Figure 4.15 Chapter 4 where attribute type set is $\{\mathbb{T}_{circle}, \mathbb{T}_{corner}, \mathbb{T}_{extremity}\}$. In contrast to Figure 4.17, the difference lies in the additional use of vertex signature*.

5.3.2 Symbol Recognition

In this section, we recall the matching concept explained in Section 4.5.1. The only modification is that the additional use of shape signatures on vertices.

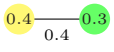
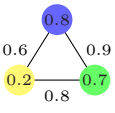
Based on our symbol description, matching of two given symbols is done by matching their corresponding ARGs. Let us consider two ARGs, query $G^q = (V^q, E^q, F_A^q, F_E^q)$ and database $G^d = (V^d, E^d, F_A^d, F_E^d)$, where the set of vertices $V = \{\mathbb{T}_1, \ldots, \mathbb{T}_t\}$, and the set of edges $E = \{E_1, \ldots, E_r\}$. Therefore, the global matching score between two graphs is,

$$
\begin{aligned}
\text{Dist.}(G^q, G^d) &= \alpha \text{ dist.}_{vertex}(G^q, G^d) + (1 - \alpha) \text{ dist.}_{relation}(G^q, G^d) \\
&= \alpha \sum_{t \in V} \delta(F_A^q(t), F_A^d(\sigma(t))) + (1 - \alpha) \sum_{r \in E} \delta(F_E^q(r), F_E^d(\varphi(r))),
\end{aligned}
$$

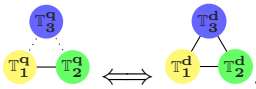
where $\alpha \in [0, 1]$ and $\delta(a, b) = \sum_{l=1}^{L} ||a_l - b_l||_2$. The parameter α provides weight while matching.

- $\alpha = 0$: only vertex signature;
- $\alpha = 1$: only edge signature; and
- $\alpha = 0.5$: equal weights to both vertex and edge signature.

In our experiments, we provide equal weights for both relations and shape distribution of the whenever they are integrated together.

Illustration Consider G^q  and G^d  . Vertices are labelled as . Then matching score between them is,

$$\text{Dist.}(G^q, G^d) = \alpha \underbrace{\begin{bmatrix} \delta(\mathcal{S}_{\mathbb{T}_1^q}, \mathcal{S}_{\mathbb{T}_1^d}) + \delta(\mathcal{S}_{\mathbb{T}_2^q}, \mathcal{S}_{\mathbb{T}_2^d}) \\ + \\ \delta(\mathcal{S}_{\mathbb{T}_3^q}, \mathcal{S}_{\mathbb{T}_3^d}) \end{bmatrix}}_{\text{dist.}_{vertex}} + (1-\alpha) \underbrace{\begin{bmatrix} \delta(\Re_{\mathbb{T}_1^q, \mathbb{T}_2^q}, \Re_{\mathbb{T}_1^d, \mathbb{T}_2^d}) \\ + \\ \delta(\Re_{\mathbb{T}_1^q, \mathbb{T}_3^q}, \Re_{\mathbb{T}_1^d, \mathbb{T}_3^d}) + \delta(\Re_{\mathbb{T}_2^q, \mathbb{T}_3^q}, \Re_{\mathbb{T}_2^d, \mathbb{T}_3^d}) \end{bmatrix}}_{\text{dist.}_{relation}}$$

where $\Re_{x,y} = \Re(x, y)$. It is clear that $\mathbb{T}_3^q$ has to be inserted when G^q is transformed to G^d. As a consequence virtual connections: $\Re_{\mathbb{T}_1^q, \mathbb{T}_3^q}$ and $\Re_{\mathbb{T}_2^q, \mathbb{T}_3^q}$ exits:  . The labelled vertices in our ARG based symbol description makes easier to handle this. In our numeric example, this behaves as

$$\text{Dist.}(G^q, G^d) = \alpha \underbrace{\begin{bmatrix} ||0.4-0.2||_2 + ||0.3-0.7||_2 \\ + \\ ||0-0.8||_2 \end{bmatrix}}_{\text{dist.}_{vertex}} + (1-\alpha) \underbrace{\left[\, ||0.4-0.8||_2 \,\right] + \beta_2 \begin{bmatrix} ||0-0.6||_2 \\ + \\ ||0-0.9||_2 \end{bmatrix}}_{\text{dist.}_{relation}}.$$

Now the turn is to give weights to dist._{vertex} and $\text{dist.}_{relation}$.

5.4 Vocabulary Clustering through Shape Analysis in ARG

One of the major problems caused by *thick* visual primitives in our vocabulary set (see Section 4.4.1 in Chapter 4) affects retrieval performance. It is primarily because of shape and size variation. Moreover, number of *thick* patterns is different from one symbol to another. Therefore, in the framework of ARG based symbol description developed in Chapter 4, we propose to integrate group of similar *thick* patterns through clustering via shape features.

In what follows, we will first discuss symbol description (*cf.* Section 5.4.1) and then matching strategy (*cf.* Section 5.4.2) for recognition process. For both symbol description and recognition, we mainly focus on *thick* pattern clustering and the way how it is integrated with ARG.

5.4.1 Symbol Description

After integrating shape features for a particular vocabulary that shows significant shape and size variation, our symbol description goes like this.

The ARG as established in Chapter 4 is now composed of splitting *thick* vertex into more specialised *thick* sub-vertices. Therefore, the total number of *thick* sub-vertices is equivalent to the number of *thick* patterns within the symbol. Figure 5.3 shows an overall concept of the proposed symbol description. In this illustration, *thick* sub-vertices are connected with the remaining vertices in addition to the use of shape signatures in it. This is primarily the change of paradigm with respect to previous ARG model.

To elaborate it and in contrast to Section 4.4 in Chapter 4, and following Figure 5.3, for any symbol which has attribute type set $\{\mathbb{T}_1, \mathbb{T}_2, \mathbb{T}_3\}$, its ARG representation can be expressed as $G = (V, E, F_A, F_E)$ and $F_A = \{((\mathbb{T}_1, \mathbb{T}_{thick}), \mathcal{S}(\mathbb{T}_1)), (\mathbb{T}_2, \mathbb{T}_{circle}), (\mathbb{T}_3, \mathbb{T}_{corner})\}$ where we assign label through shape signature in vertex $\mathcal{S}(\mathbb{T}_1)$. It goes separately for all individual *thick* patterns like $\{\mathcal{S}(\mathbb{T}_{1,\kappa}), \ldots, \mathcal{S}(\mathbb{T}_{1,\mathcal{K}})\}$, where $\mathcal{K}$ is the number of *thick* patterns in a symbol. Overall, for any

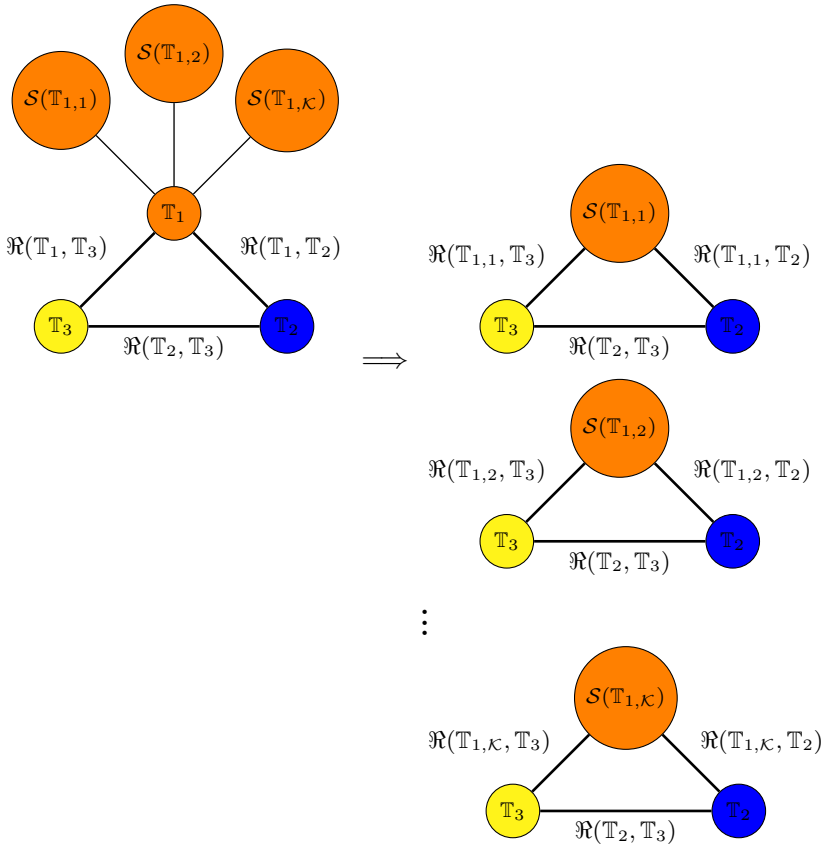


Figure 5.3: A complete ARG description for a symbol which has vocabulary type set $\sum_{\mathbb{T}} = \{\mathbb{T}_1, \mathbb{T}_2, \mathbb{T}_3\}$ labelled respectively with *thick*, *corner* and *extremity*. In this illustration, the original graph is decomposed into a number of graphs which is equivalent to the number of extracted *thick* patterns.

symbol, there are $\{G_\kappa\}_{\kappa=1,\ldots,\mathcal{K}}$ ARGs. Once again, it is important to remind that, in contrast to Figure 4.17 (in page 106 of Chapter 4), the difference lies in the separate use of shape signatures for all *thick* patterns.

According to our symbol description, since there exists arbitrary number of *thick* patterns from one symbol to another, number of ARGs to describe a complete symbol can be varied accordingly. Consequently, it takes more matching time compared to the matching mentioned previously in Chapter 4. To avoid such an intense time complexity issue, we propose to use *thick* pattern clustering so that matching can be handled efficiently without using one-to-all graph matching. The overall process goes like this. All *thick* patterns are grouped first and used them for identifying related symbols in the dataset. For instance, the query *thick* pattern belongs to the particular cluster will be taken for ranking. After that the ranking goes through the use of relational signature (*cf.* Chapter 4) matching between those symbols related to particular cluster. The more detail idea on matching is provided in Section 5.4.2.

In what follows, we first explain the general clustering algorithm, verification and validation concepts for unsupervised clustering together with illustrations and then followed by real-world example.

Clustering of patterns

Let us illustrate the problem with an example. This is shown in Figure 5.4. The shape and size of the *thick* pattern is related to kind of symbol from which it detected. For instance, a *thick* pattern coming from junction is different from a triangle-shaped one, such as a significant part of a diode symbol or from an arrow , . To separate them, the main idea is to cluster similar ones in the same group. The collected *thick* patterns in a group are assumed to extracted from the similar types of symbols.

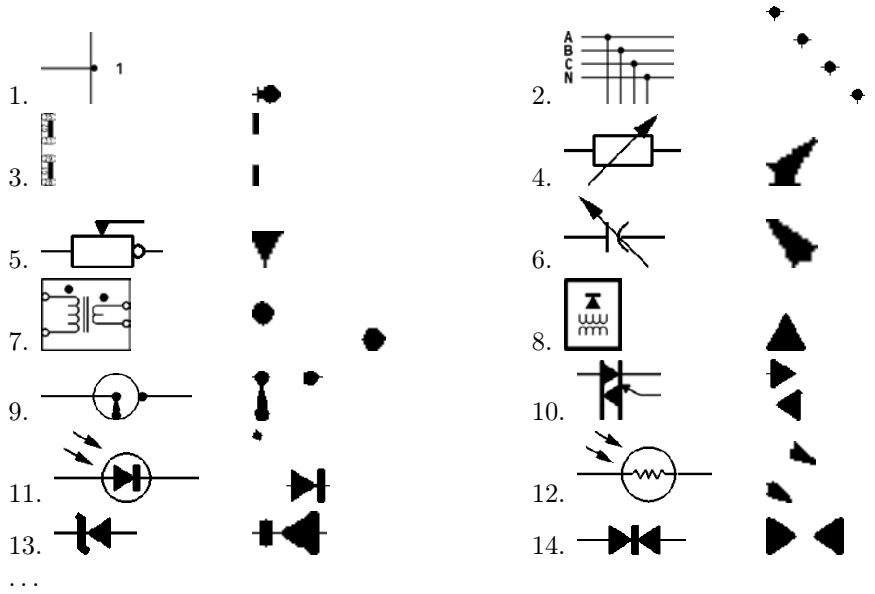


Figure 5.4: Symbols and their corresponding *thick* patterns. For visual understanding, *thick* patterns are zoomed out.

A comprehensive idea of clustering techniques has been presented in [Jain and Dubes, 1988, Reynolds et al., 2006]. Based on it, we have presented *thick* pattern clustering that is composed of two major steps.

step 1 Find the similarity or dissimilarity between every possible pair of objects.

All *thick* patterns are represented by shape signatures. For every particular description, three different metrics are used in order to get the distance (similarity or dissimilarity) matrix. They are

1. city-block: $\delta(x, y) = \|x - y\|_1 = |x - y|$,
2. euclidean: $\delta(x, y) = \|x - y\|_2 = \sqrt{(x - y)^2}$ and
3. squared euclidean: $\delta(x, y) = \|x - y\|_2^2 = (x - y)^2$,

where $\delta(x, y)$ is the distance between elements $x \in X$ and $y \in Y$.

step 2 Group the similar ones in the form of hierarchical cluster tree.

In order to group them, we implement three different types of linkage methods for clustering based on the distance matrix.

1. *Single-linkage* Clustering; is sometimes known by name *nearest neighbour* clustering. In this method, the distance between two clusters is computed as the distance between the two closest elements in two clusters.
   Mathematically, the linkage function – the distance between two clusters $D(X, Y)$ can be expressed as,

$$\text{Dist.}(X, Y) = \min \{ \delta(x, y) : x \in X,\, y \in Y \}.$$

2. *Complete-linkage* Clustering uses the maximum distance between the two clusters. Mathematically, it can be expressed as,

$$\text{Dist.}(X, Y) = \max \{ \delta(x, y) : x \in Y,\, y \in Y \}.$$

3. *Average-linkage* Clustering: The method uses the mean distance between elements of each cluster. The distance between any two clusters X and Y is taken to be the average of all distances between pairs of objects $x \in X$ and $y \in Y$:

$$\text{Dist.}(X, Y) = \frac{1}{|X| \times |Y|} \sum_{x \in X} \sum_{y \in Y} \delta(x, y).$$

Until now, we have defined the possible linkage methods which are based on distance. Based on that, the clustering process can be described as follows.

Consider $\mathcal{M} = N \times N$ proximity matrix from N *thick* patterns. An agglomerative hierarchical clustering scheme consists in erasing rows and columns in this proximity matrix each time clusters are grouped together. The clusters are then assigned as sequence numbers: $0, 1, \ldots, (n-1)$ and $\mathcal{L}(k)$ is the level of the k^{th} clustering.

1. Consider disjoint cluster having level $\mathcal{L}(0) = 0$ and sequence number $c = 0$.
2. Find the most similar pair of clusters in the current clustering, say pair (r), (s). Similarity is depend on the chosen metric and linkage method.
3. Increment the sequence number: $c = c + 1$. Group clusters (r) and (s) into a single cluster to form the next clustering c. Set the level of this clustering to $\mathcal{L}(c) = \text{Dist.}[(r), (s)]$.
   Update the proximity matrix, $\mathcal{M}$, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r, s) and old cluster (k) is defined as $\text{Dist.}[(k), (r, s)]$.
4. Repeat until all clusters are merged or it reaches cluster-threshold.

Figure 5.5 shows an example of a dendrogram using agglomerative hierarchical clustering. In this illustration, the clustering process ends up with a single cluster. The similarity between the pair is simply taken from the distance computation. For instance, patterns c_1 and c_2 are merged at the level of 1.5 distance. This is also called dendrogrammatic distance.

Although the clustering has been simplified in two steps, there remain some unanswered questions. Those will be the main part of our contribution.

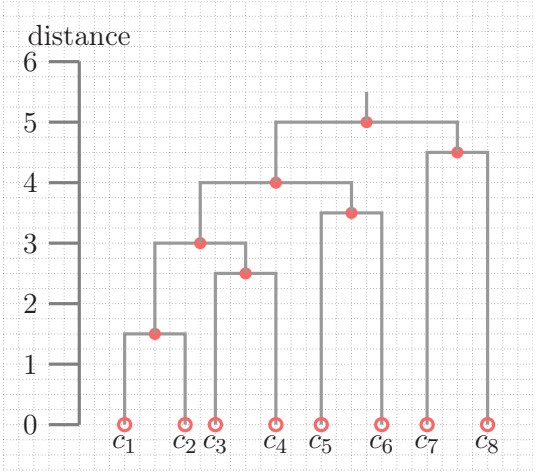


Figure 5.5: Dendrogram example using eight patterns: $c_1, c_2, \ldots, c_8$. Every pair is merged based on the similarity between them.

1. Testing all above mentioned metrics one-by-one will be time consuming as well as difficult to evaluate and or verify the clusters. How can we optimise it?

   Of course, the choice of metric affects the shape of the clusters. Therefore, we verify the clusters by using all possible combinations of metrics and linkage methods in one shot.

2. How and where to cut the cluster tree into clusters?

   It refers to the number of clusters. The illustration in Figure 5.5 only provides the clustering process until it merges all, but not the cut-off threshold. To efficiently handle appropriate number of clusters, one either has to chose the threshold manually or set the best threshold by validating the clusters. The latter one i.e., cluster validation mainly requires either unsupervised or supervised.

A full explanation of what is related to *item 1* i.e., cluster verification and what is related to *item 2* i.e., cluster validation will be provided in the following sections.

Cluster Verification and Validation

Cluster Verification As mentioned earlier, cluster analysis is highly sensitive to

1. distance metric selection and

2. criterion for determining the order of clustering.

Different approaches that use different combinations of distance metric and linkage method may yield different results. Therefore, the distance metric and clustering criterion should be chosen carefully. However, the best combination changes in the function of features and dataset characteristics. As a consequence, we need to compare all possible combinations one-by-one. To efficiently achieve this, we use 'cophenetic distance' for every combination and choose the best one before cluster validation.

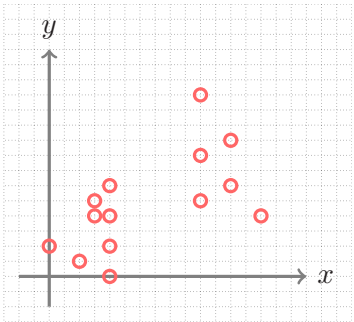


Figure 5.6: Sample points.

Cophenetic Distance and its Correlation In hierarchical clustering, the height of the link represents the distance between two clusters (see Figure 5.5). This height is known as cophenetic distance [Sokal and Rohlf, 1962, Rohlf and Fisher, 1968, Carr et al., 1999]. If the clustering is valid, the linking of objects in a cluster tree should have a strong correlation with the distance between the objects (in the distance matrix). Cophenetic correlation compares these two sets of values. Therefore, the cophenetic correlation coefficient closer to 1 is the best one – that accurately clusters the data. Overall, the cophenetic correlation coefficient is a measure of how faithfully a dendrogram preserves the pairwise distances between the original unmodelled data points.

Consider the original data X_i have been clustered to produce a dendrogram T_i. Besides, let us define the following distance measures.

- $x(i,j) = \delta(X_i, X_j)$: distance between the i^{th} and j^{th} samples.
- $t(i,j)$: the dendrogrammatic distance between the model points T_i and T_j. This distance is the height of the node at which these two points are first joined together. Figure 5.5 gives a clear visual idea.

Let x be the average of the $x(i,j)$ and t be the average of the $t(i,j)$, the cophenetic correlation coefficient c can be expressed as [Carr et al., 1999],

$$\text{Cop. Coeff.} = \frac{\sum_{i<j}(x(i,j)-x)(t(i,j)-t)}{\sqrt{\left[\sum_{i<j}(x(i,j)-x)^2\right]\left[\sum_{i<j}(t(i,j)-t)^2\right]}}.$$

Illustration To illustrate the idea, we take a set of arbitrary point coordinates (Figure 5.6) to see how the cluster verification works and how we obtain the best combination of distance metric and linkage method. For all possible pairs of combinations, we have tested the clustering and provided their corresponding cophenetic correlation coefficient in Table 5.1. In this example, the combination of euclidean distance metric and average-linkage clustering method is found to be the best compared to others. Therefore, we only use the best pair for clustering. Such a clustering will be used for validating appropriate number of clusters.

Cluster Validation The following is a set of several issues for cluster validation.

	city-block	euclidean	sq-euclidean
single-linkage	0.8469	0.8738	0.8203
complete-linkage	0.8460	0.8720	0.8203
average-linkage	0.8560	**0.8833**	0.8240

Table 5.1: Cophenetic correlation coefficient from all possible combinations of distance metric and clustering linkage method applied to the sample points in Figure 5.6. The best combination is the one closest to 1.

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.

2. Determining the correct number of clusters.

3. Evaluating how well the results of a cluster analysis fit the data without reference to external information.

4. Comparing the results of a cluster analysis to externally known results, such as class labels.

5. Comparing two sets of clusters to determine which is better.

In this section, we will be focussing on determining the correct number of clusters.

- If too many clusters are defined, they will be small in size and their elements (even inter-cluster) will be highly similar, but the analysis of many clusters can be difficult.

- If fewer clusters are defined, their larger number of elements will show less similarity to one another, but the smaller number of clusters will be easier to analyse.

Overall, it involves trade-off between number of clusters and similarity of elements in the cluster. The evaluation measures, or indices, that are applied to judge various aspects of cluster validity are traditionally classified into the following major two types:

1. supervised and

2. unsupervised.

Supervised cluster validation measures the extent to which the clustering structure discovered by a clustering algorithm matches some external structure. They are: entropy, purity, precision–recall and F-measure. Those mentioned measures convey how well cluster labels match externally supplied class labels. Supervised measures are often called external indices because they use such information which are not present in the data set. This is not the case that can happen in our problem.

Unsupervised cluster validation does not include external information. Unsupervised measures of cluster validity are often further divided into two classes:

- Cluster cohesion: It refers to compactness or tightness of the cluster. In other words, it resembles how closely related the objects in a cluster are?

  The member of each cluster should be as close to each other as possible.

- Cluster separation: It refers to isolation of the clusters i.e., how distinct or well separated a cluster is from other clusters?

  The clusters themselves should be widely separated. There are three common approaches measuring the distance between two different clusters:

  - distance between the closest member of the clusters,
  - distance between the most distant members, and
  - distance between the centres of the clusters.

Unsupervised measures are often called internal indices because they use only information present in the data set.

In our case, since we do not have external input to fix the number of clusters, we employ unsupervised techniques. The techniques we are going to use to validate the cluster, will be discussed below.

1. **Dunn Index**: The Dunn index [Dunn, 1974] defines the ratio between the minimal intra-cluster distance to maximal inter-cluster distance. In other words, it maximises the inter-cluster distance while minimising the intra-cluster distance. Dunn index for k clusters is expressed as,

$$\begin{aligned} DU_k &= min_{i=1,\ldots,k}\left\{min_{j=i+1,\ldots,k}\left(\frac{d(c_i,c_j)}{max_{m=1,\ldots,k}d(c_m)}\right)\right\} \\ \text{where } d(c_i,c_j) &= min_{x\in c_i, y\in c_j}\|x-y\| \\ d(c_m) &= max_{x,y\in c}\|x-y\| \end{aligned}$$

2. **Davies-Bouldin Index**: It [Davies and Bouldin, 1979] identifies not only the clusters which are far from each other but also compactness. Davies-Bouldin index is defined as,

$$\begin{aligned} DB_k &= \frac{1}{k}\sum_{i=1}^{k} max_{j=1,\ldots,k,i\neq j}\left\{\frac{d(c_i)+d(c_j)}{\|c_i-c_j\|}\right\} \\ d(c_i) &= \left(\frac{1}{n_i}\sum_{x\in c_i}\|x-z_i\|^2\right)^{1/2} \end{aligned}$$

   n_i is the number of elements and z_i the centroid of cluster c_i. Davies-Bouldin Index is minimised for the best number of clusters.

3. **Silhouette Index**: It [Rousseeuw, 1987] calculates the silhouette width for each sample, average silhouette width for each cluster and overall average silhouette width for a total data set. Using this approach each cluster could be represented by so-called silhouette, which is based on the comparison of its tightness and separation. The average silhouette width could be applied for evaluation of clustering validity and also could be used to decide how good is the number of selected clusters. The silhouette width is an average over all observations i.e.,

$$SI_k = \frac{1}{n}\sum_{i=1}^{n}\frac{(\hat{b}_i-\hat{a}_i)}{max(\hat{a}_i,\hat{b}_i)}$$

	clusters					
	1	2	3	4	5	6
Silhouette	—	**0.7650**	0.6523	0.6689	0.7059	0.7983
Dunn	—	**1.2878**	0.6695	0.7906	0.7071	0.5590
Davies-Bouldin	—	**0.0359**	0.0984	0.1329	0.0725	0.0428
Score Function	0.0000	**1.0000**	0.9000	0.8700	0.5620	0.5200

Table 5.2: Cluster validation test for all indices. For a number of clusters: $\{1, 2, 3, 4, 5 \text{ and } 6\}$, every index produces two clusters.

where n is the total number of elements, $\hat{a}_i$ is the average distance between the element i and all other elements in its own cluster and $\hat{b}_i$ is the minimum of the average distance between i and elements in other clusters. Silhouette Index is maximised for the best number of clusters.

4. **Score Function**: As in the Dunn and Davies-Bouldin indices, it [Saitta et al., 2007] is also based on inter-class and intra-class distance. In other words, it can be expressed as,

   - between class distance (bcd)

   $$bcd = \frac{\sum_{i=1}^{k} \|z_i - z_{tot}\|.n_i}{n.k}$$

   where k is the number of clusters of size n, z_i the centroid of cluster c_i having n_i elements and z_{tot} the centroid of all clusters.

   - within class distance (wcd)

   $$wcd = \sum_{i=1}^{k} \left(\frac{1}{n_i} \sum_{x \in c_i} \|x - z_i\| \right)$$

   Now the score function is,

   $$SF = 1 - \frac{1}{e^{e^{bcd-wcd}}}$$

   The higher the value of the SF, the more appropriate the number of clusters. Therefore, it maximises the *bcd* and minimises the *wcd*.

Illustration To attest the cluster validation indices, we take the example in Figure 5.6 and followed the verification results in Table 5.2. Following the cluster verification results in Table 5.2, Figure 5.7 shows cluster validation tests from all indices. In this illustration, every cluster validation index provides 2 clusters.

Real-world example In this illustration, we consider those issues which affect the clustering process and eventually number of clusters. Those issues are

1. shape signatures and
2. cluster validation indices.

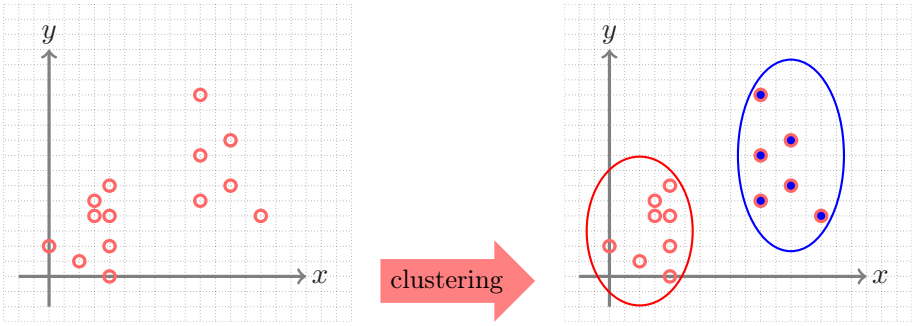


Figure 5.7: Cluster validation indices give two clusters. It is provided in Table 5.2.

		Angular Frequency							
		6	8	10	12	14	16	18	20
Radial Frequency	4	7	9	10	10	16	19	21	21
	6	9	10	17	24	24	33	28	35
	8	9	9	21	23	33	38	38	40
	10	11	11	21	28	31	40	37	34
	12	21	27	31	33	38	33	30	28

(a) Dunn Index

Radial Frequency	4	9	12	11	12	19	21	22	24
	6	10	13	19	25	22	30	29	33
	8	12	14	23	25	31	34	40	40
	10	9	11	21	25	32	38	32	35
	12	21	24	27	36	36	35	33	23

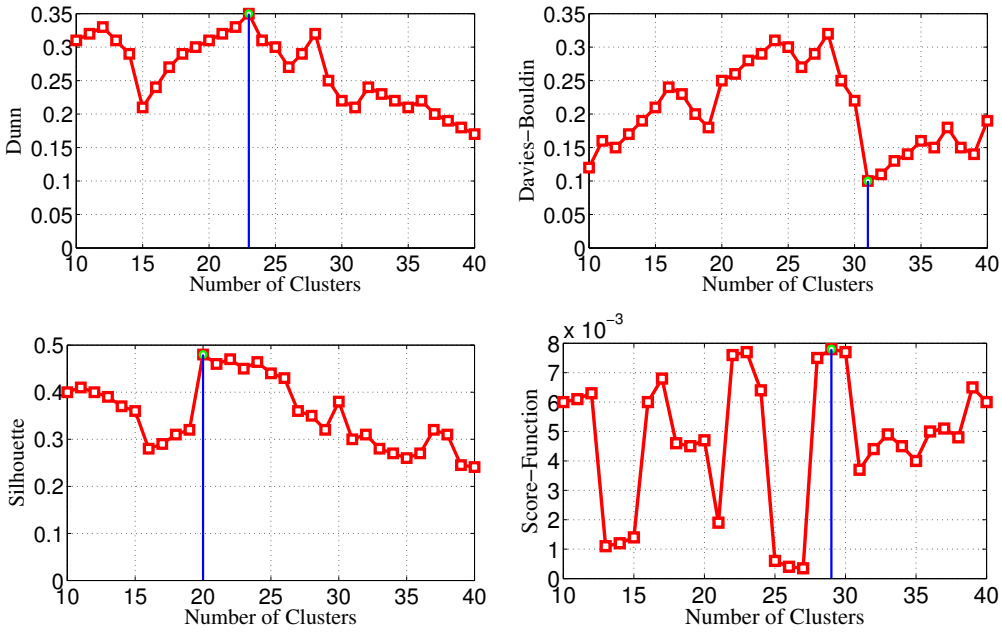
(b) Davies-Bouldin Index

Table 5.3: Number of clusters after cluster validation tests from GFD. Clusters change when feature changes. In order to show the effect of GFD parameters, we have varied radial from 4 to 12 and angular frequencies from 6 to 20 with the step of +2. Two indices: Dunn and Davies-Bouldin are taken.

Shape signatures are of course the primary factor to be considered. Since discrimination power of shape descriptors vary from one to another, the number of clusters is obviously different. As a consequence, cluster validation indices do not always provide similar results. Note that in the aforementioned example, we use coordinate points as features to provide intuitive of the overall clustering process how it works.

Now, different shape descriptors are employed one-by-one. For every shape descriptor, we apply all cluster validation indices and analyse the results. To understand it, we provide a clustering process for GFD.

GFD We have tuned the parameters: radial (4 to 12) and angular (6 to 20) frequencies to get the best combination. Tuning the best combination, is found to be expensive. In Table 5.3, we have shown the effect of shape features by tuning GFD parameters on cluster validation. It means, feature selection affects cluster analysis. Table 5.3 shows an example where there are possibilities to get different clusters from different indices. For every radial and angular frequencies combination, we obtain different number of clusters. For a single combination,



GFD parameters: radial frequency = 6 and angular frequency = 15.

Figure 5.8: A real example of how cluster validation indices behave for GFD. The vertical line in every plot represents the number of clusters, resulting from cluster validation index.

Figure 5.8 shows the way how we select the number of clusters. Every cluster validation index does not provide similar results. For instance, Davies-Bouldin index provides 32 clusters. The output from Davies-Bouldin cluster validation index is shown in Figure 5.9. It only illustrates few clusters.

However, based on the retrieval performance, we determine the best one. Therefore, tuning the best combination of parameters is expensive but it goes at once for a particular dataset.

For remaining shape descriptors like shape context, Zernike moments, $\mathcal{R}-$signature, $\mathcal{D}-$Radon, similar cluster validation process has been carried out. In Section 5.5, effect of the use of all shape descriptors on retrieval performance will be presented. This goes with the appropriate choice of cluster validation index. In other words, full experimental test will be reported for all possible combinations of shape descriptors and cluster validation indices.

5.4.2 Symbol Recognition

The recognition framework principally follows the corresponding relation alignment for matching symbols explained in Chapter 4. At this point, we explain how can we integrate *thick* clustering.

step 1 The first step is to allocate the cluster in which the query thick belongs to. It is shown in Figure 5.10. Cluster selection can be more than one if query symbol has two or more than two *thick* patterns with different shape information. Figure 5.11 shows an example of it. The idea in general, is only applied when query symbol is not taken from dataset used for clustering.

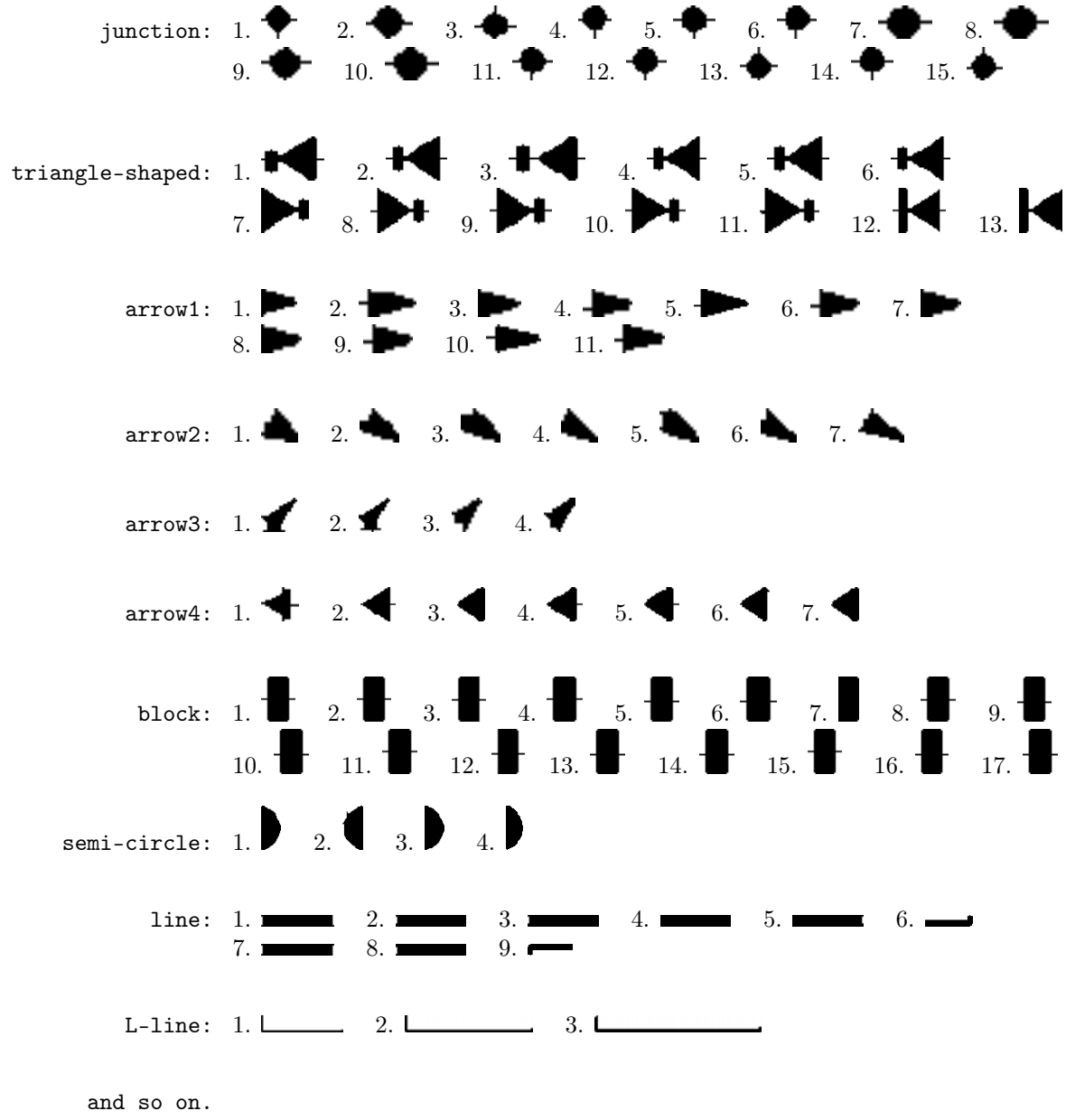


Figure 5.9: Clustering *thick* patterns using GFD shape descriptor – at a glance. It uses Davies-Bouldin cluster validation index. Few clusters are shown. Different clusters are provided with suitable name such as *junction*, *triangle-shaped* and *line*, for visual understanding.

step 2 Once the cluster(s) is(are) selected, then the symbols related to those *thick* patterns i.e., corresponding symbols are taken for matching. For matching symbols, relations are matched for every pair via ARG. Just a reminder, in Section 4.5 of Chapter 4, a basic framework of the ARG matching has been discussed.

Matching is simple and straightforward when all symbols have a single *thick* pattern. Database symbol ranking is therefore exactly similar to what we described before. This is however not a case in real-world application. As said before, symbols may have variable number of *thick* patterns and therefore different clusters can be selected even from a single query symbol. This

means that corresponding database symbols will obviously be different. In such a case, ranking procedure is becoming a bit tricky. In what follows, we explain matching strategy and database symbol ranking in two two different cases: presence of a single *thick* and many *thick* patterns in a query symbol.

Consider any symbol $\mathbf{S}$, there are $\mathcal{K}$ number of *thick* patterns. As a consequence, based on our symbol description discussed in Section 5.4.1, there are $\{G_\kappa\}_{\kappa=1,\ldots,\mathcal{K}}$ ARGs.

case 1 A single *thick* pattern in a query symbol $\mathbf{S}^q$:

In such a case, there exists a single query graph which is similar to what is expressed in Chapter 4. Database symbol $\mathbf{S}^d$ still has $\{G^d_\kappa\}_{\kappa=1,\ldots,\mathcal{K}}$ ARGs. Now the question is how to compute distance between two symbols: query and database. To handle this, we take the minimum matching score from all possible matching,

$$\text{Dist.}(\mathbf{S}^q, \mathbf{S}^d) \quad = \quad \min_{\kappa} \text{Dist.}(G^q_1, G^d_\kappa), \kappa = 1, \ldots, \mathcal{K},$$

where those graphs in $\mathbf{S}^d$ are used only if the *thick* patterns are related to the particular cluster. For example, if only two *thick* patterns in $\mathbf{S}^d$ are related with the query *thick* then the distance between $\mathbf{S}^q$ and $\mathbf{S}^d$ is min $[\text{Dist.}(G^q_1, G^d_1), \text{Dist.}(G^q_1, G^d_2)]$.

case 2 Many *thick* patterns in a query symbol $\mathbf{S}^q$:

In case where query symbol $\mathbf{S}^q$ has two or more than two *thick* patterns, every corresponding graph G^q_κ is matched with the graphs $G^d_{\kappa'}$ from the database symbol $\mathbf{S}^d$. To make it simpler, let us start with a graph G^q_1 from a query symbol and compute distance with κ database graphs $G^d_{\kappa'}$. This process goes similar to what is explained in **case 1**. We keep on continuing for all query graphs G^q_κ. If several different database symbols are related, then their distances are stored for ranking.

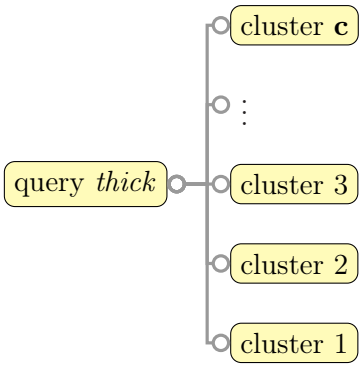
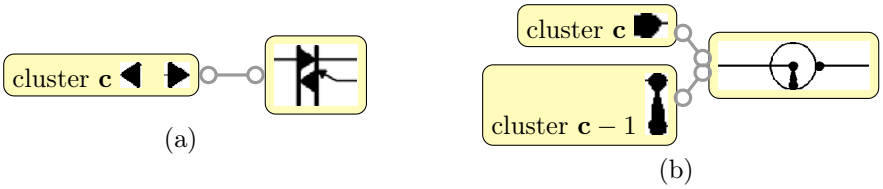


Figure 5.10: Searching for the right clusters where query *thick* pattern belongs to.

To illustrate it with an example, consider a query symbol $\mathbf{S}^q$ with two *thick* patterns and and three different database symbols $\mathbf{S}^{d_1}$, $\mathbf{S}^{d_2}$ and $\mathbf{S}^{d_3}$ with respectively three, two and four *thick* patterns. Now, we have

$\{G^q_\kappa\}_{\kappa=1,2}$ ARGs in $\mathbf{S}^q$ (see Figure 5.11 (a)), and

$\{G^{d_1}_{\kappa'}\}_{\kappa'=1,2,3}$ ARGs in $\mathbf{S}^{d_1}$,
$\{G^{d_2}_{\kappa'}\}_{\kappa'=1,2}$ ARGs in $\mathbf{S}^{d_2}$ and
$\{G^{d_3}_{\kappa'}\}_{\kappa'=1,2,3,4}$ ARGs in $\mathbf{S}^{d_3}$.



(a) (b)

Figure 5.11: *thick* pattern(s) in cluster(s) and its (their) corresponding symbol. A symbol can have more than one cluster. It is primarily due to different shaped *thick* patterns.

In this example, the first query *thick* is related to the first two *thick* patterns of database symbol $\mathbf{S}^{d_1}$. Another query *thick* is related to the first *thick* pattern of all database symbols. Such corresponding symbols are obtained with the help of searching right clusters as mentioned in **step 1**. Matching goes like this,

$$\begin{aligned}
\text{Dist.}(\mathbf{S}^q, \mathbf{S}^{d_1}) &= \min\left[\text{Dist.}(G_1^q, G_1^{d_1}), \text{Dist.}(G_1^q, G_2^{d_1})\right], \\
\text{Dist.}(\mathbf{S}^q, \mathbf{S}^{d_1}) &= \text{Dist.}(G_2^q, G_1^{d_1}), \\
\text{Dist.}(\mathbf{S}^q, \mathbf{S}^{d_2}) &= \text{Dist.}(G_2^q, G_1^{d_2}) \text{ and} \\
\text{Dist.}(\mathbf{S}^q, \mathbf{S}^{d_2}) &= \text{Dist.}(G_2^q, G_1^{d_3}).
\end{aligned}$$

Since both query graphs (due to two different *thick* patterns) are separately used to match with the same database symbol, we take the minimum distance that can be either from G_1^q or G_2^q. In every matching, it is important to notice that only interested graphs taken for matching even though a symbol can have many graphs, thanks to clustering.

Finally, in both cases, once we have computed distances between all database symbols, we can follow the similar ranking strategy mentioned in Section 4.5.2 of Chapter 4.

5.5 Experiments

In order to measure the impact compared to the last experimental results mentioned in Section 4.6 of Chapter 4, we have used the same dataset and experimental protocol.

We perform a series of tests and confront our proposed methods with existing ones. For a quick details and as said in Section 5.2.2, our tests mainly consist of integrating shape description of vocabulary with relations in two different ways.

1. to label vertices in ARG representation and

2. to cluster *thick* patterns.

In *item 1*, we first select the appropriate choice of global signal-based shape descriptors to label vertices. We then integrate vertex and edge signatures where we examine almost all possible combinations of vocabulary types and select the best one. On the other hand, in *item 2*, clustering of *thick* patterns is aimed to take similar ones in a cluster or group which eventually increase retrieval performance via corresponding relations alignment.

In the following section, we explain more about how all experimental tests go.

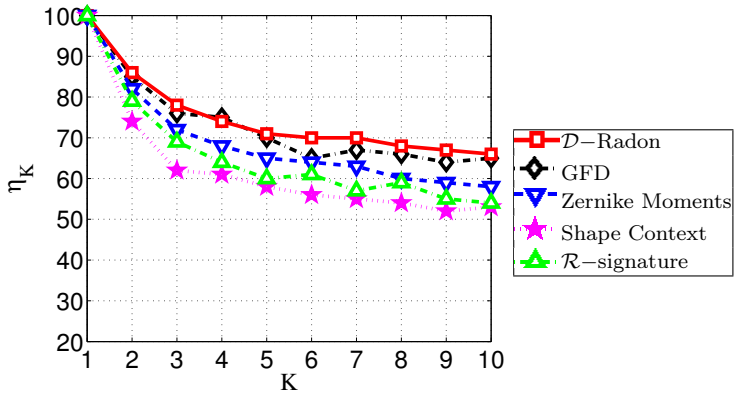


Figure 5.12: Average retrieval efficiency over requested list: 1 to 10 using global signal-based shape descriptors as vertex signature in our ARG framework.

5.5.1 Experimental Results

Vertex Signature and its Integration with Edge Signature

Applying shape descriptors to the vocabulary types (i.e., vertices in ARG) provides independent shape information, which is different from spatio-structural information via pairwise spatial relations. Therefore, we study its discriminant power separately for symbol recognition before integrating with spatial relations.

Vertex Signature through Shape Description We first employ shape descriptors as vertex signatures (without using the edge signature). Figure 5.12 shows the retrieval performance of it. We then compare its performance when applying shape descriptors on the whole shape shown in Figure 4.24 of Chapter 4. In comparison, shape features to label vertices provide better results. However, it does not provide surprising difference.

In Figure 5.12, $\mathcal{D}-$Radon as vertex signature outperforms all, with a marginal difference with GFD and Zernike moments. While, shape context and $\mathcal{R}-$signature are lagging behind with noticeable differences. In order to employ shape context, we have taken different numbers of sample points from one vertex to another since elementary visual primitives are ranging from a single dot (i.e., *extremity*) to hundreds of pixels (i.e., *thick* pattern). Therefore, it is possible to maximise the number of sample points for each vertex, otherwise the use of shape context is restricted by a single dot.

Integrating Vertex Signature with Edge signature Finally, we integrate the vertex signature with the relational signature to build complete ARG. While integrating, we have examined the use of shape signature from one vocabulary type to another as well as all possible combinations such as *thick* and *circle*, *thick*, *circle* and *corner*. In the reported results, we present a few combinations. Note that in all combinations, *thick* has been included since it provides significant shape variations. While integrating, we consider not only retrieval efficiency but also running time cost.

Fig. 5.13 shows results from using different shape signatures which are integrated with relations i.e., edge signatures. In this illustration, GFD performs well in comparison to others. It is important to notice that not all descriptors can improve the retrieval performance: $\mathcal{R}-$signature in Figure 5.13, for instance. Besides, signatures for all vertices is not always a better choice,

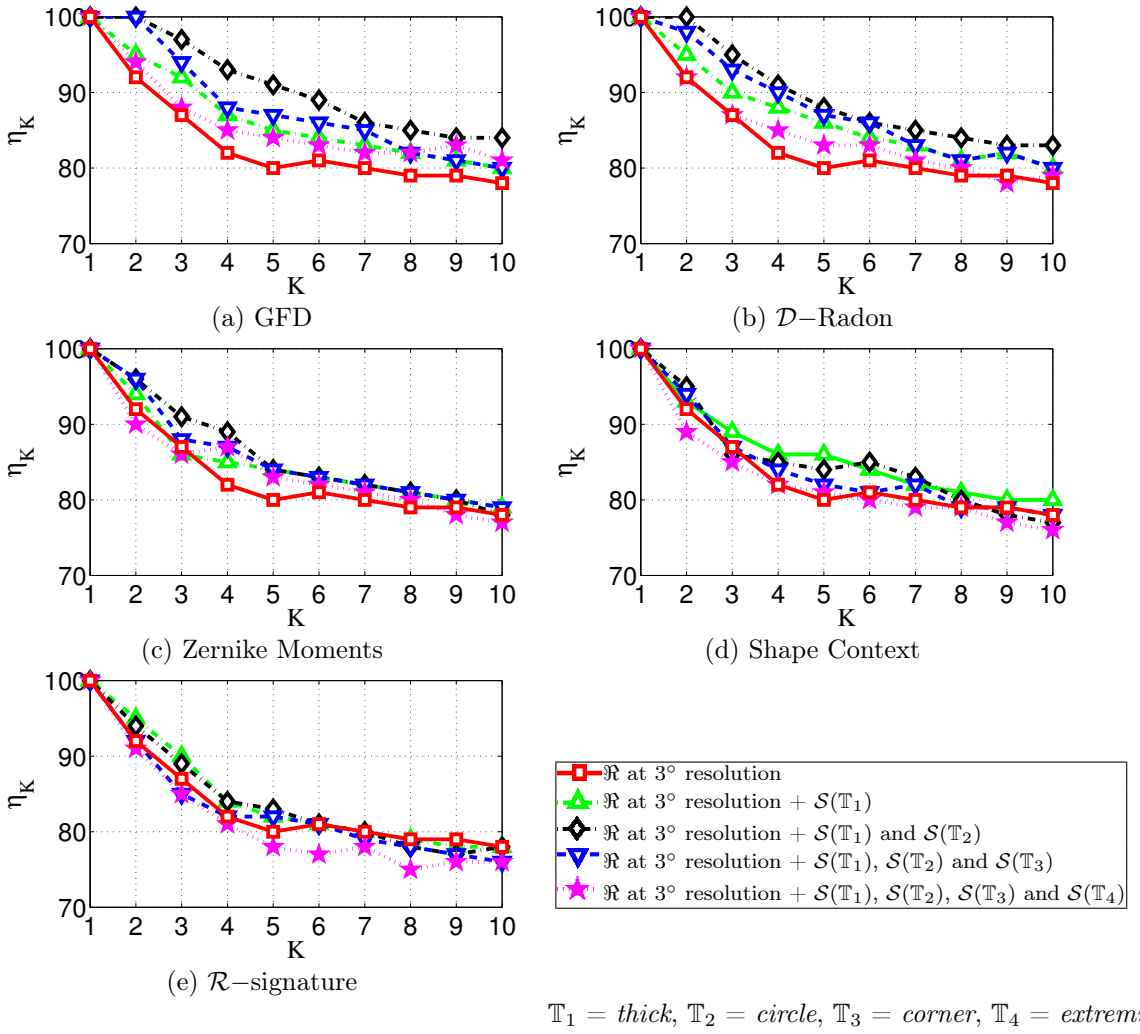
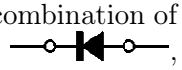
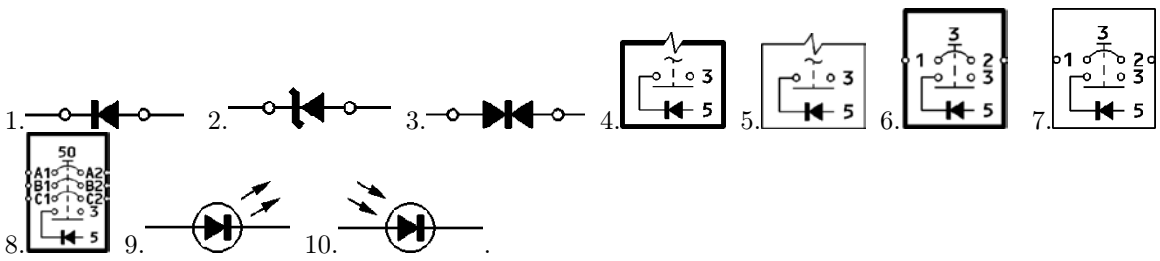


Figure 5.13: Average retrieval efficiency over requested list: 1 to 10 using global signal-based shape descriptors as vertex signature in our ARG framework. For every shape descriptor, a few combinations of signatures on vertices has been demonstrated. In average, a combination of *thick* and *circle* performs better for all descriptors, which is then taken for comparison in Figure 5.14.

but computationally expensive instead. For all descriptors, we have found that a combination of *thick* and *circle* vocabulary types provides better results in average. For example  , using GFD vertex signature on *thick*  and *circle* ○ ○. The retrieval symbols are,



Since substantial advancement has been achieved from the vertices, labelled with *thick* and *circle* vocabulary types with respect to the retrieval performance of our relational signature,

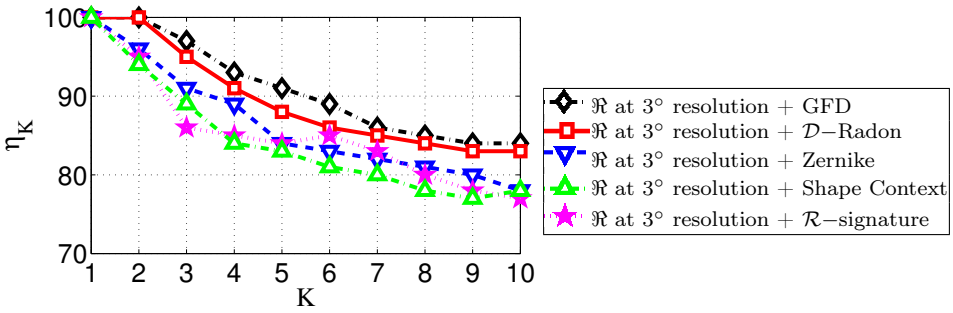


Figure 5.14: Comparison. Average retrieval efficiencies over requested list: 1 to 10 through the integration of global signal-based shape descriptors as vertex signature with edge signatures in our ARG framework. Only a combination of signatures on *thick* and *circle* (see Figure 5.13) is taken for comparison.

such a combination is taken for comparison. Comparison is provided in Figure 5.14 for all shape descriptors.

Clustering of *thick* patterns

The problem of shape variation of *thick* patterns in our vocabulary remains in earlier experiment of Chapter 4. This test is aimed to see how the integration of *thick* pattern selection advances the retrieval performance.

Note that *thick* pattern selection is based on clustering mentioned in Section 5.4.1. As said in Section 5.4, clustering performance is based on shape signatures and cluster validation index. Therefore, in the following, we take account both and find the best combination for this application. Figure 5.15 shows the comparison of performance of cluster validation indices for a major set of shape descriptors used throughout the thesis. In the tests, we observe and analyse retrieval performances on a one-to-one basis. Overall, $\mathcal{D}-$Radon outperforms all, but GFD goes almost equally having a marginal difference. Zernike moments, shape context and $\mathcal{R}-$signature are lagging behind.

Selection of shape descriptors does not only provide a complete process while it needs to account cluster validation indices, on the other hand. As said previously, different cluster validation index provides different results. As a consequence, overall retrieval performance is affected. To illustrate the aforementioned statement, let us take a close look into the results provided in Figure 5.15. For all shape descriptions, Dunn and Davies-Bouldin indices provide almost similar advancement, while the remaining Silhouette and Score-Function do not. Therefore, either Dunn or Davies-Bouldin index will be the appropriate choice for this application.

5.5.2 Experimental Results Analysis

In this section, we shortly analyse the models and or methods developed in this chapter based on the obtained experimental results. As in previous experiments in Chapter 4, we consider two different elements while accounting overall performance. They are

1. retrieval efficiency and

2. execution time.

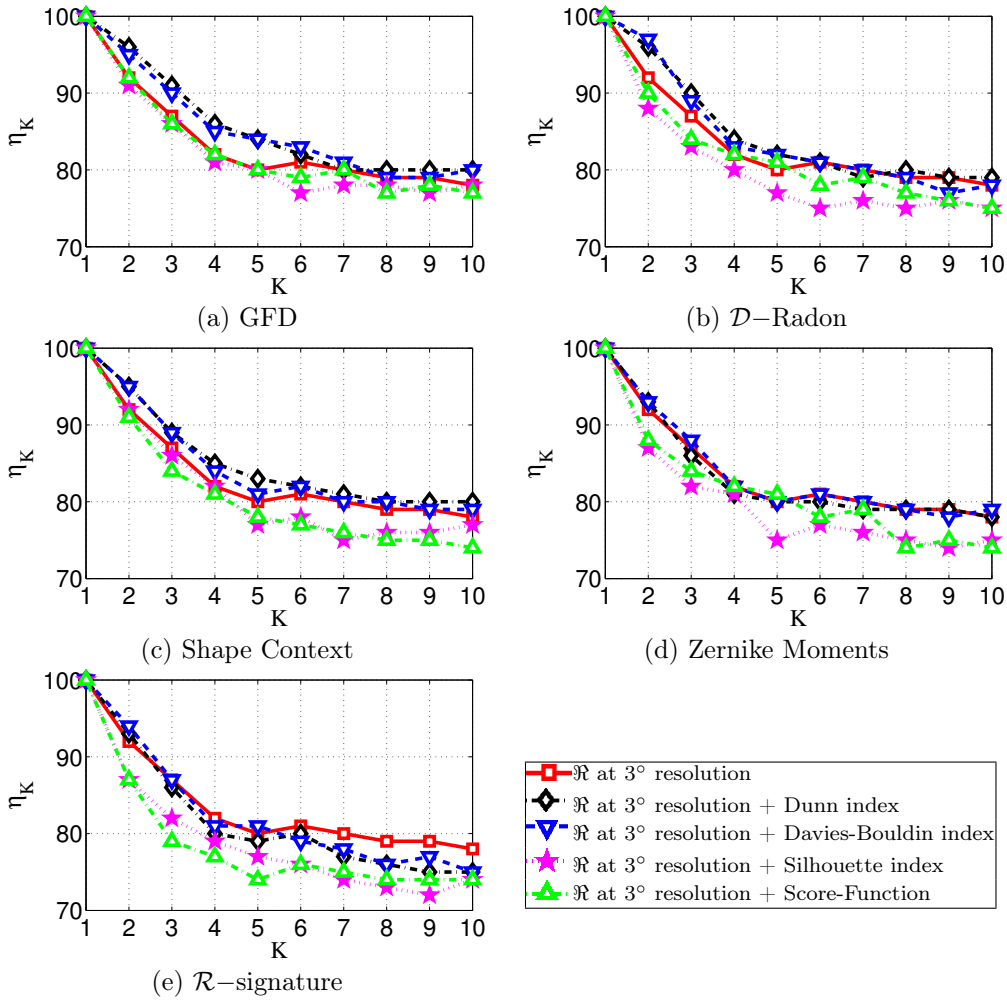


Figure 5.15: Average retrieval efficiency over requested list: 1 to 10 using global signal-based shape descriptors for *thick* patterns clustering and several different cluster validation indices.

The substantial retrieval difference made by shape signatures as vertex signature with respect to while using overall symbol shape, provides the fact that symbol description is equally important. However, it is computationally expensive since it requires t different shape signatures for t number of vocabulary types. Such a vertex signature is then intelligently integrated with edge signature. While integrating, we have received better results from the combination of shape signatures on vertices labelled with *thick* and *circle* since relational signatures provide sufficient information about the remaining vocabulary types such as *corner* and *extremity*. Therefore, it reduces matching cost of corresponding signatures between the vertices.

Figure 5.16 provides visual demonstration for a few queries. Two global signal-based descriptors: $\mathcal{D}-$Radon and GFD show almost similar behaviour except a running time difference, while others do not really increase the performance. Shape context and Zernike moments follow $\mathcal{D}-$Radon and GFD. In our application, shape context is highly restricted by the number of samples. However, we have tried to maximise it by selecting differently in different vertices, for instance. $\mathcal{R}-$signature is found to be always lagging behind.

Along with relational signature matching, query *thick* pattern selection via clustering advances retrieval performance. However, it does not provide a surprise difference due to the fact

that not all query symbols contain *thick* pattern in their vocabulary type sets. In other words, absence of *thick* vocabulary type means ranking has been made only through relational signatures alignment, eventually no change in retrieval performance.

In particular, comparison between GFD and $\mathcal{D}-$Radon in all cases throughout a series of experimental tests provides that GFD yields interesting results especially when also taking time complexity into account. Table 5.4 provides average running time for all.

Methods	Time (in sec)
1. Global Shape Descriptors	
1.1. $\mathcal{R}-$signature [Tabbone et al., 2006]	02
1.2. Zernike moments [Kim and Kim, 2000]	13
1.3. GFD [Zhang and Lu, 2002]	09
1.4. Shape context [Belongie et al., 2002]	32
1.5. $\mathcal{D}-$Radon	67
2. Our Method	
2.1. Edge signature (relations)	04
2.2. Integrating signatures (relations + shape):	
– Edge + Vertex ($\mathcal{R}-$signature)	07
– Edge + Vertex (ZM)	21
– Edge + Vertex (GFD)	17
– Edge + Vertex (SC)	39
– Edge + Vertex ($\mathcal{D}-$Radon)	123

Table 5.4: Average running time (in sec) for a single pair (*cf.* Chapter 3 and 4).

5.5.3 Advantages and Limitations

As in the previous chapter, we collectively discuss advantages and disadvantages of the proposed method in the following points:

1. Symbol Description – ARG based symbol description provides a fixed and powerful relational representation including generality and extensibility. Therefore, it has been attested from the integration of shape descriptions on vertices, keeping original ARG framework validated in Chapter 4.

   Overall, it clearly addresses the combination of the best of two different approaches: statistical structural.

2. Matching – The proposed method is just an extension of the previous Chapter 4. It brings easy and straightforward matching of statistical signatures.

3. Recognition – It advances the retrieval performance compared to Chapter 4. Our previous method that uses relations only (*cf.* Chapter 4), is limited to those symbols which have at least two different vocabulary types. Now, our method can apply even when there exists no relations due to the additional use of vertex signature.

5.6 Conclusions

In this chapter, we have presented the concept to integrate shape signatures with relational signatures. We have attested the use of spatial relations in Chapter 4 that provide a powerful

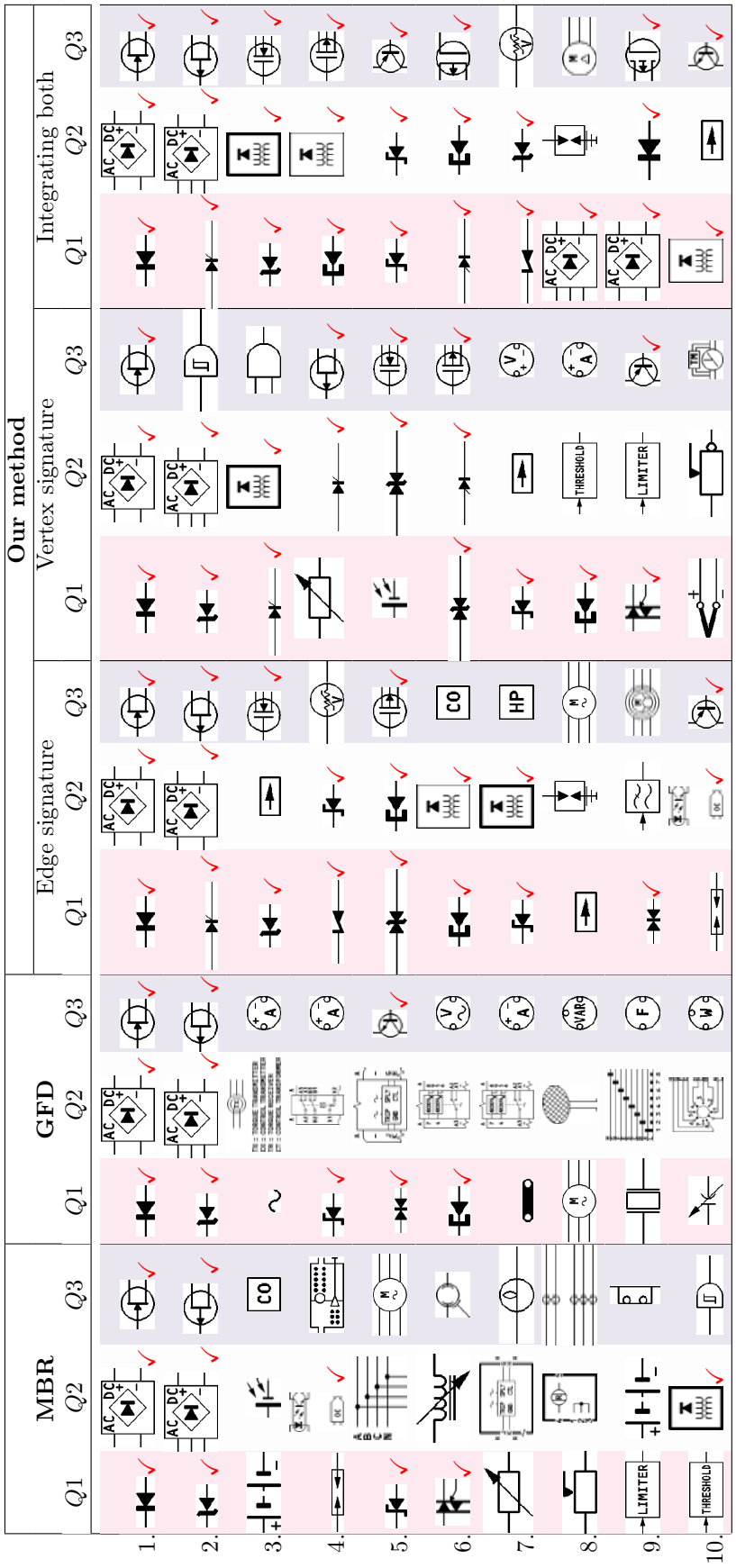


Figure 5.16: Visual illustration of symbol ranking at the output for a few queries: $Q1$, $Q2$ and $Q3$, showing ✓ for true retrieval and false, otherwise. The first symbol on the top always corresponds to the chosen query. Symbols are ranked from top to bottom based on decreasing order of similarity.

representation, conveying how visual primitives are connected. Keeping the ARG based symbol description framework, we have presented the use of shape signatures in two different ways.

- Firstly, shape signatures are labelled as vertex signatures which are then integrated with relational signatures. As a consequence, the description is used to globally recognise structure by comparing the signatures.
- Secondly, we have observed the behaviour of several different shape features by applying to the vocabulary which have significant shape variations. We employ *thick* patterns for clustering and use the related clusters with respect to query for relational signature alignment.

  Clustering of thick patterns thus opens a global concept that it can be applied for any other visual primitives.

Overall, we bring an attention to the use of hybrid approach in symbol recognition since it combines both the worlds: structural and statistical. A series of experimental tests over graphical symbol dataset validated the methods.

5.7 What is Next?

In our ARG based symbol description, each vertex has a distinct vocabulary type containing different shape and size information. In this chapter, to discriminate visual primitives, shape signatures are used in addition to the visual links (i.e., spatial relation in Chapter 4). Considering the application, our proposed method performs well. Due to shape features used for labelling visual primitives, the method can also apply even when there exists no relation. However, it does not really provide flexibility for retrieving significant known part of graphical elements from the composite form. The primary reason is that the method computes spatial relations between the vocabulary types i.e., between the groups of similar visual primitives – which is of course avoids NP-hard graph matching problem.

Therefore, in the following Chapter 6, we will introduce a concept of bag-of-features approach using spatial relations. This new framework computes all possible relations between individual visual primitives and stores them in the form of spatial bag, categorised based on topological information. This is more dedicated for symbol recognition from composite form.

End of Chapter

Keywords

Vocabulary categorisation,
Spatial-Bag-Of-Features,
Symbol Recognition.

6 Spatial-Bag-Of-Features for Symbol Recognition

Contents

6.1 Organisation of the Chapter . . . 146
6.2 Introduction . . . 146
6.2.1 Related Work . . . 147
6.2.2 Outline of the Proposed Method . . . 148
6.3 Spatial-Bag-Of-Features . . . 149
6.3.1 Vocabulary Categorisation . . . 149
6.3.2 Spatial Reasoning . . . 152
6.4 Symbol Recognition . . . 153
6.4.1 Relation Matching . . . 153
6.4.2 Symbol Recognition and Ranking . . . 154
6.5 Experiments . . . 154
6.5.1 Datasets and Evaluation Metric . . . 155
6.5.2 Experimental Results . . . 155
6.5.3 Experimental Results Analysis . . . 155
6.6 Advantages . . . 156
6.7 Conclusions . . . 160

Foreword. In Chapter 3, we have introduced and validated DTW–Radon shape descriptor for graphics recognition. This is limited to only isolated graphical patterns and therefore cannot be applied for real-world applications where graphical elements are connected with other elements and texts. In such a situation, we developed an idea of using visual links via spatial relations between the extracted visual primitives that compose the complete symbol (*cf.* Chapter 4). Furthermore, in Chapter 5, we have used shape features on the extracted visual primitives in two different ways in addition to the spatial relations framework validated before. These are however, not well suited for retrieving significant known part of graphical elements in the composite form.

In this chapter, we present a method for symbol recognition based on spatial-bag-of-features (BOFs) that are computed from extracted visual primitives which we called 'vocabulary'. Our feature consists of pairwise spatial relations from all possible combinations of individual visual primitives. The key characteristic of the overall process is to use topological information to guide directional relations. Consequently, directional relation matching takes place only with those candidates having similar topological configurations. A series of experiments over several different datasets were performed to validate the method. Experimental tests provide interesting results by clearly establishing user-friendly symbol retrieval applications and its robustness towards symbol spotting problems. ■

6.1 Organisation of the Chapter

This chapter is organised as follows. In Section 6.2, we introduce the main idea of the method. It includes a review of related work in Section 6.2.1 and an outline of the proposed method in Section 6.2.2. We explain the way we handle BOFs using spatial relations in Section 6.3. We derive a symbol recognition method and explain it in Section 6.4. Full experimental results are reported in Section 6.5 and confront our method with current state-of-the-art approaches. It includes a comprehensive experimental result analysis. In Section 6.6, we provide an extension of the proposed method. Finally, the whole chapter is concluded in Section 6.7.

6.2 Introduction

We introduced a method for spatial relation matching in Chapter 4. In the corresponding ARG description, vertices represented grouped vocabulary types rather than using every detected individual elementary parts. This was to avoid the underlying NP-hard sub-graph matching problem. Such a description is limited to only those symbols which have at least two or more than two vocabulary types. The problem is that this is not always the case. Let us elaborate the situation with the help of a visual illustration. In Figure 6.1, two different vocabulary types (*circle* and *corner*) are detected, and are used to represent two different vertices in our ARG based symbol description. In the absence of *circle* or to represent only rectangle via ARG there exist no relation.

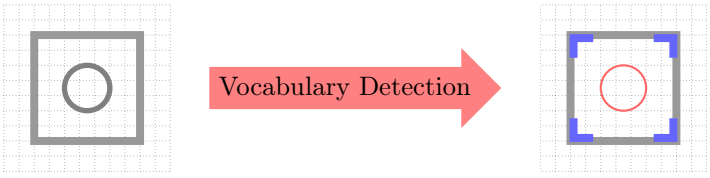


Figure 6.1: A symbol and its visual elementary parts i.e., primitives: *circle* and four *corner*s.

To handle such a situation, an immediate solution is to use intra-type spatial relations, which still needs to have at least two visual elementary parts in a single vocabulary type. Therefore, we have integrated shape description in vertices and examined it in Chapter 5. Such shape signatures integration with spatial relations yields a complete ARG description of the symbol. This eventually costs while matching. In addition, shape signatures can be changed to a some extent if one of the vocabulary parts is missed while extracting. For instance, in Figure 6.1, shape description changes if one of the *corners* is missed. Moreover, the whole method does not provide flexibility to retrieve significant known parts in the composite form since it takes a group of similar visual primitives as a single vertex. To handle this, we use a BOFs approach using

spatial relations for all possible combinations of individual visual primitives. This is going to be the core part of the chapter.

6.2.1 Related Work

As mentioned in Chapter 2, graphics recognition has rich state-of-the-art. In structural approaches in particular, ARG or RAG and tree-based structures are the prominent methods to represent symbols. In general, methods use spatial relations between line segments, visual primitives, curves, circles and so on. However, they hardly consider or solve the problem of symbol localisation within the real technical documents. In addition, those methods offer intense computational complexity. BOFs approach will provide a key solution.

Let us take a general definition of BOFs model with an example. An image can be treated as a document, and features extracted from the image are considered as the 'visual words'. Computer vision researchers used such a concept for image representation. It allows a dictionary-based modelling, and each document looks like a 'bag', which contains some words from the dictionary. Figure 6.2 shows an example where each item represents a 'feature' i.e., 'visual word'. In this illustration, we simply take the the frequencies of item in the specified feature space i.e., equally spaced vertical projections. In general, extraction of several local key-points (or regions) are considered to be basic elements, 'words'. As an example, the methods such as Hessian-Affine descriptor [Mikolajczyk and Schmid, 2004] and SIFT features [Lowe, 2004] have been increasingly used. Besides, regular grid-based method like [Li et al., 2005], is another simple idea to detect features.

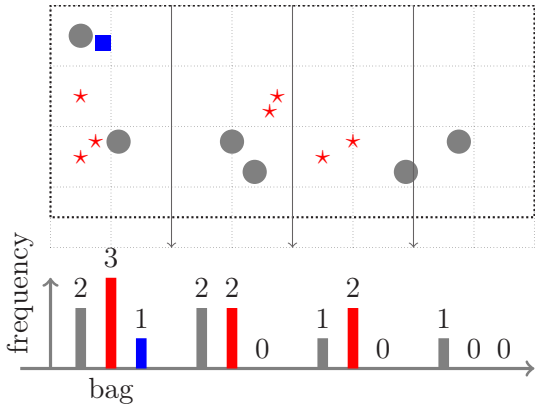


Figure 6.2: Bag-Of-Features model – an example.

In graphics recognition, spatial-bag-of-features approach does not really exist in the literature. As said before, the approach is well-known in computer vision domain. A comprehensive idea of text-retrieval approach has been explained in [Philbin et al., 2007, Jegou et al., 2008], aimed for fast retrieval at run time. To handle this, interests points and descriptors are found in every dataset image and the descriptors are the clustered (using k-means, for instance) and quantised to give a 'visual word' representation for each image. Such 'visual words' are used for indexing. In addition, re-ranking can be performed by using integrating spatial verification system, estimating an affine homography from a single image correspondences between query image and each target image. This means that the possibility of fast matching via the use of spatial properties of the large vocabularies has been studied in [Philbin et al., 2007]. The similar concept can be found in [Marszalek and Schmid, 2006, Jégou et al., 2009]. In the recent

literature, spatial-bag-of-features has been studied to encode geometric information of objects and use it for efficient image retrieval [Cao et al., 2010]. It uses the projection of local features into a specified number of bins – bins can be a linear projection of 10 equidistant angle in $[0°, 180°]$, or partitioning the images into increasingly fine sub-regions and computing histograms of local features [Lazebnik et al., 2006], for instance. As an example, we can follow Figure 6.2 to realise it. Under the purview of computer vision, BOFs method has the following advantages.

1. its compactness i.e., requires less storage requirements [Jégou et al., 2009];
2. rapid search due to inverted file system [Jégou et al., 2009] and or using *cosine* similarity computation by conducting dimension reduction and leveraging residual error information, and efficient indexing [Zhang et al., 2009]

Inspiring from the aforementioned concept and advantages, we introduce it in graphics recognition in a different way. In our context, pairwise directional relations are considered as the candidates of 'features'. We first categorise extracted visual primitives based on the topological information. To illustrate it, Figure 6.3 shows a few topological categories. Then we compute directional relations to give metrical details of them. While doing recognition, directional relations matching takes place between the candidates which share similar topological configurations and similar vocabulary type information. Such a topological guidance significantly reduces matching time.

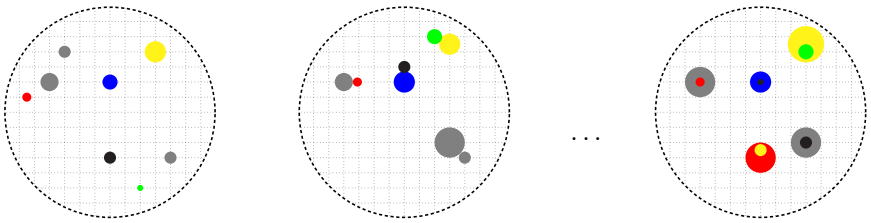


Figure 6.3: Our spatial-BOFs model – a symbolic example. Each item represents a visual primitive and its colour represents a vocabulary type information.

Overall, our method does not need ARG representation to describe the complete symbol nor does it require high computational complexity. Furthermore, it can be applied for all types of symbols and served a robust way towards symbol spotting applications, for instance. Full experiments will be reported in Section 6.5. In the following section, we give outline of the proposed method.

6.2.2 Outline of the Proposed Method

Our spatial-BOFs approach handles all possible combinations of individual visual primitives. The key characteristics of overall process are two-fold:

1. vocabulary categorisation and
2. spatial reasoning.

We first categorise visual primitives in a pair, based on topological relations (*cf.* Figure 6.3). These vocabulary type information of these visual primitives are predefined. We then compute directional relations between them. This means that directional relations are stored in

the particular category based on their topological information. Consequently, directional relation matching takes place only with those which share similar topological and vocabulary type information. Figure 6.4 shows a block-diagram to illustrate the concept of the proposed method.

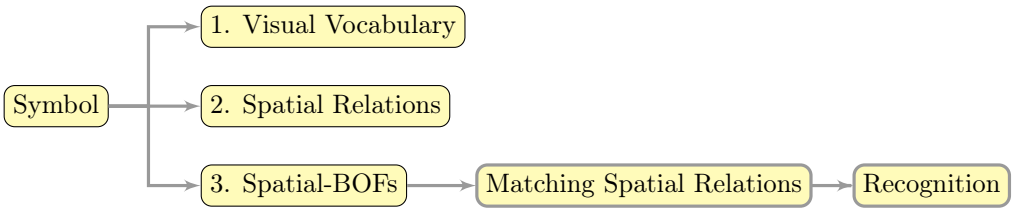


Figure 6.4: An architecture of the symbol recognition method based on spatial-bag-of-features.

Overall, our method is still suffering combinatorial problem when spatial relations are stored. But for recognition, it does not require all relations for matching nor does it need to build arbitrarily large ARG to describe the symbol, thanks to vocabulary categorisation. Besides, we can also use a few but significant/relevant relations rather than taking all from a test symbol as a query. Such a collection of relations as a query states the problems related to user-friendly symbol retrieval as well as symbol spotting, for instance.

Our method is motivated from what is mentioned in Appendix C. In Appendix C, it uses a first–order logic (FOL) by using spatial relations and visual elementary parts. Both the methods use spatial relations for all possible visual primitives.

In Appendix C, we use inductive logic programming (ILP) to produce common characterisation (i.e., common relations) of the user-defined sub-set of the symbols. In this method, we aim to retrieve similar symbol according to the query posed. Query can be either any symbol or a set of relations between the visual primitives that describes significant part of the symbols. For more detail, we will report in Section 6.6.

6.3 Spatial-Bag-Of-Features

We address a BOFs approach with the help of terminal vocabulary and spatial relations existing between them. In contrast to those methods attested in previous chapters, it is rather another way to use vocabulary and spatial relations.

As said in Section 6.2.2, before computing directional relations between them, we handle vocabulary categorisation according to their topological information:

step 1 vocabulary categorisation and

step 2 spatial reasoning.

In what follows, we explain them in detail.

6.3.1 Vocabulary Categorisation

Keeping the same vocabulary type information as mentioned in the previous chapters, we categorise them in a pair based on their topological configurations. In general, visual primitives are categorised into six categories: *disconnected*, *externally connected*, *overlap*, *contain/inside*, *cover/covered by*, and *equal*. Figure 6.5 shows a framework for vocabulary categorisation. In the generic form, we can express it as,

$$\mathcal{C} \quad = \quad \{(\mathcal{C}_1, \mathcal{C}_{disconnected}), (\mathcal{C}_2, \mathcal{C}_{externally\ connected}), (\mathcal{C}_3, \mathcal{C}_{overlap}), \ldots, (\mathcal{C}_6, \mathcal{C}_{equal})\}.$$

For simplicity, we rewrite it as $\{\mathcal{C}_{cat}\}_{cat=1,\ldots,6}$.

To realise whole process and provide an intuitive feeling how it behaves, a complete illustration of vocabulary categorisation is provided in Figure 6.6. In each category, vocabulary type information is also kept.

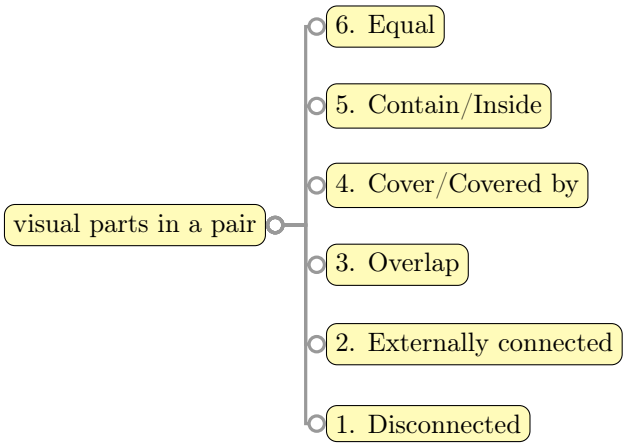


Figure 6.5: Vocabulary categorisation in a pair. All possible pairwise combinations of visual primitives are categorised based on their topological configurations.

For every combination of visual primitives (let us say $\wp_i$ and $\wp_j$), we employ the 9-intersection model [Egenhofer and Herring, 1991, Renz and Nebel, 1998] as explained in Section 4.3.1. It is used to obtain topological relations through the intersections of the boundaries ($\partial*$), interiors ($*^o$) and exteriors ($*^-$) of two sets $\wp_1$ and $\wp_2$. Their definitions using set of operations $=$, $\neq$, $\subseteq$ and $\cap$ are taken, as mentioned in [Güting, 1988]:

$$
\begin{array}{lcl}
\wp_1 = \wp_2 & := & \text{points}(\wp_1) = \text{points}(\wp_2) \\
\wp_1 \neq \wp_2 & := & \text{points}(\wp_2) \neq \text{points}(\wp_2) \\
\wp_1 \ \textit{inside} \ \wp_2 & := & \text{points}(\wp_1) \subseteq \text{points}(\wp_2) \\
\wp_1 \ \textit{outside} \ \wp_2 & := & \text{points}(\wp_1) \cap \text{points}(\wp_2) = \emptyset \\
\wp_1 \ \textit{intersects} \ \wp_2 & := & \text{points}(\wp_1) \cap \text{points}(\wp_2) \neq \emptyset
\end{array}
$$

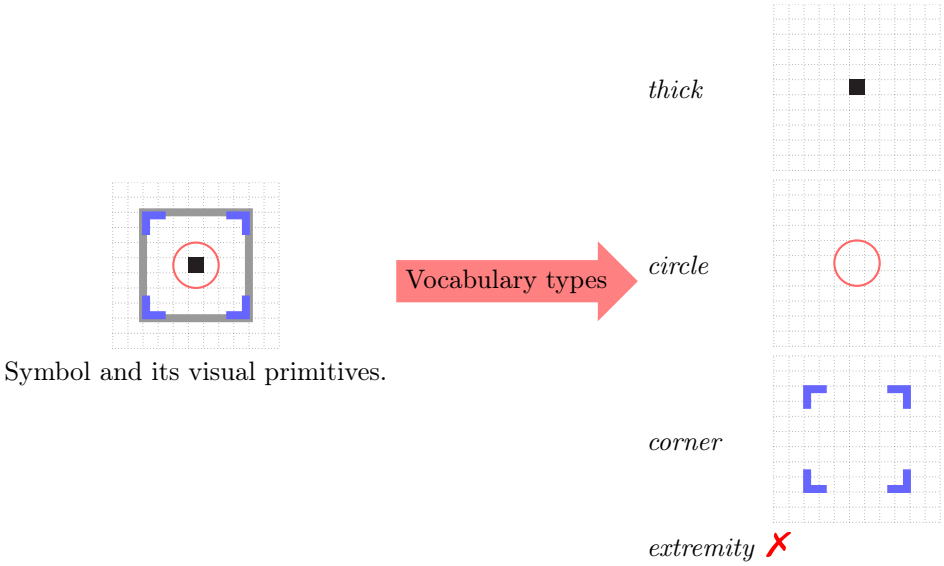
Since the *intersects* definition covers both *equal* and *inside*, they must be separated. Therefore, the previous definitions have been augmented with the consideration of boundary and interior so that the *overlap* and *neighbour* can be distinguished [Pullar and Egenhofer, 1988].

$$
\begin{array}{lcl}
\wp_1 \ \textit{overlaps} \ \wp_2 & := & \partial\wp_1 \cap \partial\wp_2 \neq \emptyset \text{ and} \\
 & & \wp_1^o \cap \wp_2^o \neq \emptyset \\
\wp_1 \ \textit{neighbour} \ \wp_2 & := & \partial\wp_1 \cap \partial\wp_2 \neq \emptyset \text{ and} \\
 & & \mathbb{A}^o \cap \mathbb{B}^o = \emptyset
\end{array}
$$

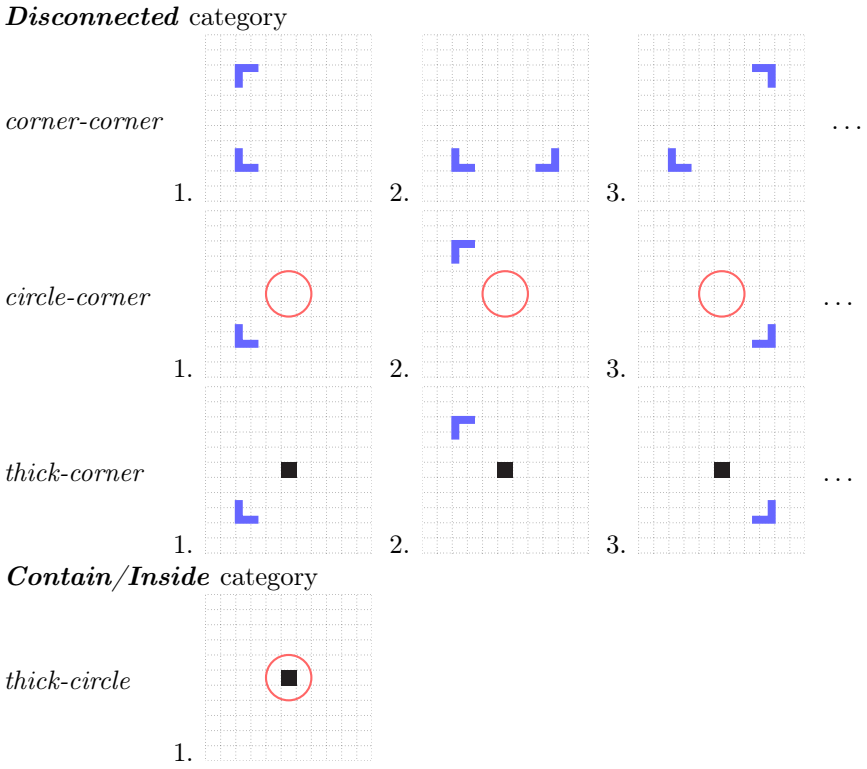
The topological configuration i.e., Topology$(\wp_1, \wp_2)$ is a vector in this space in which components equal 0 if the corresponding intersection is empty, and 1 otherwise.

For a symbol, let us remind that we have $\sum_{\mathbb{T}} = \{\mathbb{T}_t\}$ for $1 \leq t \leq |\mathbb{T}_t|$ vocabulary type set. For any t vocabulary type, there are n number of visual primitives $\{\wp_i\}_{i=1,\ldots,n}$. Therefore, the number of

- possible combinations within its own type (intra-type) is, $n_t = \frac{n \times (n-1)}{2}$ and
- possible inter-type combinations is, $m_{t,t'} = n \times n'$.



A. Visual elementary parts and their types.



B. Visual primitives categorisation based on their topological configurations.

Figure 6.6: An example of vocabulary categorisation. Item **A** shows the extracted vocabulary type sets. In item **B**, all possible number of combinations of visual primitives are shown. These are grouped into two topological categories: *disconnected* and *contain/inside*.

Altogether, the total number of possible combinations is,

$$\sum_{t} n_t + \sum_{t,t'} m_{t,t'}.$$

In Table 6.1, we realise the vocabulary categorisation. For better understanding, we separate intra-type and inter-type vocabulary combinations. In Table 6.1, each column represents the vocabulary type information and their possible pairwise combinations between the visual primitives in the framework of topological categorisation that goes row-wise. It is important to notice that all combinations do not exist. For example, *thick* vocabulary type is found to be happened only in *disconnected* category. But, *circle* type can be possibly found in all categories except *equal*. Therefore, in this illustration, one can remark the importance of vocabulary type information.

	Intra-type				Inter-type					
	$\mathbb{T}_1, \mathbb{T}_1$	$\mathbb{T}_2, \mathbb{T}_2$	$\mathbb{T}_3, \mathbb{T}_3$	$\mathbb{T}_4, \mathbb{T}_4$	$\mathbb{T}_1, \mathbb{T}_2$	$\mathbb{T}_1, \mathbb{T}_3$	$\mathbb{T}_1, \mathbb{T}_4$	$\mathbb{T}_2, \mathbb{T}_3$	$\mathbb{T}_2, \mathbb{T}_4$	$\mathbb{T}_3, \mathbb{T}_4$
1. *Disconnected*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. *Externally connected*	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗
3. *Overlap*	✗	✓	✓	✗	✓	✗	✗	✓	✗	✗
4. *Cover/Covered by*	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗
5. *Contain/Inside*	✗	✓	✗	✗	✓	✗	✗	✓	✓	✗
6. *Equal*	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Index

vocabulary type set: $\mathbb{T}_1$ = *thick*, $\mathbb{T}_2$ = *circle*, $\mathbb{T}_3$ = *corner* and $\mathbb{T}_4$ = *extremity*

categorisation: ✓ = YES and ✗ = NO

Table 6.1: Possible number of combinations that exist for all pairs of visual primitives. We separate intra-type and inter-type combinations for better understanding. The illustration follows Figure 6.5.

In the following, we will discuss how we compute directional relations between individual visual primitives.

6.3.2 Spatial Reasoning

In the previous section, we have categorised visual primitives using their topological configurations including vocabulary type information. Based on this, a pair of visual primitives can be expressed as,

$$\{(\wp_i^t, \wp_j^{t'}), \text{Topology}(\wp_i^t, \wp_j^{t'})\},$$

where $\wp_i^t$ is any ith visual part of any vocabulary type t in the set $\sum \mathbb{T}$. This means that the first part represents the pair of visual primitives with their vocabulary type information while the second part provides their topological configuration. Both represent categorisation of the pair.

Topological relations based categorisation of visual primitives do not provide precise information of spatial relations. For example, *disconnected* pairs of studied visual primitives can have several different orientations. Figure 6.6 shows an example where several different pairs of visual primitives are categorised in *disconnected* topological space. Therefore, directional relations $\Re$ between them needs to be computed accordingly. Now, we have

$$\{(\wp_i^t, \wp_j^{t'}), \text{Topology}(\wp_i^t, \wp_j^{t'}), \Re(\wp_i^t, \wp_j^{t'})\}.$$

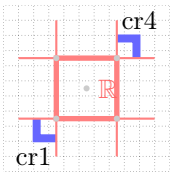
Since the first two parts refer to its categorisation, the whole expression can be reduced simply into $\Re_{cat}(\wp_i^t, \wp_j^{t'})$, where $\Re_{cat}(\centerdot)$ represents a relation in $\mathcal{C}_{cat}$ category. As an example, relation between two visual primitives *thick* and *corner* with *disconnected* topological information can be expressed as $\Re_{disconnected}(\wp_1^{thick}, \wp_2^{corner})$.

As mentioned in Section 6.3, several different ways to handle pairwise directional relations i.e., angle-based and projection-based models, are applied. To remind the models as attested in Chapter 4, let us follow an illustration for classical MBR model in Figure 6.7. According to what is developed in Chapter 4, for example, relations between the two visual primitives $\wp_1$ and $\wp_2$ can be formally expressed as,

$$\Re(\wp_1, \wp_2) = \{\mathcal{M}(\wp_1, \mathbb{R}), \mathcal{M}(\wp_2, \mathbb{R})\} := \text{projection model and}$$

$$\Re(\wp_1, \wp_2) = \{\mathcal{H}(\wp_1, \mathbb{R}_{p_c}), \mathcal{H}(\wp_2, \mathbb{R}_{p_c})\} := \text{radial line model}$$

where $\mathcal{M}$ is relational matrix with respect to reference point set $\mathbb{R}$ and $\mathcal{H}$ is relational histogram with respect to the reference point $\mathbb{R}_{p_c}$. In this chapter, instead of using them separately, we average them so that it respects symmetry.



$$\mathcal{M}(\text{cr1}, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathcal{M}(\text{cr4}, \mathbb{R}) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 6.7: A *disconnected* pair and directional relations using MBR model.

Therefore, following Figure 6.3 and Table 6.1, we compute relations for all possible combinations and store them accordingly. We also compute $\Re$ from all other relation models. Performances are then compared among them. The comparison will be provided and analysed in Section 6.5. Before that, we explain how recognition is handled, in the following section.

6.4 Symbol Recognition

As in Chapter 4, our symbol recognition is based on relation $\Re$ matching between the corresponding pairs of visual primitives. In every relations alignment, we follow vocabulary categorisation mentioned in Section 6.3.1. This means that matching happens between those pairs which share exactly the same vocabulary type information in the selected category. Thanks to vocabulary categorisation, matching does not need to go for all relations which are irrelevant. In what follows, we will first provide an idea of how relation matching happens. After that, the whole recognition process between any query symbol $\mathbf{S}^q$ and database symbol $\mathbf{S}^d$ will be explained.

6.4.1 Relation Matching

Let us take a combination of visual primitives $\wp_i^t$ and $\wp_j^{t'}$ as a query, respectively provides their vocabulary type information t and t'.

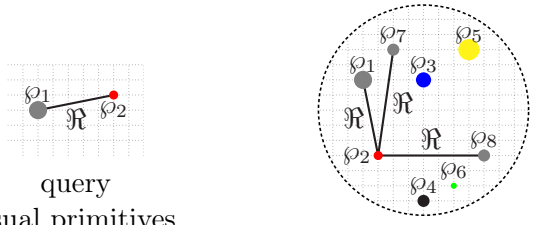
1. Select the category $\mathcal{C}_{cat}$ to which it belongs to i.e., the particular bag which contains similar topology as it does. As said before, to handle this, we use vocabulary type information. Now, query relation can be expressed as $\Re^q_{cat}(\wp_i^t, \wp_j^{t'})$.

2. Relation matching goes like this:

$$\delta\left(\Re^q_{cat}(\wp^t_i, \wp^{t'}_j), \Re^d_{cat}(\wp^t_i, \wp^{t'}_j)\right),$$

where $\delta(\star, *) = ||\star - *||_2$.

Figure 6.8 shows an example to illustrate how particular category is chosen in the database by using vocabulary type information of the query visual primitives. As a consequence, in this illustration, relations are used only from the selected visual primitives i.e., three relations are used for relation matching. For the best candidate, we take the minimum of the three.



query visual primitives.

visual primitives in database *disconnected* category.

Figure 6.8: Relation matching example in a particular category: *disconnected*. Each item represents a visual primitive and colour represents its vocabulary type. Since query visual primitives are *disconnected*, the category is chosen accordingly. Moreover, vocabulary type information are taken into account for matching so that it avoids irrelevant matching.

6.4.2 Symbol Recognition and Ranking

In the previous section, we have provided relation matching based on its topological and vocabulary type information and illustrated it with a visual example. But, we did not mention that visual primitives in particular category belong to which database symbol. In this section, the latter statement will be explained so that the whole symbol recognition can be handled.

Consider a query symbol $\mathbf{S}^q$ and database symbol $\mathbf{S}^d$ with t vocabulary type sets. For every pairwise combination of visual primitives that compose query symbol, relation matching occurs as

$$\text{Dist.}(\mathbf{S}^q, \mathbf{S}^d) \quad = \quad \sum_{cat} \min_{\mathcal{C}_{cat}} \delta\left(\Re^q_{cat}(\wp^t_i, \wp^{t'}_j), \Re^d_{cat}(\wp^t_i, \wp^{t'}_j)\right) \text{ for all } i \text{ and } j.$$

This means that all corresponding relation matching is summed up only when they are from same database symbol.

In a similar manner, we compute distances between all database symbols with respect to a query symbol and we follow the similar ranking strategy mentioned in Section 4.5.2 of Chapter 4.

6.5 Experiments

We perform a series of experimental tests and confront our proposed method with the same description on those presented in the previous chapters. In order to test our BOFs approach, we use the same set of spatial relation models used in the previous contributions.

1. Basic spatial relation models like
   - cone-shaped [Miyajima and Ralescu, 1994],
   - angle histogram [Wang and Keller, 1999] and
   - minimum boundary rectangle (MBR) [Papadias and Theodoridis, 1997].
2. Our model
   - projection model and
   - radial line model.

   The main difference with respect to Chapter 4 is that the way we apply spatial relations. In Chapter 4, relations are used between grouped vocabulary types. In this method, it goes for all individual visual primitives.

Once an optimal model is chosen, we still can compare it with our previous work – ARG based description in Chapter 4 as well as a major set of representative global shape descriptors validated in the previous chapters. In order to avoid redundancy, we will just do confront in the text (no visual results will be duplicated).

6.5.1 Datasets and Evaluation Metric

To validate the proposed spatial-bag-of-features method, we have tested over two different datasets: FRESH dataset [FRESH, 2007] and GREC dataset [GREC, 2003], as mentioned in the previous chapters. Similarly, we have followed the same evaluation metrics related to them: *retrieval efficiency* for FRESH dataset and *recognition rate* for GREC dataset.

6.5.2 Experimental Results

Figure 6.9 shows retrieval efficiency for the FRESH dataset over requested list: 1 to 10. In case of the GREC dataset, Table 6.2 provides recognition rates, for all models.

Compared to the ARG based description in Chapter 4 for FRESH dataset, spatial relation models possess similar behaviour but with a small improvement. In order to compare with the global signal-based shape descriptors, we present mainly $\mathcal{D}-$radon and sometimes GFD (the best of all, as validated before). The retrieval performance of our method supersedes by approximately 16% on average, for FRESH dataset. Whilst for GREC dataset, it lags behind them with approximate recognition rate of 3%. However, our method is still interesting. To express its advantages, we will provide further experiments in Section 6.6. Before that, we analyse the results first.

6.5.3 Experimental Results Analysis

In this section, we analyse the models based on experimental results provided in Figure 6.9 from FRESH dataset and Table 6.2 from GREC dataset. As explained in previous experiments, we consider recognition and retrieval performance including execution time.

In Figure 6.9, spatial relation models possess similar behaviour on retrieval performance (*cf.* Chapter 4). To the point, retrieval efficiency is found to be improved. Such an advancement is due to the fact that the current method uses individual visual part to compute relations – which shows its discriminative power. Among all, the radial line model outperforms all others. The

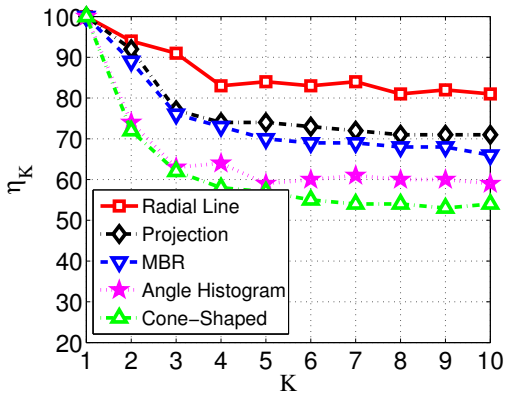


Figure 6.9: Average retrieval efficiency over requested list: 1 to 10 for FRESH dataset.

		ideal			scale		distort		
		set 1 5×1	set 2 20×1	set 3 50×1	set 1 5×5	set 2 20×5	set 1 5×5	set 2 15×5	Average
Spatial BOFs	1. Cone-shaped	80	90	88	76	72	84	78	81
	2. Angle Histogram	100	90	92	76	74	84	78	83
	3. MBR	100	95	96	84	84	88	82	89
	4. Projection Model	100	100	96	84	85	88	84	90
	5. Radial line Model	100	100	100	100	92	100	88	**97**

Table 6.2: Recognition rate in % for a few categories of GREC2003 dataset.

similar behaviour is found to be happened in case of the GREC dataset (Table 6.2), from all models. But the difference is due to lack of sufficient vocabulary extraction.

Regarding computational complexity issue, our method avoids all database symbols for matching. It is because of vocabulary type information as well as categorisation principle discussed in Section 6.3.2 and Section 6.4. As a consequence, execution time is reduced by more than approximate factor of 4 on average with respect to when no vocabulary categorisation is used. In case of global signal-based descriptors, we refer to Chapter 4 and 3 for comparison.

6.6 Advantages

Until now, recognition and retrieval performances based on spatial-bag-of-features have been explained with a series of experiments over two different datasets. Compared with our previous method, it does not provide notable difference and with state-of-the-art of shape descriptors, it is lagging behind. However, the method can provide more flexibility by providing user-friendly symbol retrieval as well as assisting symbol spotting problem. To illustrate the latter statement, we will again explicitly analyse the proposed method in two different perspectives:

1. whether the proposed method is able to retrieve significant known part of the symbol from the composite form, and

2. flexibility in choosing a query[8].

[8]Flexibility – it refers to how flexible user inputs visual primitives and their spatial organisation as a query, instead of taking a complete symbol.

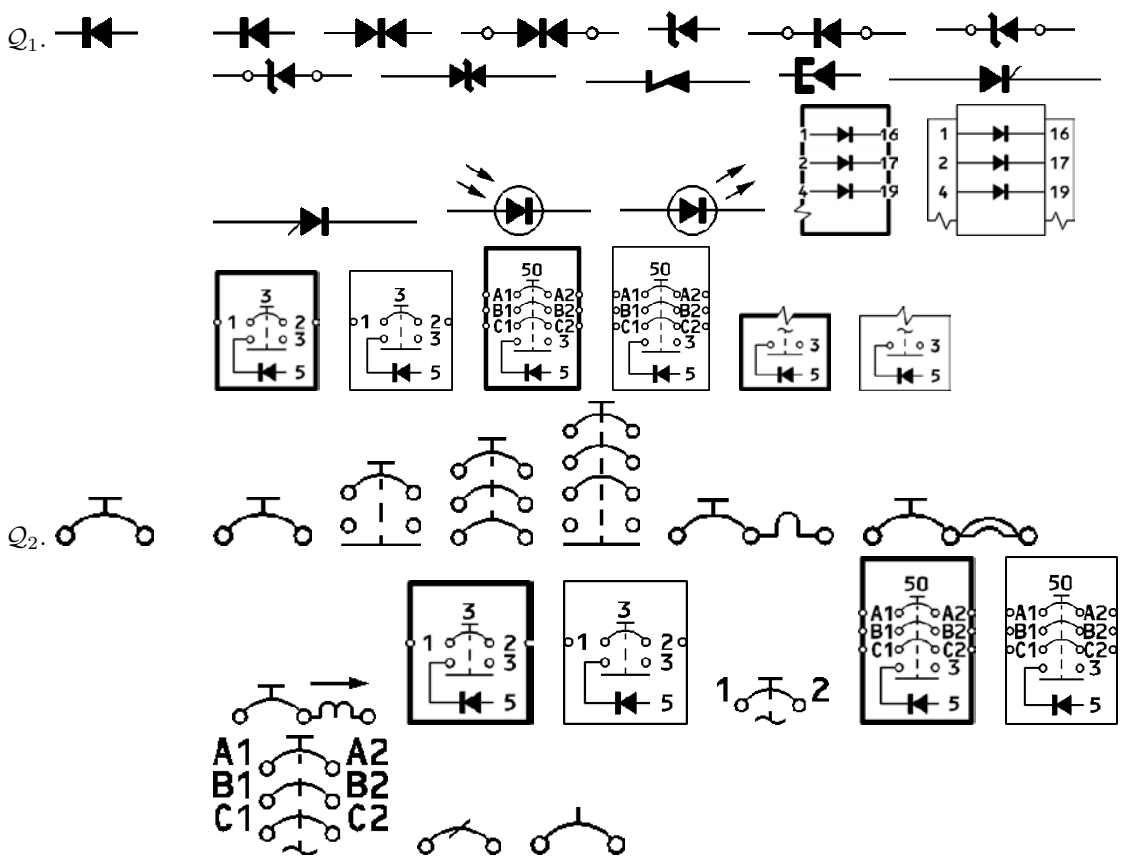
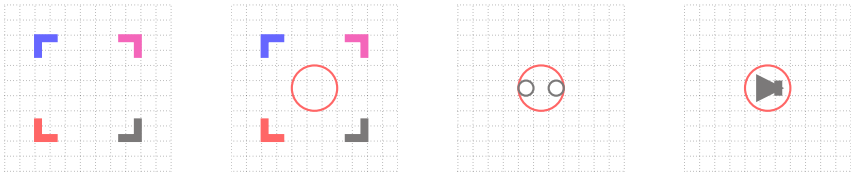


Figure 6.10: Symbol retrieval using isolated symbols as queries: $\mathcal{Q}_1$ and $\mathcal{Q}_2$.

Overall, we will project advantages by explicitly pointing out its appropriate use.

To illustrate *item 1*, we use isolated symbols as queries to see whether our method is able to retrieve symbols in the composite form. To handle this, FRESH dataset is used since GREC dataset is composed of only isolated symbols. Figure 6.10 shows a couple symbols as queries



(a) (b) (c) (d)

Figure 6.11: A few examples of a collection of visual primitives (including their spatial organisation) of only type or more than one, as queries.

and their corresponding retrieved symbols from the database. The illustration shows that known significant part of the symbols in the composite form, is also possible retrieved. This happens due to the fact that our relations alignment uses the vocabulary type as well as their category information. This means that not all visual primitives from the symbols are taken into account (*cf.* Section 6.4). As a consequence, it is possible to get distance between two symbols is zero, even though two candidates are different from each other, provided that query symbol

exists somewhere in the composite symbol. If not, retrieval symbols are ranked based on the decreasing order of similarity. Furthermore, since our matching uses topological guidance, metric directional relation does not really take over boolean ones. It is not however, reported in the experimental results. But at this point, we employ boolean relations to illustrate its performance of symbol retrieval in Figure 6.10. The difference between boolean and metric relations exists only when structural information is needed.

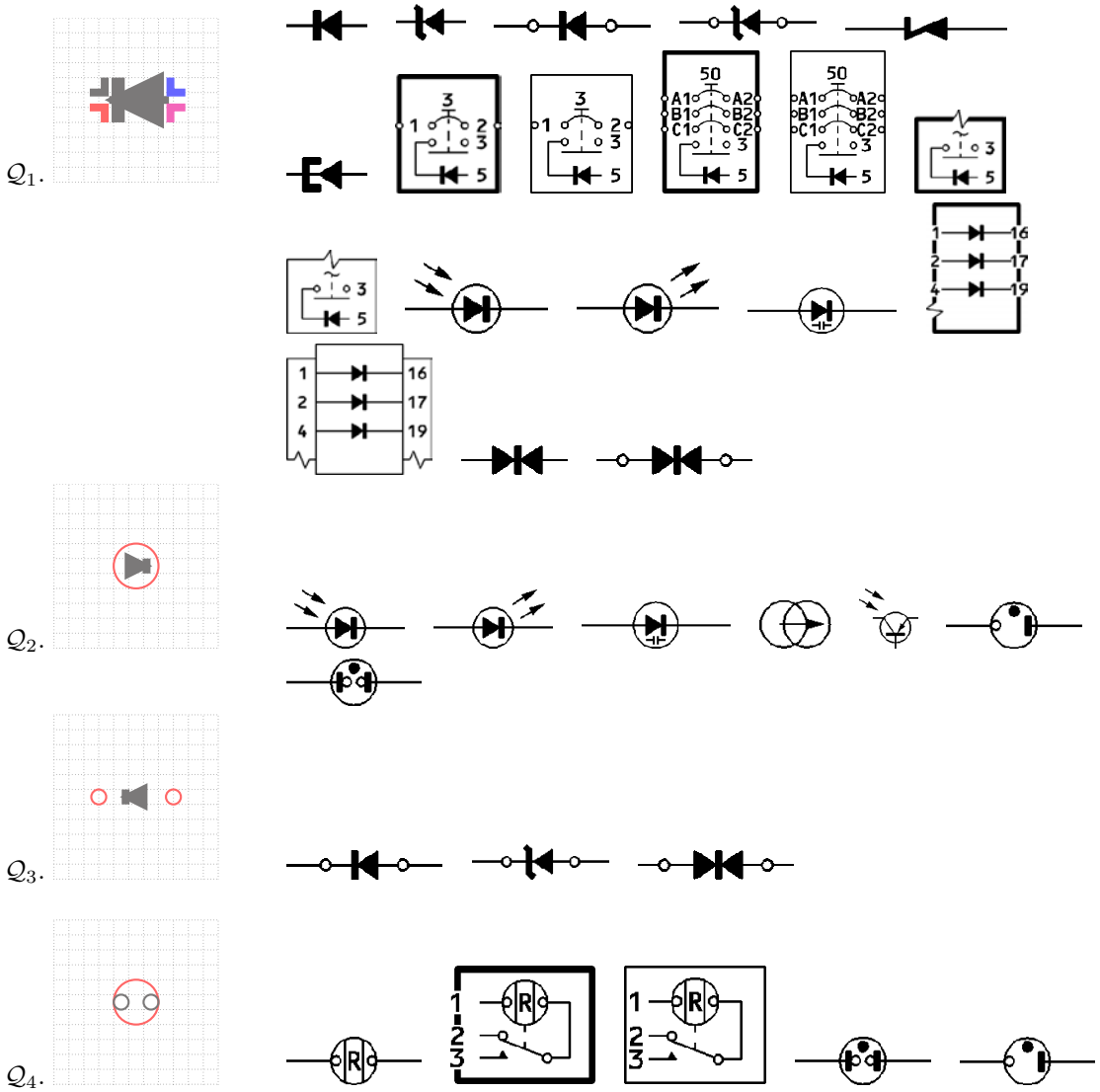


Figure 6.12: Symbol retrieval example for FRESH dataset. It is based on spatial organisation of a collection of visual primitives as queries.

Our method needs a set of visual primitives which collectively provide the description of symbols. Following examples as shown in Figure 6.10, we are inspired to use spatial organisation of visual primitives as a query, that can be used for the recognition of significant parts of the symbols. Such a query reflects the idea mentioned in *item 2*. For example, a set of four *corners* facing to each other represents a *rectangle*. Similarly, an architectural symbol having a *circle* inside a *rectangle* can be described as a set of two different vocabulary types: *circle* and *corner* where a *circle* is in the middle of four *corner*s. The key issue is that user can take any spatial

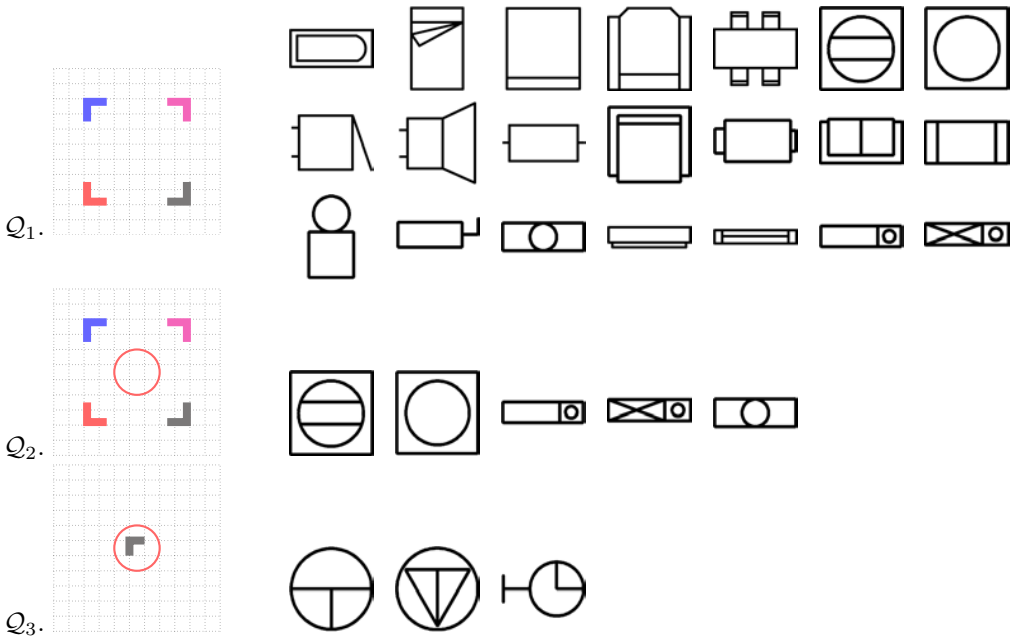


Figure 6.13: Symbol retrieval example for GREC dataset, using spatial organisation of a collection of visual primitives as queries.

organisation of the visual primitives which they think is important. Figure 6.11 shows a few examples of it. Using them as queries instead of using whole symbol, we can retrieve all related symbols from the database. In the following examples, both datasets are employed. Figure 6.12 and Figure 6.13 respectively provide a few examples of symbol retrieval.

1. **FRESH dataset** in Figure 6.12: Four different queries: $\{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_4\}$ are employed. Compared to $\mathcal{Q}_1$ and $\mathcal{Q}_2$, $\mathcal{Q}_3$ filters retrieval due to the presence of *circles*. Similarly, $\mathcal{Q}_4$ provides interesting retrieval.

2. **GREC dataset** in Figure 6.13: Three different queries: $\{\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3\}$ are employed. Query $\mathcal{Q}_1$ retrieves symbols which have *rectangle* within it, described by four *corner*s. While in query $\mathcal{Q}_2$, retrieval is made in addition to *circle* in it. Similarly, $\mathcal{Q}_3$ provides an interesting retrieval results.

In both tests, we do not mention the use of shape and size information of visual primitives, but we account for the spatial relations.

Overall, it is found that our method is appropriate to handle either symbol as a query or simply a collection of set of visual primitives together with their spatial information. Based on these tests, the key advantages of the method can be summarised as follows.

1. It can be treated as 'user-friendly' symbol retrieval application because user can select visual primitives and their spatial organisation to make a query. For instance,
   `'find symbol which has a thick component inside a circle'`.
   `'find symbol which has a rectangle containing a circle'`.
   On the contrary to FOL description of symbol in Appendix C that uses semantic concept for classification of selected or user-defined sub-set of symbols from the dataset, based on the common relations between the known visual primitives.

2. Symbol size does not make any difference unless there exists difference in topological configurations. Symbol retrieval using query $\mathcal{Q}_1$ and $\mathcal{Q}_2$ in Figure 6.13 shows that there is no effect in retrieval due to size difference in *rectangle* and a *circle*.

6.7 Conclusions

In this chapter, we present spatial-bag-of-features approach based on visual primitives that compose a symbol. Our method highlights topological relations guidance in addition to the their vocabulary type information. Such a topological guidance makes our method efficient in two different ways:

1. matching goes only to the relevant candidates i.e., it does not require all computed spatial relations; and

2. running time has been drastically reduced i.e., rapid search is possible.

We have validated our method with a series experimental tests over several different datasets related to graphical symbols. We can also forecast that our method can be extended to spotting problem.

End of Chapter

Keywords

Radon Features,
Dynamic Programming,
Shape Descriptors,
Visual Vocabulary,
Spatial Relations,
Spatial-Bag-Of-Features,
Graphics Recognition.

7 Summary

Foreword. In this chapter, we first highlight our contributions to Graphics Recognition. After that we address the continuation of the work. Future work will be primarily based on what we have established as well as the extension based on the shortcomings identified in the previous chapters. ■

7.1 Conclusions

This dissertation is mainly concerned with the use of spatial relations and shape analysis for graphics recognition. First, we have focussed on global signal-based descriptors. In this regard, we have introduced a shape descriptor based on dynamic programming and radon features, for graphics recognition. We have then concentrated on using spatial relations between the extracted visual primitives in several different ways to represent the symbols in addition to the use of shape descriptions where there exists shape variation. Furthermore, we have validated bag-of-features approach using spatial relations, aimed to recognise upto significant known parts (i.e., important graphical elements) of the symbols including isolated ones. In what follows, we conclude based on the major contributions one after another.

1. Symbol recognition usually means to recognise symbol in the isolated form. In this framework, we have introduced a method for symbol recognition which is based on dynamic programming for matching the Radon features.

   The aim of the method is to provide complete shape information of any patterns either simple (silhouette) or complex (internal contents is important), through simple projection at every angle, without normalising into a fixed length vector such that it can be globally used as a generic model. Pattern representation is simple and immediate. However, it offers high computational complexity due to dynamic programming employed in matching the Radon features. Therefore, in this method, we have explicitly tested the possible number of projections in order to select the optimal number of bins so that execution time can be reduced. Based on that, for silhouette shape (for only contour information), it is not the appropriate choice because of high computational matching complexity while it provides surprising difference where internal contents are important.

2. We have presented a method which purely uses spatial relations between the possible pairs of labelled vocabulary types such as *circle* and *corner*. These are then used as a basis for building an attributed relational graph (ARG) that fully describes the symbol. Due to a smart choice of using uniquely labelled vertices, we avoid the general NP-hard sub-graph matching problem. The spatial relation provides structural information of spatial objects while preserving spatial information.

   The proposed spatial relation model to generate relational signature, is simple and flexible. It has an ability to express spatial relations between any number of visual primitives. Besides, it can applied where no reference is provided. On the other hand, doing recognition purely based on spatial relations does not provide complete shape information of the visual primitives that may be of any complexity as well as size variation.

3. Spatial relations do not exploit complete shape information. Each vertex has a distinct vocabulary type containing different shape and size information. Therefore, in this contribution, we have integrated shape descriptions of the visual primitives and spatial relations together in two ways, keeping identical ARG based symbol description framework as before. Firstly, shape signatures are labelled as vertex signatures. Secondly, shape features are applied only to the vocabulary which shows significant shape variation.

   Overall, the method is based on the similarity between spatial organisation and shape features of the visual primitives that compose the symbol. Considering our applications, integrating shape descriptions does not provide a surprise difference since shape variations do not exist for all vocabulary types. In such a situation, one can integrate any type of shape descriptors only to the vocabulary where shape variations affect the recognition performance. Considering time complexity issue, such a flexible as well as efficient combination of statistical and structural approach, in fact provides interesting framework. This framework can be implemented in most of the image understanding problem since relational signatures provide precise spatial information and shape descriptions add more discrimination power.

4. Since relations exist between two vocabulary types, the method that uses only relations cannot be applied to the symbols when only one vocabulary type is extracted. Use of shape descriptors as mentioned in *item 3* recovers such a limitation. However, it does not provide surprising recognition performance difference and does not give any flexibility for retrieval applications.

   To handle this, we have introduced spatial-bag-of-features, inspired by computer vision problem. The method computes all possible relations between every individuals i.e., visual primitives instead of using a collection of similar types of vocabulary as a single vertex. The key characteristic of the method is to categorise bag-of-features of spatial relations according to the topological configurations they possess. As a consequence, relations alignment takes only with the candidates which share exactly same topological relations.

   The query can be a symbol or any number of visual primitives having their spatial organisation. The latter form of query can be applied to a 'user-friendly symbol retrieval' application. For example, `find symbol which has a thick component inside a circle`. Such a query in place of using a complete symbol, provides flexibility in accordance with the user's importance. Due to which, it can select relevant symbols of any kind, without a delay. It is important to notice that selecting a query that has a large number of spatial relations, takes more time even though relations matching go only to the related categories.

Therefore, when also considering time complexity issue, the method is in fact very much suited for selecting simple queries or such visual primitives so that it can retrieve known and significant parts within the symbol.

7.2 Future Perspectives

As mentioned in Appendix C, our aim is to lead to a more "semantic related recognition" process. We first express symbols by a number of visual primitives that may be of any complexity and connecting relationships. This representation is then used as an input to an inductive logic programming (ILP) solver, to automatically learn non-trivial descriptions of symbols, based on a formal description.

To describe first-order logic (FOL) description, the extracted vocabulary is combined with relative positioning (i.e., spatial relations). ILP is then used to extract "semantic" contexts from the domain knowledge. The contributions we have presented in this thesis: visual vocabulary, spatial relations and shape descriptions are going to be employed in the framework of semantic related recognition process. Our spatial relations provide the generic relationships in addition to the structural information since it yields natural relations instead of all-or-none relations. An example of it is expressed in the framework of spatial-bag-of-features approach in Chapter 6. Such a method provides an interesting concept to bootstrap the global aim presented in Appendix C. Furthermore, use of shape description (statistical) to represent visual primitives will provide precise information while deriving FOL description.

End of Chapter

Part III

Appendix

A
Evaluation Metric

Contents

A.1 Background 167
A.2 Evaluation Metric 168
A.2.1 Recognition 168
A.2.2 Retrieval 168
Fully labelled and balanced dataset 169
Imbalanced dataset 170
A.3 Conclusions 171

Foreword. In this appendix, we aim to present how performance of the algorithms will be evaluated. It is mainly focussed on difficulties associated with the provided dataset and ground-truths formation. Overall, we provide examples to illustrate where, which metric is appropriate. ■

A.1 Background

Matching score between two images conveys how similar or dissimilar to the chosen test image the database image is.

Based on matching scores, for every query image $\{\mathcal{I}^q\}_{q=1,\ldots,Q}$ over all database images $\{\mathcal{I}^d\}_{d=1,\ldots,D}$, distance vector can be expressed as,

$$Dist.(\mathcal{I}^q, \mathcal{I}^d) \quad = \quad \left[\mathcal{S}^{q,1}, \ldots, \mathcal{S}^{q,D}\right]$$

where matching score $\mathcal{S}$ is defined as the distance between two images: $\mathcal{S}^{q,d} = \delta(\mathcal{I}^q, \mathcal{I}^d) = \sum |\mathcal{I}_i^q - \mathcal{I}_i^d|$, for instance. Note that computing distance between two candidates is depend on what technique we use. Matching techniques are often induced by how images are described – which is not our concern. Our concern, as said before, is to identify the appropriate evaluation metric based on that matching score, following the varying nature of the datasets.

To generalise similarity that exists between the two given test candidates, similarity metric is usually used. To do so, scores are first normalised into $[0, 1]$ by,

$$\overline{\mathcal{S}} \;=\; \frac{\mathcal{S} - \mathcal{S}_{\min}}{\mathcal{S}_{\max} - \mathcal{S}_{\min}}.$$

Consequently, we formally express similarity as,

$$\text{Similarity}(\mathcal{I}^q, \mathcal{I}^d) = 1 - \overline{Dist.}(\mathcal{I}^q, \mathcal{I}^d) \begin{cases} 1 & \text{for the closest candidate} \\ 0 & \text{for the farthest candidate.} \end{cases}$$

Therefore, images are ranked according to the decreasing order of similarity in $[1, 0]$.

Example Consider a test image $\mathcal{I}^q$ that has been matched with $\{\mathcal{I}^d\}_{d=1,\ldots,10}$ database images, distance vector is,

$$\begin{aligned} Dist.(\centerdot) &= [10\ 8\ 2\ 1\ 3\ 4\ 7\ 8\ 9\ 5]\,, \\ \text{then } \mathcal{S}_{\min} &= 1 \text{ and } \mathcal{S}_{\max} = 10, \\ \overline{Dist.}(\centerdot) &= \left[1\ \frac{7}{9}\ \frac{1}{9}\ 0\ \frac{2}{9}\ \frac{3}{9}\ \frac{6}{9}\ \frac{7}{9}\ \frac{8}{9}\ \frac{4}{9}\right]. \\ \text{Now, Similarity}(\centerdot) &= \left[0\ \frac{2}{9}\ \frac{8}{9}\ 1\ \ \frac{7}{9}\ \frac{6}{9}\ \frac{3}{9}\ \frac{2}{9}\ \frac{1}{9}\ \frac{5}{9}\right]. \end{aligned}$$

A.2 Evaluation Metric

Based on Similarity(.) measurement, we consider the following measures. Those metrics are used based on what we aim for – either recognition or retrieval. Besides, varying evaluation metric is not only depend on the user's choice but also application and nature of the datasets. In what follows, we will discuss more about it, in parallel with different metrics.

A.2.1 Recognition

The straightforward nearest neighbour algorithm to classify the closest candidate [Beyer et al., 1999] is what we call recognition. In other words, the test image is supposed to be matched or recognised with the database image from which it produces the lowest matching score. Formally, we can express it as,

$$\text{Recognition}^d \;=\; \underset{d}{\text{argmin}} \left[\mathcal{S}^{q,1}, \ldots, \mathcal{S}^{q,D}\right].$$

But after normalisation, Similarity(.) = 1 for recognised candidate. Therefore, following the previous example, image in *index 3* is the matched candidate for the given query.

A.2.2 Retrieval

Proximity search space is increased to select the number of similar candidates for the requested short-list. There are several different ways to evaluate retrieval performance of the studied methods or approaches. At this point, we focus on to the nature of datasets. They are in general,

1. fully labelled and balanced dataset and

2. imbalanced dataset.

Datasets in the real-world, are not labelled and balanced. As a consequence, it is very difficult to compare state-of-the-art approaches in the same framework since evaluation metric can be varied. In the past, researchers used to do manually in order to make datasets balanced with fixed ground-truths such that comparison can be made. However, it may not convey real-world problem. Recently, much effort has been put to handle this without biasing [Smith, 2010].

Now, let us focus on evaluation metric for retrieval based on nature of the datasets.

Fully labelled and balanced dataset

It refers to identical number of ground-truths for all known classes. For example, shape datasets *kimia shape99* [Sebastian et al., 2001] consists of 9 classes, each one is having 11 samples. In such a case, we use mainly two different retrieval measures.

1. *Precision* and *Recall* – For every chosen query, traditional *precision* and *recall* measures account in the following way.

   Consider a query, precision can be formally expressed as,

   $$\text{precision} = \frac{n}{k}$$

   where n is the number of retrieved relevant candidates and k is the number of retrieval candidates. *Recall* now can be expressed as,

   $$\text{recall} = \frac{n}{N}$$

   where N is the number of relevant candidates i.e., ground-truth) for the given query. In other words, it can be looked at as the probability that a relevant candidate is retrieved by the query.

   It is trivial to achieve *recall* of 100% by returning all candidates in response to any query. Therefore *recall* alone is not enough but one needs to measure the number of non-relevant candidates also, for example by computing the *precision*.

   Another measure that combines *precision* and *recall* is the *harmonic mean*, the traditional $F-measure$ or balanced $F-score$:

   $$F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

   This is also known as the F_1 measure, because *recall* and *precision* are evenly weighted.

   It is a special case of the general F_β measure (for non-negative real values of β):

   $$F_\beta = (1 + \beta^2) \times \frac{\text{precision} \times \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

   Two other commonly used F measures are the F_2 measure, which weights *recall* twice as much as *precision*, and the $F_{0.5}$ measure, which weights *precision* twice as much as *recall*.

As a reminder, *precision* takes all retrieved images into account which can also be evaluated at a given cut-off rank, considering only the top-most results returned by the algorithm. This measure is called *precision* at K or *precision@k* since $k = 1, \ldots, K$ and K is the requested list i.e., cut-off ranking. Within the framework, *retrieval rate* has been increasingly used, which is sometimes known by the another measure *retrieval accuracy* [Sebastian et al., 2001, Rendek and Wendling, 2008, Beyer et al., 1999]. Formally, it is defined as the ratio of the number of candidates correctly classified according to the requested lists.

2. *Bull's eye score* – In *bull's eye test*, proximity search space is increased up to 2 times the number of relevant candidates in the dataset for each class. In such a given search space, every test sample is compared to all dataset candidates and the number of correctly classified ones is reported. Therefore, *bull's eye score* [Latecki et al., 2000, Belongie et al., 2002, Grigorescu and Petkov, 2003, Sebastian et al., 2003] can be expressed as,

$$\text{bull's eye score} = \frac{\sum n}{N}.$$

As before, n is the number relevant candidates and N, ground-truth for the given query.

Example Consider a query $Q1$ has 10 ground-truths i.e., $N = 10$ and therefore, we request for $K = 10$. The boolean result is as follow:

`result_Q1` = [1 1 0 1 0 1 1 1 0 0],

where 1 refers to correct match and 0, incorrect. Now, the *precision* and *recall* are computed as,

`precision_Q1` = [$\frac{1}{1}$ $\frac{2}{2}$ $\frac{2}{3}$ $\frac{3}{4}$ $\frac{3}{5}$ $\frac{4}{6}$ $\frac{5}{7}$ $\frac{6}{8}$ $\frac{6}{9}$ $\frac{6}{10}$],

`&  recall_Q1` = [$\frac{1}{10}$ $\frac{2}{10}$ $\frac{2}{10}$ $\frac{3}{10}$ $\frac{3}{10}$ $\frac{4}{10}$ $\frac{5}{10}$ $\frac{6}{10}$ $\frac{6}{10}$ $\frac{6}{10}$].

Once *precision* and *recall* are computed, $F-$score is straightforward. Now, *retrieval accuracy* or *retrieval rate* is the last element of the *precision*. In retrieval computation, requested list is usually taken as the value of ground-truth i.e., $K = N$. In this example, we observe that *recall* is limited to 60%. If we increase the *recall*, *precision* is going to be decreased. *Bull's eye score* is accounted. In *bull's eye test*, $K = 2 \times N$. Now the retrieval result is,

`result_Q1` = [1 1 0 1 0 1 1 1 0 0 1 0 0 1 0 0 1 1 0 0].

Therefore, *bull's eye score* = 100%.

Imbalanced dataset

It refers to different numbers of ground-truths for different classes. Therefore, in such a case, *retrieval efficiency* [Mehtre et al., 1995] is the better choice since *precision* and *recall* curve would be biased. For every test image, *retrieval efficiency* for a requested short-list of size K is expressed as,

$$\eta_K = \begin{cases} \frac{n}{N} & \text{if } N \leq K \\ \frac{n}{K} & \text{otherwise,} \end{cases}$$

where n is the number of returned relevant candidates and N the total number of relevant ones in the dataset. Note that η_K computes the traditional *recall* if $N \leq K$ and *precision* measure otherwise. Therefore, the output contains similar candidates either in terms of precision or recall,

for a given short-list. In other words, similar candidates will not be missed in the requested list. As said earlier, the key advantage of the metric is that average retrieval curve is not biased even with the different ground-truths for different queries. For more clarification, the following example will certainly help.

Example Consider a query $Q1$ has 6 ground-truths i.e., $N = 6$ and we request for $K = 10$. The boolean result is as follow:

$$\texttt{result\_Q1} = [1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0],$$

where 1 refers to correct match and 0, incorrect. Now, the precision and recall are computed as,

$$\texttt{precision\_Q1} = [\tfrac{1}{1}\ \ \tfrac{2}{2}\ \ \tfrac{2}{3}\ \ \tfrac{3}{4}\ \ \tfrac{3}{5}\ \ \tfrac{4}{6}\ \ \tfrac{5}{7}\ \ \tfrac{6}{8}\ \ \tfrac{6}{9}\ \ \tfrac{6}{10}],$$
$$\texttt{\&\ \ recall\_Q1} = [\tfrac{1}{6}\ \ \tfrac{2}{6}\ \ \tfrac{2}{6}\ \ \tfrac{3}{6}\ \ \tfrac{3}{6}\ \ \tfrac{4}{6}\uparrow\ \ \tfrac{5}{6}\ \ \tfrac{6}{6}\ \ \tfrac{6}{6}\ \ \tfrac{6}{6}].$$

Retrieval efficiency, for the same query can be computed as,

$$\texttt{retrival efficiency\_Q1} = [\tfrac{1}{1}\ \tfrac{2}{2}\ \tfrac{2}{3}\ \tfrac{3}{4}\ \tfrac{3}{5}\ \tfrac{4}{6}\ \tfrac{5}{6}\ \tfrac{6}{6}\ \tfrac{6}{6}\ \tfrac{6}{6}]$$

Precision has been biased from the index 6 (see arrow underneath). While in *retrieval efficiency*, recall is combined from where it is biased. Therefore, no chance of biasing is found when using *retrieval efficiency* as the evaluation metric.

A.3 Conclusions

Computing recognition is straightforward. It can be used for all types of datasets. However, retrieval measure is found to be tricky and therefore, it needs an appropriate choice.

Aforementioned two different examples point out the appropriate use of retrieval evaluation metric. *Precision* and *recall* measures are straightforward when dataset is labelled and balanced. It can be also applied in imbalanced dataset if and only if requested list is less than or equal to the minimum ground-truths of the query presented in the test. For instance, if query Q^q has the smallest number of relevant sample images in the dataset, then that is taken for maximum requested list to measure *precision* and *recall* without biasing. If not, *retrieval efficiency* is the best choice since it combines both of them.

End of Appendix

B

Affine Transformation Invariant Spatial Relation

Contents

B.1 Spatial Relations 173

B.1.1 Radial line Model – Working Principle 174

B.1.2 Affine Transformation Invariant properties 174

B.2 Experiments 175

B.2.1 Dataset 175

B.2.2 Results 177

B.2.3 Analysis 177

B.3 Conclusions 178

Foreword. As a reminder, in Chapter 4, we have discussed about that the proposed relational signature is translation and scale invariant. But, it does not satisfy rotation invariant property and we stick on it. It is mainly because of our dataset i.e., we do not have real image rotation.

In this appendix, we aim to present the extensibility of the radial line model i.e., in particular, we show how it can be rotation invariant. To validate the concept, we have tested over extended dataset. Besides, we have observed the behaviour with respect to the state-of-the-art approaches such as shape descriptors. ■

B.1 Spatial Relations

In this section, we first shortly recall radial line model and its working principle. Then we go for explaining affine transformation invariant properties.

B.1.1 Radial line Model – Working Principle

As mentioned in Chapter 4, Section 4.3.2, our relational signature is inspired by the concept of fuzzy relations [Freeman, 1975] that takes degree of truth which is more natural than using standard, all-or-none relations.

For a given unique reference point, we rotate radial line over a cycle to generate angular histograms. The principle idea is identical to what we have mentioned in Chapter 4. The only difference lies in the fact that we attempt to generate angular histogram which is invariant to rotation.

B.1.2 Affine Transformation Invariant properties

As said before, to avoid potential ambiguity for referencing one objects out of two in a pair, we compute a unique reference point in such a way that it satisfies rotation invariant property. We set up a unique reference point $\mathbb{R}_{p_c}$ computed from centroids $\mathbb{C}_{\mathbb{A}}$ and $\mathbb{C}_{\mathbb{B}}$ of given pairs $\mathbb{R}_p = \frac{\mathbb{C}_{\mathbb{A}}+\mathbb{C}_{\mathbb{B}}}{2}$.

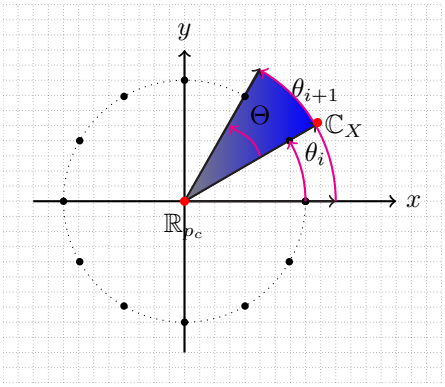


Figure B.1: Computing spatial relations using radial–line rotation. The rotation can be either clock-wise direction ↻ or ↺, however it must be fixed. In this illustration, we keep ↺. Radial–line rotation starts from the angle made by $\mathbb{R}_{p_c}$ and $\mathbb{C}_{\mathbb{X}}$. $\mathbb{C}_{\mathbb{X}}$ can be either $\mathbb{C}_{\mathbb{A}}$ or $\mathbb{C}_{\mathbb{B}}$.

Following Figure B.1, one can note the following points:

- Any angle made by $\mathbb{R}_p$ and $\mathbb{C}_{\mathbb{X}}$ ($\mathbb{X}$ can be $\mathbb{A}$ or $\mathbb{B}$) is re-projected on the horizontal axis such as to make the histogram rotation invariant.
- Translation does not affect at all, since it uses $\mathbb{C}_{\mathbb{X}}$.
- In a similar manner, scaling does not produce any difference in $\mathcal{H}$ as it is normalised in every sector made by θ_i and θ_{i+1}.

Illustrations Figure B.1.2 show an examples from real-world example (*cf.* Figure 4.16 in Chapter 4). The illustration conveys how relational signatures behave with image rotation. From the observations, we have found that image rotation does not really affect the radial line model. It shows however, infinitesimal change in amplitude due to digitisation problem.

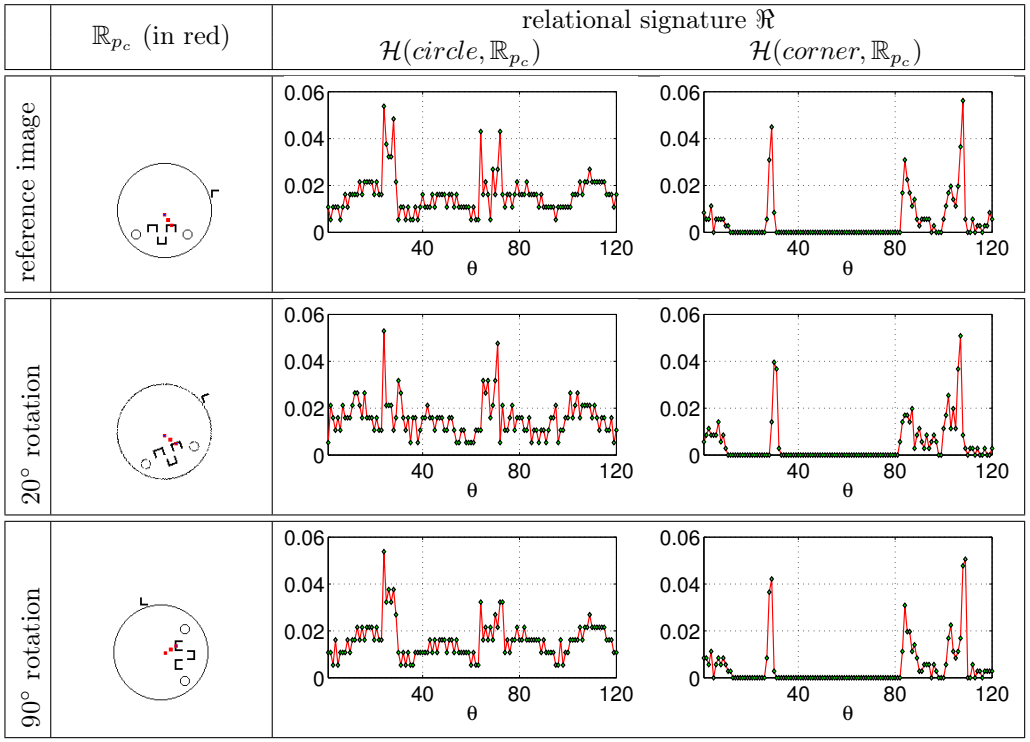


Figure B.2: Histograms at 3° resolution. Real-world example is taken from *symbol 1* in Figure 4.15. Vocabulary types: *circle* and *corner* are used, in order just to provide the rotation invariant property.

B.2 Experiments

B.2.1 Dataset

Out of 500 symbols in our original dataset, we have taken the sub-set of it which is composed of 180 symbols. For each symbol, we have 15 samples, consisting of rotation, translation, scale (uniform as well as stretching), noisy as well as their combinations.

1. *rotate* – image is rotated by 45°, 90° and 180°.

2. *translate* – translation has been made by increasing image size non-uniformly – for example, horizontal frame is increased by 20 times while keeping vertical frame the same and vice versa.

3. *scale* – for scaling, we applied both uniform and non-uniform concepts. Uniform scaling refers to keep equal weights for all pixels and non-uniform, stretching in one direction, for instance.

4. *distort* – it is completely different from vectorial distortion as found in [GREC, 2003]. It is something like adding noise to binary image. We add random white pixels by using sparsity matrix having same size with image. Those random white density of pixels eventually, distort parts of the image.

5. *noise* – it is based on random generation of pixels by using variance and mean.

Figure B.3 shows a few samples of it.

Index
rotate – 1 to 3; *translate* + *scale* – 4 to 6; *translate* + *distort* – 7 to 9;
translate + *noise* – 10 – 12; *translate* + *distort* + *rotate* – 13 to 15.

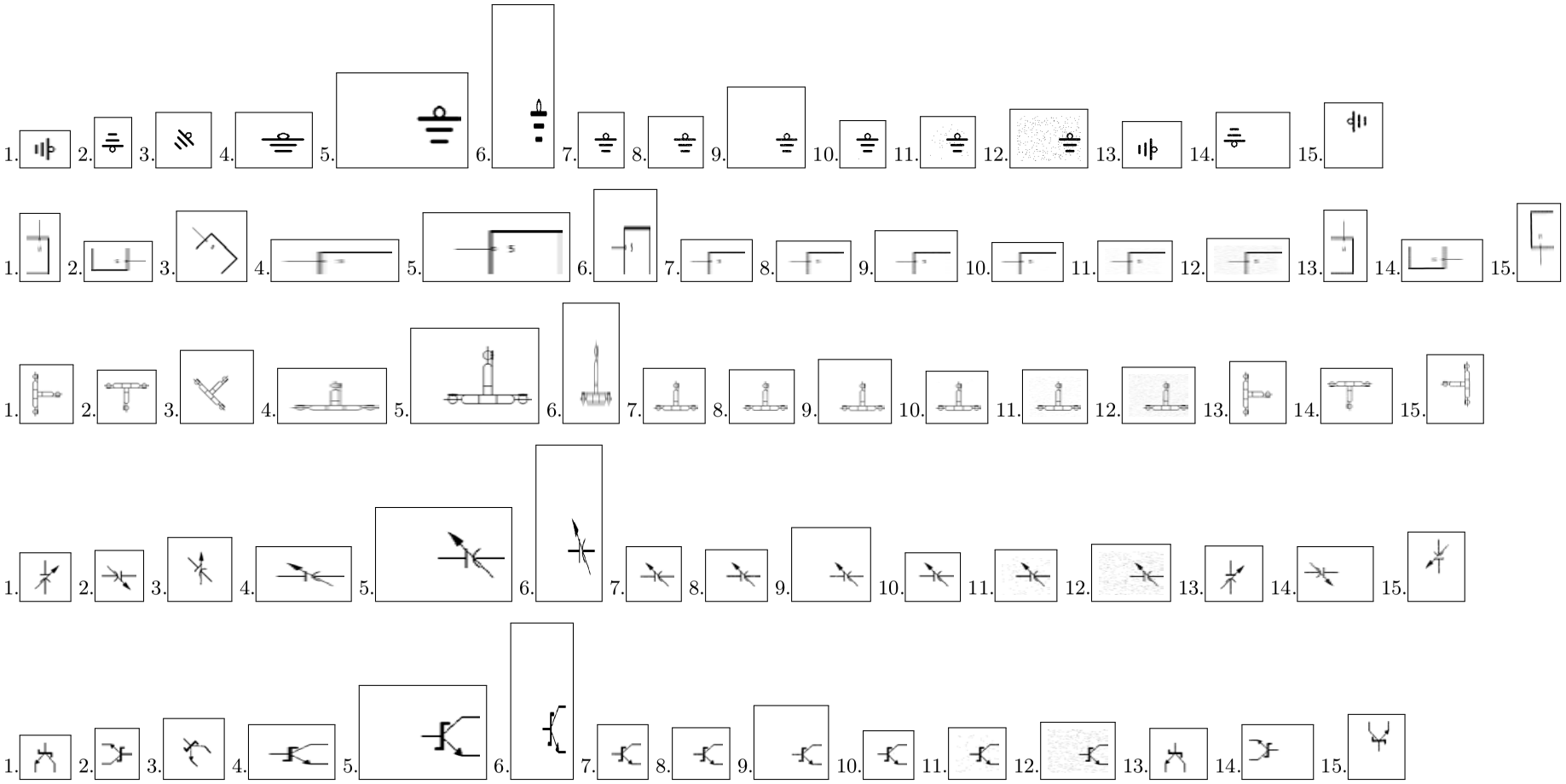


Figure B.3: Sample images for a few classes. Each class has 15 instances having noisy, rotation, scaling and translation as well as their combinations. Note that the size of the image is not equal to one–another.

B.2.2 Results

Our dataset is fully balanced and labelled. As a consequence, we use recognition rate – as an evaluation metric, for every sample of all classes (*cf.* Appendix A). In this test, every test sample is matched with all model symbols and the closest one is reported. The closest candidate corresponds to the labelled class. If the reported closed candidate is correct, then it is recognised otherwise, mis-recognised.

We aim to notice the behaviour of the method, in a few categories, taken from the combination of *rotate*, *translate*, *distort*, *noise* and *scale*. We not only apply our method but also a major set of global signal-based shape descriptors. They are

1. Zernike moments (ZM) [Kim and Kim, 2000],
2. generic Fourier descriptors (GFD) [Zhang and Lu, 2002],
3. shape context (SC) [Belongie et al., 2002],
4. $\mathcal{R}-$signature [Tabbone et al., 2006] and
5. $\mathcal{D}-\text{Radon}B$ ($B = 180$).

Table B.1 shows results for each category including their possible combinations.

	Rotate	Translate + Scale	Translate + Distort	Translate + Noise	Translate + Distort + Rotate	Average
$\mathcal{R}-$sign.	67	14	52	19	44	39
$\mathcal{D}-$Radon	77	19	82	42	77	59
ZM	41	11	13	05	33	21
SC	42	10	11	12	47	25
GFD	98	38	96	49	95	75
Spatial Relation	78	32	68	14	61	51

Table B.1: Recognition rate in % for all categories.

B.2.3 Analysis

Overall, GFD is found to be performed well in all sample image. Our method that uses spatial relations, provides however interesting results. ZM can not really discriminate symbol shapes since we have used 180 classes. The similar behaviour is found in SC. In contrast, $\mathcal{R}-$signature performs better. While, $\mathcal{D}-$Radon yields average results. In the following paragraph, we analyse method according to the samples used in the dataset.

1. Rotation samples at 45° affect our method since we are not able to generate corners.
2. Translation alone cannot degrade methods. But the effect has been seen when it is combined with noise. It is primarily because of the fact that centroid has been changed.
3. Scaling samples – non-uniform scaling affects all method. In our vocabulary, scaling produces unnecessary *thick* patterns.
4. Distortion does not really affect GFD and $\mathcal{D}-$Radon. But, our method degrades a bit since visual primitives are not present as in the model symbols.

5. Noisy samples highly degrade our method since there are innumerable dots over the image. In such a case, relations can be used with out *extremity* vocabulary.

B.3 Conclusions

We have provided that how relational signatures can be affine transformation invariant where we particularly focussed on *rotation* samples. In the experimental results, we have found that rotation samples affect only in such a situation where visual primitives are not present – absence of *corners* when rectangle is rotated by 45°, for instance. Besides, the test has also been extended to other samples. Our method is also compared with state-of-the-art of shape descriptors. Considering the dataset, our method that uses relational signatures provide interesting results over SC, ZM and $\mathcal{R}-$signature.

End of Appendix

C

Symbol Learning via Inductive Logic Programming based on Formal Description

Contents

C.1 Introduction 179
C.2 Inductive Logic Programming 180
C.2.1 Fundamentals of the Inductive Logic Programming 180
C.2.2 Global Behaviour – how does ILP work? 181
C.2.3 State-of-the-Art 182
C.3 Symbol Learning via Inductive Logic Programming based on Formal Description 184
C.3.1 Visual Vocabulary 185
C.3.2 Symbol Description 185
C.3.3 Symbol Recognition 187
C.4 Conclusions 191

Foreword. In this chapter, we use inductive logic programming (ILP) to automatically learn non trivial descriptions of symbols, based on a formal description.

The overall goal of the approach is to express graphic symbols by a number of visual primitives that may be of any complexity (i.e., not necessarily just lines or points) and connecting relationships that can be deduced from straightforward state-of-the-art image treatment and analysis tools. This representation is then used as an input to an ILP solver, aiming to deduce non obvious characteristics that may lead to a more semantic related recognition process. To validate the method, it is applied to FRESH dataset (as used in all chapters). ■

C.1 Introduction

The main and primary goal of image analysis is to eventually find the means of bridging the semantic gap between low level descriptions of an image and the concepts of what is presented

within it. The global state-of-the-art assumption is that this may be obtained through an adequately conceived interaction between shape descriptions and the comparison or distance measurements between them on the one hand, and classification or grouping techniques to associate these descriptions to higher level concepts on the other hand.

Trying to express visual information using 'natural' descriptions has actually been the original underpinning drive behind structural pattern analysis. Most often, this is done by first extracting low level visual cues that form the basic lexical data, and by proceeding by some grouping algorithm in order to express relationships or properties that are then translated into more and more complex 'vocabulary' that triggers higher level rules, eventually expressing terminal concepts [Mas Romeu et al., 2007].

Our approach is slightly different in the way that we do not try to construct a chain of syntactic rule triggering, but rather build our vocabulary on direct extraction of (more or less) complex structures in the images. This vocabulary is based on unpublished work [FRESH, 2007] that characterises symbols by a set of very robust, local structures. The vocabulary will be formally explained later. These structures need not necessarily be extracted by a structural or syntactic methods. In our case, for instance, we have developed specific and specialised detectors for circles [Lamiroy et al., 2007], oriented corners, loose endpoints and rectangles. Once a symbol is expressed as a set of elementary items, we use a reduced version of the force histogram based approach [Matsakis and Wendling, 1999, Matsakis et al., 2001, Matsakis et al., 2004] to position all items relatively one to another by using a quantitative assessment of directional spatial relationships (such as 'to the right of', 'above', 'south of' ...) between two items in a way that corresponds quite closely to natural language and perceptional coherent relative positioning. This allows us to express symbol descriptions with first-order logic (FOL) predicates, expressing their type as per the vocabulary and expressing the relative positioning one to another.

This framework gives us a straightforward way of describing the image that combines both expressiveness and very high flexibility. On the one hand, one can reduce or extend the size of the vocabulary in function of what robust descriptors are available. They may even be obtained using statistical or signal based extractions [Nguyen et al., 2008]. Furthermore the relations that express the relative positioning need not only be as simple as those represented, and can even include more quantitative information (e.g., [Bloch, 2005, Bennett and Agarwal, 2007, K.C. et al., 2009b]).

The remaining problem is how to explore what this new representation can offer in terms of recognition, classification of learning of concepts. We are going to do this in the following sections, by feeding these data to a ILP process. In Section C.2, we give an outline of inductive logic programming, its global behaviour including pertinent state-of-the-art. The state-of-the-art mainly concerns about the way how images are described. Then in Section C.3, we explore our problem: learning visual clues for symbol recognition using real–world industrial problem [FRESH, 2007]. We make conclusions of the work in Section C.4.

C.2 Inductive Logic Programming

C.2.1 Fundamentals of the Inductive Logic Programming

Inductive logic programming (ILP) [Plotkin, 1971, Muggleton et al., 1994, Nienhuys-Cheng and Wolf, 1997] is a research area formed at the intersection of machine learning (ML) and logic programming (LP). In other words, ILP combines automatic learning and first-order logic (FOL) programming. In the following, we first describe the FOL and then the use of ILP based on it.

Machine Learning A learner learns the behaviour from the examples (data) for specific task to improve overall performance [Mitchell, 1997]. It has been popularly used in the field of artificial intelligence (AI) [Michalski et al., 1986, Kodratoff and Michalski, 1990, Michalski and Tecuci, 1994]. Machine learning, like all subjects in AI, requires cross-disciplinary proficiency in several areas, such as probability theory, statistics, pattern recognition, cognitive science, data mining, adaptive control, computational neuroscience and theoretical computer science. Machine learning is primarily focussed to automatically recognise complex patterns and make intelligent decisions based on data – however, the difficulty lies in the fact that the set of all possible behaviours given all possible inputs. In one word, human knowledge of the world is often expressed in the form of intuitive theories: systems of abstract concepts that organise, predict and explain the observations of the world [Tenenbaum, 2008] – which resembles overall idea of human and machine learning perspectives.

Machine Learning uses several different algorithms such as, decision tree learning (DTL), association rule learning (ARL), artificial neural network (ANN), genetic programming (GP), Bayesian network (BN), clustering, support vector machine (SVM), reinforcement learning (RL) and ILP. Applications for machine learning mainly include natural language processing (NLP), syntactic pattern recognition (SPR), search engines, medical diagnosis, bio-informatics, classifying DNA sequences, speech and handwriting recognition and object recognition in computer vision. In this work, we apply ILP to learn domain background knowledge for symbol recognition.

The core objective of a learner is to generalise from its experience [Bishop, 2006]. In other words, the training examples in particular are used for experiencing and the learner has to extract something more general such that it allows to produce useful answers in new cases. Note that the data can be seen as examples that illustrate *relations* between the observed *variables*. It has been popularly implemented in terms of FOL.

In the following, the way we present FOL such that ILP can be used it for recognition. FOL will be discussed first and then ILP.

First-Order Logic It is something like natural language which assumes the world is made of *objects*, with individual *identities* as well as *characteristics* that distinguish, and *relations* hold them [Russell and Norvig, 2010]. Note that some of these relations are *functional*. Therefore, it is rather a powerful representation and reasoning system. In FOL system, relations are applied to objects to build predicates such as,

example1: `married(Bill, Francoise)`
example2: `above(Bird, House)`, and
example3: `British(Francoise)`.

In *example1*, `married` refers to relation between terms or variables: `Bill` and `Francoise` i.e., `Bill married Francoise`. For *example2*, an object `Bird` is `above` another object `House` and in a similar manner, an object `Francoise` is a `British` citizen, expressed in *example3*.

The language of FOL is shortly provided in Table C.1. Based on it, a very simple semantic of FOL and an example on how to express, can be written as,

semantic of FOL: $(\forall\ \texttt{x}\ \texttt{Red(x)} \equiv \texttt{Red(Obj1)} \wedge \texttt{Red(Obj2)} \wedge \texttt{Red(Obj3)} \wedge \ldots)$
an example: $\forall\ \texttt{x}\ (\texttt{Bird(x)} \wedge \neg\ \texttt{Ostrich(x)}) \Rightarrow \texttt{Flies(x)}.$

C.2.2 Global Behaviour – how does ILP work?

Inductive logic programming requires three main sets of information, the automatic solving and deduction theory set aside:

Table C.1: Language of FOL – grammar [Russell and Norvig, 2010].

Sentence ::=	AtomicS | ComplexS
AtomicS ::=	True | False | RelationSymb(Term, ...) | Term = Term
ComplexS ::=	(Sentence) Sentence Connective Sentence | ¬Sentence | Quantifier Sentence
Term ::=	FunctionSymb(Term, ...) | ConstantSymb | Variable
Connective ::=	∧ | ∨ | ⇒ | ⇔
Quantifier ::=	∀ Variable | ∃ Variable
Variable ::=	a | b | ... | x | y | ...
ConstantSymb ::=	A | B | ... | John | 0 | 1 | ... | π |...
FunctionSymb ::=	F | G | ... | Cosine | Height | FatherOf | + | ...
RelationSymb ::=	P | Q | ... | Red | Brother | Apple | > |...

1. a set of known vocabulary, rules, axioms or predicates, describing the domain knowledge base $\mathcal{K}$;
2. a set of positive examples $\mathcal{E}^+$ the system is supposed to describe or characterise with the set of predicates of $\mathcal{K}$;
3. a set of negative examples $\mathcal{E}^-$ that should be excluded from the deducted description or characterisation.

Given these data, the ILP solver is then able to find the set of properties $\mathcal{P}$, expressed with the predicates and terminal vocabulary of $\mathcal{K}$ such that the largest possible subset of $\mathcal{E}^+$ verifies $\mathcal{P}$, and such that the largest possible subset of $\mathcal{E}^-$ does not verify $\mathcal{P}$. For more clear understanding, ILP develops predicate descriptions from examples and background knowledge. The examples, background knowledge and final descriptions are all described as logic programs. The complete ILP scheme can be expressed as shown in Figure C.1.

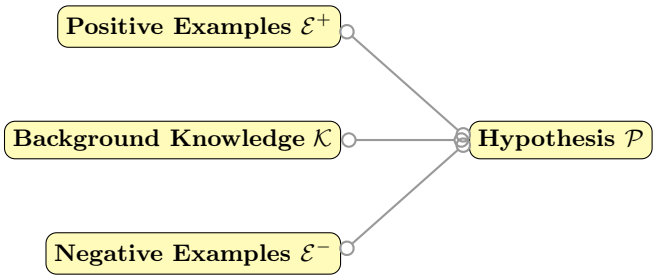


Figure C.1: ILP scheme.

Overall, the theory of ILP is based on proof theory and model theory for the first order predicate calculus. Inductive hypothesis formation is characterised by techniques including *inverse resolution*, *relative least general generalisations*, *inverse implication*, as well as *inverse entailment*. These are clearly mentioned in Oxford University Computing Lab[9].

C.2.3 State-of-the-Art

ILP has been already been successfully used in many areas. However, this section is mainly related to document analysis and recognition – any type of symbols including handwritten characters. More specifically, we review the way how images and or texts are described.

[9] http://www.comlab.ox.ac.uk/activities/machinelearning/ilp_theory.html

Structural Approach on Character Recognition Use of ILP has been illustrated in document and image analysis structures [Amin et al., 1996, Amin, 2003, Ceci et al., 2007]. The principle aim is to learn characters based on structural description and recognition has been made via ILP. To handle this, primitives as well as types of relations are pre-defined. For instance, domains can be `character` and `stroke`, relations can be `part_of`, `horizontal`, `east`, ... Now the description of character `A` can be expressed as,

`character(A)`	`A` is a `character`
`part_of(B,A)`	`B` is a part of `A`
`stroke(B)`	`B` is a `stroke`
`dot(C)`	`C` is a `dot`
`east(B,C)`	`B` is to the east of `C`
...	

Figure C.2: Character description based on idea presented in [Amin, 2003].

Semantic Extraction based on Text Relations For extraction of semantics from written text [Claveau and Sébillot, 2004, Horváth et al., 2009] is one of the contributions. In [Horváth et al., 2009], complex queries are handled based on a simple but essential step of relation extraction – can be regarded as a typical problem of learning logic programs. Each example is represented by a definite first-order Horn-clause in terms of dependency tree as shown in Figure C.3. In this illustration, we put constants two represent tree by relational structure i.e., a set of facts:

```
is(a1). Fraun(a2). optician(a3). a(a4). German(a5).
```

Therefore, it can be expressed using relational operator $\Re$. For example $\Re$`(a2,a1)` and $\Re$`(a2,a3)`. Based on aforementioned set of facts, let us take an example, `Fraun is a German Scientist`: `is_a(Fraun,Optician)` is encoded by `is_a(a2,a3)`.

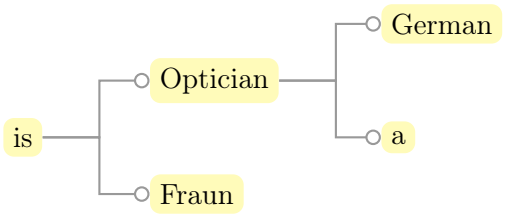


Figure C.3: Dependency tree for a sentence: 'Fraun is a German Optician' [Horváth et al., 2009].

Information Extraction for Unstructured Text ILP has also been increasingly used to identify unstructured text forms – a contemporary research issue in information extraction systems [Patel et al., 2010]. Named entities like names of persons, locations, and companies are the major matters, which are popularly used in message understanding (MU) as well as automatic content extraction (ACE) [Grishman and Sundheim, 1996, Song et al., 2009]. Named entity recognition (NER) using ILP has been incorporated to linguistic experts [Patel et al., 2010]. It does not mean that the ILP substitutes linguists however, it can excellently complement them by improving the ability to complete set of significant rules.
Based on the explanation of *entities* as well as in mentioned Figure C.4, background knowledge can be written as,

```
Person          Dr. Martin expressed his thoughts.
Location        He lives in Nancy.
Organisation    He watch people dancing in Martin-Theatre.
...
```

Figure C.4: Name of the entities.

```
b_entity(Person).   b_entity(Organisation).   b_entity(Location).   ...
b_word(Dr.).        b_word(Martin).           b_word(lives).       ...
b_tag(Verb).        b_tag(Noun).                                   ...
...
```

Therefore, learned rule for a single word `Dr.` can be modelled as,

$$\texttt{p\_entity(X,0,Person)} ::= \texttt{p\_word(X,-1,Dr.)}, \texttt{p\_tag(X,0,Noun)}.$$

Logical Network and Probabilistic Model via Semantic Distribution Besides, recently probabilistic logical model has been used [Fierens, 2010] based on [Raedt and Kersting, 2008] and [Getoor and Taskar, 2007]. An example of a first-order logical probability tree can be illustrated in Figure C.5. In this illustration, it uses `student(S)` and `course(C)` with simple probability FOL (predicate).

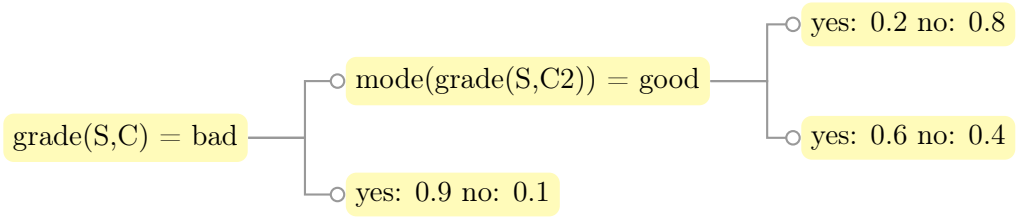


Figure C.5: First-order logical probability tree [Fierens, 2010].

Once images and or texts are modelled based on formal description, they are fed into ILP. ILP approach depends on reasoning.

C.3 Symbol Learning via Inductive Logic Programming based on Formal Description

In this section, electrical graphical symbols will be used to express FOL based on visual elementary parts. The dataset is taken from an electric wiring component database [FRESH, 2007][10].

Handling visual elementary parts from graphical symbol instead of using overall shape has several advantages. The advantages can be summarised in two key points:

1. Symbols can be localised within the composite form – the primary advantage and
2. The description provides a powerful representation of symbol using relationships between them.

Therefore, the detected visual elementary parts are formally organised by using all possible pairwise relations. To accomplish recognition, ILP solver will be used. Figure C.6 illustrates symbol recognition based on formal description via the use of ILP solver.

[10]The full experimental data file can be obtained on demand by contacting the authors.

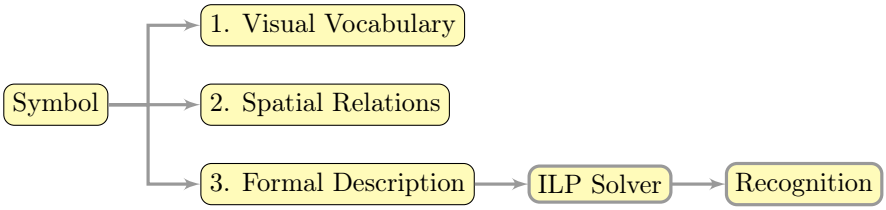


Figure C.6: An architecture for semantic related symbol recognition using inductive logic programming based on formal description.

As mentioned earlier, the approach allows for learning common properties within classes of symbols such as to express non-trivial knowledge of visual representation of more semantic concepts. This work currently concentrates on the *Image Analysis* part of the problem, and therefore uses ILP as a black-box framework. The ILP solver used in our experiments – Aleph – is freely available from the Oxford University Computing Lab[11]. Therefore, the overall idea is developed by using subsets of symbols from the dataset, to see how well ILP behaves.

In the following Section C.3.1, visual elementary parts are thoroughly discussed. Then graphical symbol is completely described using formal representation in Section C.3.2. ILP implementation for symbol recognition is provided in Section C.3.3.

C.3.1 Visual Vocabulary

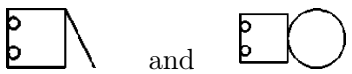
We define a set of well controlled visual elementary parts as a *vocabulary*. These are sometimes called by visual primitives throughout the thesis. They can be of any kind from any type of bag-of-features, related to what is visually pertinent in the application context under consideration. Our current vocabulary is related to electrical symbols, but can be easily extended to adapt to other domains. Currently, we have four different vocabulary types: *thick*, *circle*, *corner* and *extremity*. It has been clearly explained in Section 4.4.1, Chapter 4.

Figure C.7 shows a few sample images and the detected visual elementary parts keeping their spatial orientation. In this illustration, for symbol 141_2, there are three vocabulary types except *thick*. Symbol 180_3 is composed of all vocabulary types. In a similar manner, there are only two vocabulary types: *circle* and *corner*, in case of an symbol 226_2.

C.3.2 Symbol Description

The number of detected visual primitives along with their spatial organisation determine the overall configuration of the symbol. Based on the visual *vocabulary* mentioned in the earlier Section C.3.1, symbol description using FOL-vocabulary will be explained in the following.

Let us take two sample images:



symbol 225_2 symbol 226_2

for instance, to illustrate clearly how the complete symbol is described.

```
 %    symbol 225_2**********************************
type(prim_170,cornerne). type(prim_171,cornernw).
type(prim_172,cornerse). type(prim_173,extremity).
```

[11] http://web2.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.html.

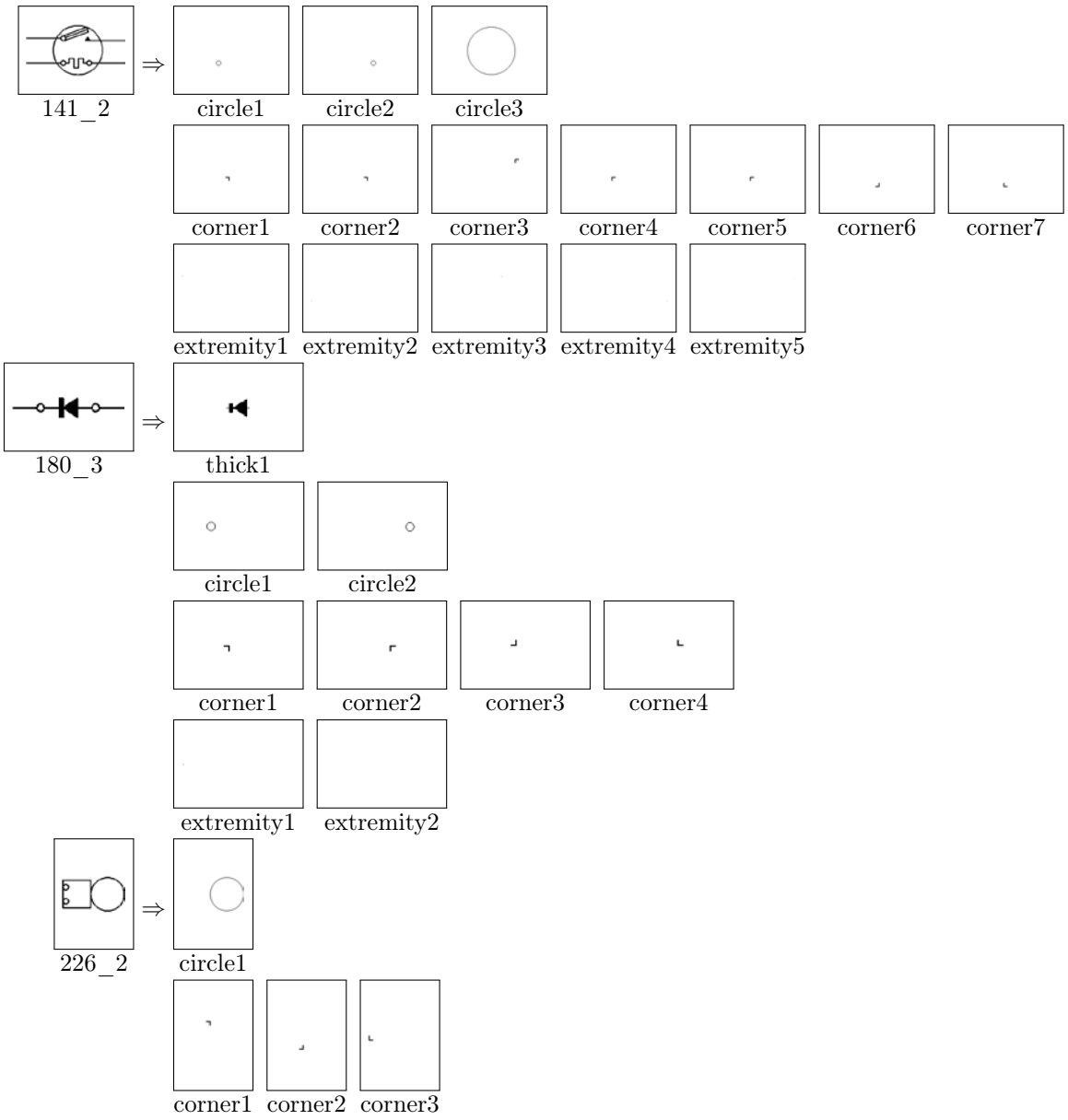


Figure C.7: Visual primitives or vocabulary types: *thick*, *circle*, *corner* and *extremity* for corresponding symbols.

```
has_element(img_225_2,prim_170).
has_element(img_225_2,prim_171).
has_element(img_225_2,prim_172).
has_element(img_225_2,prim_173).

nw(prim_170,prim_171). n(prim_170,prim_172).
nw(prim_170,prim_173). se(prim_171,prim_170).
ne(prim_171,prim_172). n(prim_171,prim_173).
s(prim_172,prim_170). sw(prim_172,prim_171).
w(prim_172,prim_173). se(prim_173,prim_170).
```

```
s(prim_173,prim_171). e(prim_173,prim_172).
%fin 225_2**********************************

%    symol 226_2**********************************
type(prim_174,circle).
type(prim_175,cornerne).
type(prim_176,cornerse).
type(prim_177,cornersw).

has_element(img_226_2,prim_174).
has_element(img_226_2,prim_175).
has_element(img_226_2,prim_176).
has_element(img_226_2,prim_177).

e(prim_174,prim_175). e(prim_174,prim_176).
inside(prim_174,prim_177). w(prim_175,prim_174).
n(prim_175,prim_176). nw(prim_175,prim_177).
w(prim_176,prim_174). s(prim_176,prim_175).
w(prim_176,prim_177). inside(prim_177,prim_174).
se(prim_177,prim_175). e(prim_177,prim_176).
%fin 226_2**********************************
```

Based on the aforementioned symbol description (take symbol 225_2), the first two lines assign the types of visual primitives what the symbol contains i.e.,

```
type(prim_XX, visual_primitive).
```

After that, the following four lines defines they are from the `img_name` by

```
has_element(img_name,prim_XX).
```

The last six lines provides the possible pairwise relationships between them: for instance,

```
nw(prim_XX,prim_YY)
```

i.e., `prim_XX` is to the *northwest of* `prim_YY`.

C.3.3 Symbol Recognition

Once we have symbol description via formal descriptions as described in Section C.3.2, it is applied to ILP solver. The ILP solver provides a rule which characterises a common properties within the classes of the symbols.

Overall, our approach allows for learning common properties from set of chosen symbols such as to express non-trivial knowledge of visual representations based on semantic concept. In the following, we start with introducing principle concept on the way how ILP solver works.

Basic Concepts

In order to show what kind of data we actually manipulate, we have selected symbols 225_2 and 226_2 from Figure C.8 as positive examples. All others as considered as counter examples.

The output of the ILP solver consists of a `[theory]` section, containing the rules that define the positive example set. For each rule of the theory, the solver gives matching statistics, indicating the precision of the rules (how many positive examples covered, and how many negative examples). For a perfect match, the theory section should consist of one single rule covering

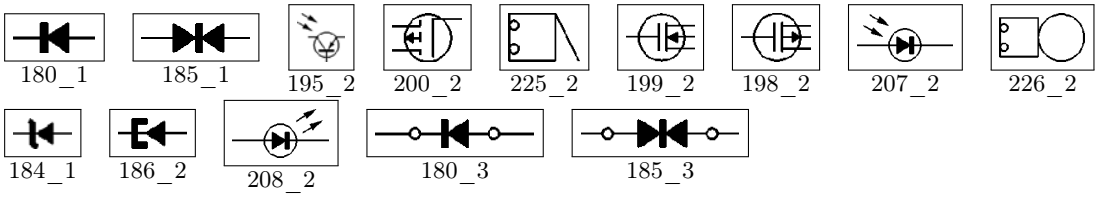


Figure C.8: First image set for ILP experimentation.

all positive examples and no negative examples. Further experiments will show that this is not always attainable. Sometimes the theory is composed of multiple rules, each of which covering a subset of the positive examples. Sometimes negative examples are covered by the theory as well. In our example, this gives:

```
[theory]
[Rule 1] [Pos cover = 2 Neg cover = 0]
symbol(A):-
    has_element(A,B), type(B,cornerne),
    has_element(A,C), n(B,C), type(C,cornerse).

[positive examples covered]
    symbol(img_225_2).
    symbol(img_226_2).
[negative examples covered]

test
[covered]
symbol(img_225_2):-
     has_element(img_225_2,prim_170),
     type(prim_170,cornerne),
     has_element(img_225_2,prim_172),
     n(prim_170,prim_172),type(prim_172,cornerse).
```

Following the `[theory]` section come two sections giving the examples covered by the theory: `[positive examples covered]` and `[negative examples covered]`. These two sections simply explicit the occurrences of examples covered by the rule-set.

The last part `[covered]` is simply an example of one of the covered occurrences, as to allow 'visual' verification.

The full interpretation of the output of our solver is that symbols 225_2 and 226_2 can be formally and completely distinguished from the other symbols by the fact that the dispose of two vertically aligned corners as shown in the following.

- ⌝ *north-east* corner (i.e., `cornerne`)
- ⌟ south-est corner (i.e., `cornerse`)

A visual inspection of Figure C.8 does not allow to find any counter examples. A more formal analysis of the image descriptions confirms this.

Global Behaviour

Let symbols {195_2, 198_2, 199_2, 200_2, 207_2, 208_2} from Figure C.8 be positive examples, representing the symbol of which the representation is to be learnt, and all others be counter examples. The ILP solver gives the following result:

```
symbol(A):-
   has_element(A,B), type(B,circle),
   has_element(A,C), inside(C,B),
   type(C,cornernw).
```

This experiment, translated into natural language, means that the chosen examples all have circles containing a *north-west* corner element ⌜. With this rule, all positive examples are perfectly classified with respect to the negative ones.

However, when selecting another set of symbols, like {180_1, 180_3, 184_1, 185_1, 185_3, 186_2}, the system is not able to reduce the set to a single predicate:

```
[Rule 1] [Pos cover = 1 Neg cover = 0]
symbol(img_180_1).

[Rule 2] [Pos cover = 2 Neg cover = 0]
symbol(A):-
   has_element(A,B), type(B,circle),
   has_element(A,C), e(B,C),
   type(C,cornernw).

[Rule 3] [Pos cover = 3 Neg cover = 0]
symbol(A):-
   has_element(A,B), type(B,blackthick),
   has_element(A,C), type(C,cornersw),
   has_element(A,D), ne(C,D).
```

Symbol 180_1 is not covered by the predicates, and the remaining, positive examples, are split up into two distinct sub-classes, each of them covered by a separate rule. The first one (i.e., `[Rule 2]`), describing the symbols as containing a circle ○ and a corner ⌜, placed to the east of it. The second one is far more interesting, and gives an outstanding reason of using ILP solving. `[Rule 3]` indeed mentions that the corresponding symbols contain a *thick* component ⊣ and corner ⌞, but, more interestingly, that it contains a third, *unspecified* – *any* –, primitive at a *north-east* position of the corner. The fact that it is possible to express these very generic relationships (regardless of the underlying shape!) is something numeric learning or classification techniques [Malon et al., 2008, Kuncheva, 2005] cannot achieve.

Choosing a Side

Let us now consider symbols {179_006_2, 179_007_2, 179_008_2, 179_009_2, 179_010_2, 179_011_2} from Figure C.9 as positive examples, while remaining symbols are the counter examples. The generated theory is hardly able to find any common rules between them. Only two positive examples are covered within one rule, while the others are generated independently. This, actually, is quite normal, since vocabulary used to represent the symbols (*circle*, *thick* components, *corner* ...) is very badly suited for distinguishing between them. The main point is, however, that, if we invert the positive and negative examples (i.e., we try to learn a common characterisation of the set of counter examples: {193_2, 194_2, 195_2, 196_2, 197_2, 198_2, 199_2, 200_2, 201_1}, the ILP solver generates one single rule.

```
symbol(A):-
   has_element(A,B), type(B,cornernw).
```

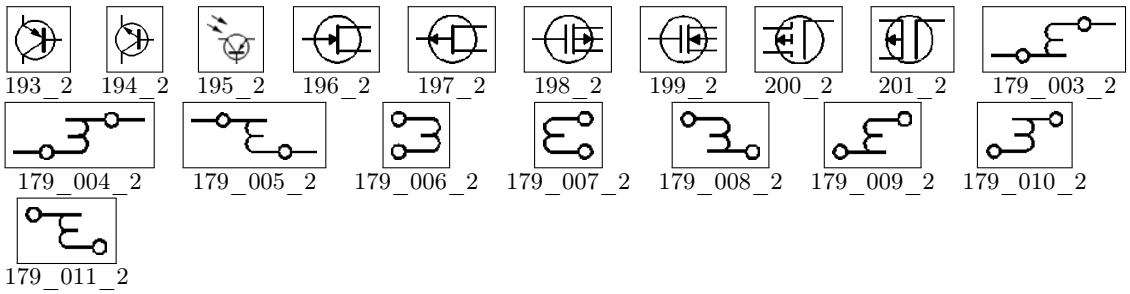


Figure C.9: Second image set for ILP experimentation.

The developed rule covers all the examples in a single predicate and only a primitive is needed to describe the set. It is thus very important to try and characterise both positive and negative example sets.

Another point of interest with this experiment is that the generated rule might seem 'visually' weird with respect to human interpretation. Indeed, one would naturally describe the example set as '*circles containing stuff*'. Although this rule is perfectly acceptable in our framework, giving something like:

```
symbol(A):- has_element(A,B), type(B,circle),
  has_element(A,C), inside(B,C).
```

It is, unfortunately, more complex than the rule the automatic solver found. This is due to the fact that the ILP solver works in a 'closed world' of predicates, vocabulary and examples and cannot infer that a given solution might be 'more generic' than another with respect to human interpretation standards.

Learning Set Induced Limits

In the previous sections we addressed the question on how the predefined vocabulary affects the learning process. In this section we address the influence of the learning samples. Let us consider the case of a specific semantic concept: a diode. The positive and negative learning examples are taken from Figure C.10.

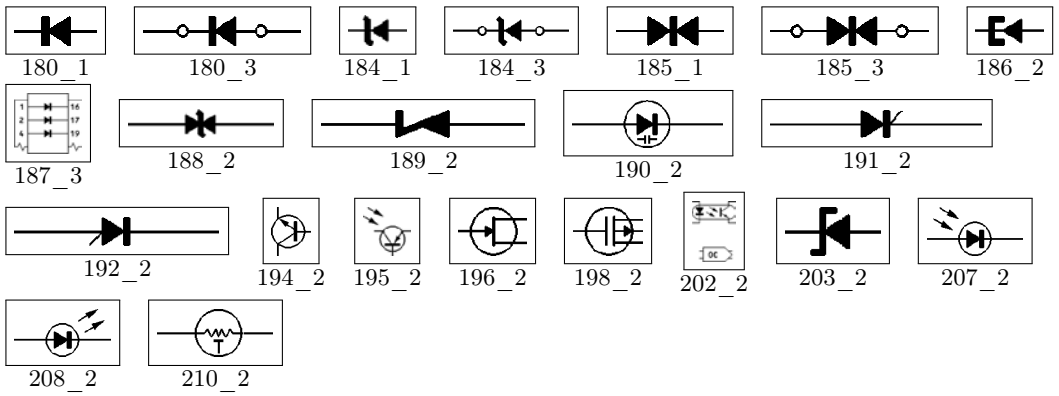


Figure C.10: Third image set for ILP experimentation.

First, let the set of positive samples be {180_1, 180_3, 184_1, 185_1, 185_3,186_2}. The negative ones are {194_2, 198_2, 210_2}. The produced theory is:

```
symbol(A):-
   has_element(A,B), type(B,blackthick).
```

This theory thus implies that every symbol which contains a *thick* object is a diode. This rule, of course, is too 'simple', due to the fact that the learning set was too limited (it is, however, quite correct in the context of the closed learning set world). Let's add a negative example which contains black thick objects: the negative sample set becoming {194_2, 198_2, 210_2, 195_2}.

```
symbol(A):-
   has_element(A,B), type(B,blackthick),
   has_element(A,C), type(C,cornerse).
```

As expected, the rule evolved to take into account the new counter-examples, but still only relies on the presence of two kinds of primitive: 'blackthick' and 'cornerse'. We can now add negative examples which contain these primitives. The set of negative examples become {194_2, 198_2, 210_2, 195_2, 196_2, 202_2} and the theory is now:

```
symbol(A):-
   has_element(A,B), type(B,blackthick),
   has_element(A,C), type(C,cornerse),
   has_element(A,D), e(D,C), inside(D,B).
```

At this stage, the spatial relations become the important criterion. The interpretation of the rule is:

```
'having a south-east corner which is on the right of another object
 which contains the thick object'.
```

This is not exactly the rule that we might be expecting, but we have to be aware that it is a rule based on only 6 negative examples and that the primitive detection is not perfect. By extending to the full set of 19 positive examples and the 6 negative examples previously used, the system obtains:

```
symbol(A):-
   has_element(A,B), type(B,blackthick),
   has_element(A,C), type(C,cornernw),
   has_element(A,D), e(C,D), type(D,cornerse).
```

What is interesting to note here, is that, compared to statistical learning models, the system adapts the complexity of the classification with respect to the learning data, without need for any parametrization of any sorts. On the other side, however, it is also quite straightforward to see that, if the learning set is contradictory with respect to the available vocabulary, it cannot deduce any classification rule.

C.4 Conclusions

This chapter presents the first step towards another approach of symbol recognition and representation, by combining robust elementary form detectors that compose a predefined, but extensible vocabulary. This vocabulary is combined with relative positioning in order to obtain a FOL based description of the symbols, on which ILP can be used to extract "semantic" contexts or concepts. The interesting part of this is that the description of the symbols can now be easily mixed with other, more context related information. Particularly, approach can be summarised in two major points.

1. The main advantage of this approach is, that information need not necessarily be visually represented (for example, from surrounding text), and it thus opens a new scope of possible combined text/image concept characterisation and learning. It is even possible to expand the framework to generate symbols from the obtained descriptions (either for visual validation of classification results or for automatic illustration generation), as shown in [Lamiroy et al., 2009].

2. Compared to statistical learning models, the system adapts the complexity of the classification with respect to the learning data, without need for any parametrization of any sorts. Besides, it is possible to express generic relationships (regardless of the underlying shape!) is something numeric learning or classification techniques [Malon et al., 2008, Kuncheva, 2005] cannot achieve. On the other side, however, it is also quite straightforward to see that, if the learning set is contradictory with respect to the available vocabulary, it cannot deduce any classification rule. The method, as it currently stands, is limited by the expressive power of the used vocabulary.

Further work is therefore consisting of extending the initial vocabulary, by introducing the notion of connexity, refining the inclusion predicate and using relative distance and size (close, far, large, small ...). These are all straightforward extensions that are readily available from an image analysis standpoint. Further, less straightforward, work will concern inclusion of more numerical or statistical form descriptions that might be able to better quantify differences between the shapes and will need to rely on Markov logic [Richardson and Domingos, 2006, Domingos et al., 2008] to handle the numerical part. More prospective work will be to connect this to formal concept analysis [Bernhard Ganter and (eds.), 2005] and Galois lattices to achieve unsupervised learning of visual concepts. One possible way to use spatial bag-of-features for recognition is discussed and validated in Chapter 6.

End of Appendix

Bibliography

[Ablameyko, 1997] Ablameyko, S. (1997). *Introduction to Interpretation of Graphic Images*. SPIE, Optical Engineering Press.

[Adam et al., 1999] Adam, S., marc Ogier, J., Cariou, C., Gardes, J., Mullot, R., and Lecourtier, Y. (1999). Combination of invariant pattern recognition primitives on technical documents. In *Graphics Recognition, Springer, Lecture Notes in Computer Science Series*, pages 238–245.

[Ah-Soon and Tombre, 2001] Ah-Soon, C. and Tombre, K. (2001). Architectural symbol recognition using a network of constraints. *Pattern Recognition Letters*, 22(2):231–248.

[Aksoy, 2009] Aksoy, S. (2009). Spatial relationship models for image information mining.

[Alginahi, 2010] Alginahi, Y. (2010). Preprocessing techniques in character recognition. *Character Recognition*.

[Amin, 2003] Amin, A. (2003). Recognition of hand-printed characters based on structural description and inductive logic programming. *Pattern Recognition Letters*, 24(16):3187–3196.

[Amin et al., 1996] Amin, A., Sammut, C., and Sum, K. C. (1996). Learning to recognize hand-printed Chinese characters using Inductive Logic Programming. *International Journal of Pattern Recognition and Artificial Intelligence*, 10(7):829–847.

[Attalla and Siy, 2005] Attalla, E. and Siy, P. (2005). Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching. *Pattern Recognition*, 38(12):2229 – 2241.

[Bailey and Srinath, 1996] Bailey, R. R. and Srinath, M. (1996). Orthogonal moment features for use with parametric and non-parametric classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):389–399.

[Bar and Ullman, 1993] Bar, M. and Ullman, S. (1993). Spatial context in recognition. *Perception*, 25:324–352.

[Barbu et al., 2006] Barbu, E., Raveaux, R., Locteau, H., Adam, S., Héroux, P., and Trupin, É. (2006). Graph classification using genetic algorithm and graph probing application to symbol recognition. In *Proceedings of the IAPR International Conference on Pattern Recognition*, pages 296–299.

[Barrat and Tabbone, 2010] Barrat, S. and Tabbone, S. (2010). A bayesian network for combining descriptors: application to symbol recognition. *International Journal on Document Analysis and Recognition*, 13(1):65–75.

[Bellman and Kalaba, 1959] Bellman, R. and Kalaba, R. (1959). On adaptive control processes. *Automatic Control*, 4(2):1–9.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522.

[Bennett and Agarwal, 2007] Bennett, B. and Agarwal, P. (2007). Semantic categories underlying the meaning of 'place'. In *Proceedings of International Conference on Spatial Information Theory*, volume 4746 of *Lecture Notes in Computer Science*. Springer-Verlag.

[Bernhard Ganter and (eds.), 2005] Bernhard Ganter, G. S. and (eds.), R. W., editors (2005). *Formal Concept Analysis: Foundations and Applications*. Number 3626. Springer-Verlag.

[Bernier and Landry, 2003] Bernier, T. and Landry, J.-A. (2003). A new method for representing and matching shapes of natural objects. *Pattern Recognition*, 36(8):1711–1723.

[Beyer et al., 1999] Beyer, K. S., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is "nearest neighbor" meaningful? In *Proceedings of International Conference on Database Theory*, pages 217–235.

[Bhattacharya and Chaudhuri, 2009] Bhattacharya, U. and Chaudhuri, B. B. (2009). Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):444–457.

[Bhowmik et al., 2006] Bhowmik, T. K., Parui, S. K., Bhattacharya, U., and Shaw, B. (2006). An hmm based recognition scheme for handwritten oriya numerals. In *Proceedings of International Conference on Information Technology*, pages 105–110, Washington, DC, USA. IEEE Computer Society.

[Biederman, 1972] Biederman, I. (1972). Perceiving real-world scenes. *Science*, 177(43):77–80.

[Bishop, 2006] Bishop, C. M. (2006). *pattern Recognition and Machine Learning*. Springer.

[Bloch, 1999] Bloch, I. (1999). Fuzzy relative position between objects in image processing: a morphological approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(7):657–664.

[Bloch, 2005] Bloch, I. (2005). Fuzzy spatial relationships for image processing and interpretation: a review. *Image and Vision Computing*, (23):99–110.

[Bodic et al., 2009] Bodic, P. L., Locteau, H., Adam, S., Héroux, P., Lecourtier, Y., and Knippel, A. (2009). Symbol detection using region adjacency graphs and integer linear programming. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1320–1324.

[Boulgouris and Chi, 2007] Boulgouris, N. V. and Chi, Z. X. (2007). Gait recognition using radon transform and linear discriminant analysis. *IEEE Image Processing*, 16(3):731–740.

[Bunke and Messmer, 1995] Bunke, H. and Messmer, B. T. (1995). Efficient attributed graph matching and its application to image analysis. In *CIAP*, pages 45–55, London, UK. Springer-Verlag.

[Bunke and Riesen, 2011] Bunke, H. and Riesen, K. (2011). Recent advances in graph-based pattern recognition with applications in document analysis. *Pattern Recognition*, 44(5):1057–1067.

[Bunke and Sanfeliu, 1990] Bunke, H. and Sanfeliu, A., editors (1990). *Syntactic and Structural Pattern Recognition*. World Scientific.

[Bunke and Shearer, 1998] Bunke, H. and Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259.

[Burge and Kropatsch, 1998] Burge, M. J. and Kropatsch, W. G. (1998). A minimal line property preserving representation of line images. In *Proceedings of IAPR Workshop on Structural and Syntactical Pattern Recognition*, pages 355–368.

[Cao et al., 2010] Cao, Y., Wang, C., Li, Z., Zhang, L., and Zhang, L. (2010). Spatial-bag-of-features. In *Computer Vision and Pattern Recognition*, pages 3352 –3359.

[Carr et al., 1999] Carr, D. B., Young, C. J., Aster, R. C., and Zhang, X. (1999). Cluster analysis for ctbt seismic event monitoring. In *a study prepared for the U.S. Department of Energy*.

[Cave and Kosslyn, 1993] Cave, C. B. and Kosslyn, S. M. (1993). The role of parts and spatial relations in object identification. *Perception*, 22(2):229–248.

[Ceci et al., 2007] Ceci, M., Berardi, M., and Malerba, D. (2007). Relational data mining and ilp for document image processing. *Applied Artificial Intelligence*, 21(8):317–342.

[Centeno, 1997] Centeno, J. S. (1997). Segmentation of Thematic Maps Using Colour and Spatial Attributes. In *GREC*, pages 233–239.

[Changhua et al., 2000] Changhua, L., Bing, Y., and Weixin, X. (2000). Online hand-sketched graphics recognition using attributed relational graph. In *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, pages 2549–2553.

[Chaudhuri and Garain, 2000] Chaudhuri, B. B. and Garain, U. (2000). An approach for recognition and interpretation of mathematical expressions in printed document. *Pattern Analysis and Applications*, 3(2):120–131.

[Chiang et al., 1998] Chiang, J. Y., Tue, S. C., and Leu, Y. C. (1998). A new algorithm for line image vectorization. *Pattern Recognition*, 31(10):1541–1549.

[Chong et al., 2003] Chong, C., Raveendran, P., and Mukudan, R. (2003). A comparative analysis of algorithms for fast computation of zernike moments. *Pattern Recognition*, 36:731–742.

[Claveau and Sébillot, 2004] Claveau, V. and Sébillot, P. (2004). From Efficiency to Portability: Acquisition of Semantic Relations by Semi-Supervised Machine Learning. In *Proceedings of International Conference on Computational Linguistics*, pages 261–267, Geneva, Switzerland.

[Coetzer, 2005] Coetzer, J. (2005). *Off-line Signature Verification*. PhD thesis, University of Stellenbosch.

[Collin and Colnet, 1994] Collin, S. and Colnet, D. (1994). Syntactic analysis of technical drawing dimensions. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(5):1131–1148.

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

[Cordella and Vento, 2000a] Cordella, L. P. and Vento, M. (2000a). Symbol and shape recognition. In Chhabra, A. K. and Dori, D., editors, *Graphics Recognition, Recent Advances*, volume 1941 of *Lecture Notes in Computer Science*, pages 167–182. Springer.

[Cordella and Vento, 2000b] Cordella, L. P. and Vento, M. (2000b). Symbol recognition in documents: a collection of techniques? *International Journal on Document Analysis and Recognition*, 3(2):73–88.

[Cortelazzo et al., 1994] Cortelazzo, G. M., Mian, G. A., Vezzi, G., and Zamperoni, P. (1994). Trademark shapes description by string-matching techniques. *Pattern Recognition*, 27:1005–1018.

[Cover and Thomas, 2006] Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory, 2nd Edition*. John Wiley.

[da Fontoura Costa and Junior, 2001] da Fontoura Costa, L. and Junior, R. M. C. (2001). *Shape Analysis and Classification: Theory and Practice*. CRC Press – Book series on Image Processing.

[Davies and Bouldin, 1979] Davies, D. L. and Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

[Deans, 1983] Deans, S. R. (1983). *Applications of the Radon Transform*. Wiley Interscience Publications, New York.

[Dehak, 2002] Dehak, S. (2002). *Inference Quantitative des Relations Spatiales Directionnelles*. Thèse de doctorat, École Nationale Supérieure des Télécommunications.

[Delalandre et al., 2008] Delalandre, M., Valveny, E., and Lladós, J. (2008). Performance evaluation of symbol recognition and spotting systems: An overview. In *Proceedings of International Workshop on Document Analysis Systems*, pages 497–505.

[Doermann, 1998] Doermann, D. S. (1998). An Introduction to Vectorization and Segmentation. In Tombre, K. and Chhabra, A. K., editors, *GREC— Algorithms and Systems*, volume 1389 of *Lecture Notes in Computer Science*, pages 1–8.

[Domingos et al., 2008] Domingos, P., Kok, S., Lowd, D., Poon, H., Richardson, M., and Singla, P. (2008). Markov logic. In *Probabilistic Inductive Logic Programming*, pages 92–117.

[Dori, 1995] Dori, D. (1995). Vector-based arc segmentation in the machine drawing understanding system environment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1057–1068.

[Dosch and Lladós, 2003] Dosch, P. and Lladós, J. (2003). Vectorial signatures for symbol discrimination. In *Graphics Recognition, Springer, Lecture Notes in Computer Science Series*, pages 154–165.

[Dosch et al., 2000] Dosch, P., Masini, G., and Tombre, K. (2000). Improving arc detection in graphics recognition. In *Proceedings of the IAPR International Conference on Pattern Recognition*, volume 2, pages 243–246.

[Duda et al., 2001] Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification - 2nd edition*. Wiley Interscience Publications, New York.

[Dunn, 1974] Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104.

[Dutta, 1991] Dutta, S. (1991). Approximate spatial reasoning: integrating qualitative and quantitative constraints. *International Journal of Approximate Reasoning*, 5:307–331.

[Egenhofer and Herring, 1991] Egenhofer, M. and Herring, J. R. (1991). Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases. In *University of Maine, Research Report*.

[Egenhofer and Shariff, 1998] Egenhofer, M. and Shariff, A. R. (1998). Metric Details for Natural-Language Spatial Relations. *ACM Transactions on Information System*, 16(4):295–321.

[E.Jungert, 1993] E.Jungert (1993). Qualitative spatial reasoning for determination of object relations using symbolic interval projections. In *IEEE Symposium on Visual Languages*, pages 24–27.

[Fankhauser et al., 2011] Fankhauser, S., Riesen, K., and Bunke, H. (2011). Speeding up graph edit distance computation through fast bipartite matching. In Jiang, X., Ferrer, M., and Torsello, A., editors, *Graph-Based Representations in Pattern Recognition*, volume 6658 of *Lecture Notes in Computer Science*, pages 102–111. Springer.

[Feng et al., 2009] Feng, G., Viard-Gaudin, C., and Sun, Z. (2009). On-line hand-drawn electric circuit diagram recognition using 2d dynamic programming. *Pattern Recognition*, 42(12):3215–3223.

[Fierens, 2010] Fierens, D. (2010). On the relationship between logical bayesian networks and probabilistic logic programming based on the distribution semantics. In *Proceedings of the 19th international conference on Inductive logic programming*, pages 17–24, Berlin, Heidelberg. Springer-Verlag.

[Fornés, 2009] Fornés, A. (2009). *Writer Identification by a Combination of Graphical Features in the Framework of Old Handwritten Musical Documents*. PhD thesis, Computer Vision Center, University Autonoma de Barcelona, Spain.

[Fornés et al., 2010] Fornés, A., Lladós, J., Sánchez, G., and Karatzas, D. (2010). Rotation invariant hand-drawn symbol recognition based on a dynamic time warping model. *International Journal on Document Analysis and Recognition*, 13(3):229–241.

[Fränti et al., 2000] Fränti, P., Mednonogov, A., Kyrki, V., and Kälviäinen, H. (2000). Content-based matching of line-drawing images using the hough transform. *International Journal on Document Analysis and Recognition*, 3(2):117–124.

[Freeman, 1975] Freeman, J. (1975). The modelling of spatial relations. *Computer Graphics and Image Processing*, 4:156–171.

[FRESH, 2007] FRESH (2007). Final report on symbol recognition with evaluation of performances. Deliverable 2.4.2. - FP6-516059.

[Fujiyoshi et al., 2010] Fujiyoshi, A., Suzuki, M., and Uchida, S. (2010). Grammatical verification for mathematical formula recognition based on context-free tree grammar. *Mathematics in Computer Science*, 3(3):279–298.

[Garnesson and Giraudon, 1990] Garnesson, P. and Giraudon, G. (1990). Spatial context in an image analysis system. In *Proceedings of European Conference on Computer Vision*, pages 579–582, London, UK. Springer-Verlag.

[Getoor and Taskar, 2007] Getoor, L. and Taskar, B. (2007). *Introduction to Statistical Relational Learning*. MIT Press, Cambridge.

[Gevers and Smeulders, 1992] Gevers, T. and Smeulders, A. W. M. (1992). Σnigma: An Image Retrieval System. In *Proceedings of the IAPR International Conference on Pattern Recognition*, volume 2, pages 697–700.

[Gordo et al., 2010] Gordo, A., Fornés, A., Valveny, E., and Lladós, J. (2010). A bag of notes approach to writer identification in old handwritten musical scores. In Doermann, D. S., Govindaraju, V., Lopresti, D. P., and Natarajan, P., editors, *Proceedings of International Workshop on Document Analysis Systems*, ACM International Conference Proceeding Series, pages 247–254. ACM.

[Goyal and Egenhofer, 2001] Goyal, R. K. and Egenhofer, M. J. (2001). Similarity of Cardinal Directions. In *Advances in Spatial and Temporal Databases*, volume 2121 of *Lecture Notes in Computer Science*, pages 36–55.

[GREC, 2003] GREC (2003). International symbol recognition contest at grec2003.

[Grigorescu and Petkov, 2003] Grigorescu, C. and Petkov, N. (2003). Distance sets for shape filters and shape recognition. *IEEE Transactions on Image Processing*, 12(10):1274–1286.

[Grishman and Sundheim, 1996] Grishman, R. and Sundheim, B. (1996). Message understanding conference - 6: A brief history. In *Proceedings of the International Conference on Computational Linguistics*.

[Groen et al., 1985] Groen, F. C., Sanderson, A. C., and Schlag, J. F. (1985). Symbol recognition in electrical diagrams using probabilistic graph matching. *Pattern Recognition Letters*, 3(5):343–350.

[Güting, 1988] Güting, R. H. (1988). Geo-relational algebra: A model and query language for geometric database systems. In *Proceedings of the International Conference on Extending Database Technology: Advances in Database Technology*, pages 506–527.

[Heidemann, 2004] Heidemann, G. (2004). Combining spatial and colour information for content based image retrieval. *Computer Vision and Image Understanding*, 94:234–270.

[Heutte et al., 2004] Heutte, L., Nosary, A., and Paquet, T. (2004). A multiple agent architecture for handwritten text recognition. *Pattern Recognition*, 37(4):665–674.

[Hilaire and Tombre, 2006] Hilaire, X. and Tombre, K. (2006). Robust and accurate vectorization of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):890–904.

[Horváth et al., 2009] Horváth, T., Paass, G., Reichartz, F., and Wrobel, S. (2009). A logic-based approach to relation extraction from texts. In *ILP*, volume 5989 of *Lecture Notes in Computer Science*, pages 34–48. Springer.

[Hse and Newton, 2004] Hse, H. and Newton, A. R. (2004). Sketched symbol recognition using zernike moments. In *Proceedings of the IAPR International Conference on Pattern Recognition*, pages 367–370.

[J-K and S-Z, 2005] J-K, K. and S-Z, H. (2005). Radon transform orientation estimation for rotation invariant texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):1004–1008.

[Jain and Dubes, 1988] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[Jain et al., 2000] Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37.

[Janssen and Vossepoel, 1997] Janssen, R. D. T. and Vossepoel, A. M. (1997). Adaptive Vectorization of Line Drawing Images. *Computer Vision and Image Understanding*, 65(1):38–56.

[Jayadevan et al., 2009] Jayadevan, R., Kolhe, S., and Patil, P. (2009). Dynamic time warping based static hand printed signature verification. *Journal of Pattern Recognition Research*, 4(1):52.

[Jegou et al., 2008] Jegou, H., Douze, M., and Schmid, C. (2008). Hamming embedding and weak geometric consistency for large scale image search. In Forsyth, D. A., Torr, P. H. S., and Zisserman, A., editors, *Proceedings of European Conference on Computer Vision*, volume 5302 of *Lecture Notes in Computer Science*, pages 304–317. Springer.

[Jégou et al., 2009] Jégou, H., Douze, M., and Schmid, C. (2009). Packing bag-of-features. In *Proceedings of International Conference on Computer Vision*, pages 2357–2364.

[Jiang et al., 1999] Jiang, X., Münger, A., and Bunke, H. (1999). Synthesis of representative graphical symbols by computing generalized median graph. In *Graphics Recognition, Springer, Lecture Notes in Computer Science Series*, pages 183–192.

[Jouili and Tabbone, 2011] Jouili, S. and Tabbone, S. (2011). Towards performance evaluation of graph-based representation. In *Proceedings of the IAPR Graph-Based Representations in Pattern Recognition*, pages 72–81.

[Jouili et al., 2010] Jouili, S., Tabbone, S., and Lacroix, V. (2010). Median graph shift: A new clustering algorithm for graph domain. In *Proceedings of the IAPR International Conference on Pattern Recognition*, pages 950–953.

[Kankanhalli et al., 1995] Kankanhalli, M. S., Mehtre, B. M., and Wu, J. K. (1995). Cluster-based color matching for image retrieval. *Pattern Recognition*, 29:701–708.

[Kasturi et al., 2002] Kasturi, R., O'Gorman, L., and Govindaraju, V. (2002). Document image analysis: A primer. *Character Recognition*, 27(1):3–22.

[K.C. et al., 2009a] K.C., S., Lamiroy, B., and Ropers, J.-P. (2009a). Inductive logic programming for symbol recognition. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1330–1334. IEEE Computer Society.

[K.C. et al., 2010] K.C., S., Nattee, C., and Lamiroy, B. (2010). Spatial similarity based stroke number and order free clustering. In *Proceedings of the IAPR International Conference on Frontier Handwriting Recognition*, pages 652–657. IEEE Computer Society.

[K.C. et al., 2009b] K.C., S., Wendling, L., and Lamiroy, B. (2009b). New ways to handle spatial relations through angle plus mbr theory on raster documents. In *Proceedings of IAPR International Workshop on Graphics Recognition*, pages 291–302, La Rochelle, France.

[K.C. et al., 2009c] K.C., S., Wendling, L., and Lamiroy, B. (2009c). Unified pairwise spatial relations: An application to graphical symbol retrieval. In *GREC*, pages 163–174.

[Keogh and Pazzani, 1999] Keogh, E. J. and Pazzani, M. J. (1999). Scaling up dynamic time warping to massive dataset. In *European PKDD*, pages 1–11.

[Kim et al., 1993] Kim, J. H., Suh, J. W., and Kim, J. H. (1993). Recognition of logic diagrams by identifying loops and rectilinear polylines. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 349–352.

[Kim and Kim, 2000] Kim, W.-Y. and Kim, Y.-S. (2000). A region-based shape descriptor using zernike moments. *Signal Processing: Image Communication*, 16(1-2):95 – 102.

[Kimia et al., 1995] Kimia, B., Tannenbaum, A., and Zucker, S. (1995). Shapes shocks and deformations i: the components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224.

[Kodratoff and Michalski, 1990] Kodratoff, Y. and Michalski, R. S. (1990). *Machine Learning: An Artificial Intelligence Approach*. 3rd edition.

[Kruskall and Liberman, 1983] Kruskall, J. B. and Liberman, M. (1983). The symmetric time warping algorithm: From continuous to discrete. In *Time Warps, String Edits and Macromolecules: The Theory and Practice of String Comparison*, pages 125–161. Addison-Wesley.

[Kudo and Sklansky, 2000] Kudo, M. and Sklansky, J. (2000). Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1):25–41.

[Kuncheva, 2005] Kuncheva, L. I. (2005). Diversity in multiple classifier systems. *Information Fusion*, 6(1):3–4.

[Lamiroy et al., 2007] Lamiroy, B., Gaucher, O., and Fritz, L. (2007). Robust Circle Detection. In Flavio Bortolozzi and Robert Sabourin, editors, *Proceedings of International Conference on Document Analysis and Recognition*, pages 526–530, Curitiba Brasil.

[Lamiroy and Guebbas, 2009] Lamiroy, B. and Guebbas, Y. (2009). Robust and precise circular arc detection. In *Graphics Recognition, Springer, Lecture Notes in Computer Science Series*, pages 49–60.

[Lamiroy and Guebbas, 2010] Lamiroy, B. and Guebbas, Y. (2010). Robust and precise circular arc detection. In Ogier, J.-M., Liu, W., and Lladós, J., editors, *Graphics Recognition. Achievements, Challenges, and Evolution, 8th International Workshop, GREC 2009, La Rochelle, France, July 22-23, 2009. Selected Papers*, volume 6020 of *Lecture Notes in Computer Science*, pages 49–60. Springer-Verlag.

[Lamiroy et al., 2009] Lamiroy, B., Langa, K., and Leoutre, B. (2009). Assessing classification quality by image synthesis. In *Proceedings of IAPR International Workshop on Graphics Recognition*, La Rochelle, France.

[Latecki et al., 2000] Latecki, L. J., Lakämper, R., and Eckhardt, U. (2000). Shape descriptors for non-rigid shapes with a single closed contour. In *Computer Vision and Pattern Recognition*, pages 1424–1429.

[Lazebnik et al., 2006] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition*, pages 2169–2178.

[Leavers, 2000] Leavers, V. (2000). Use of the two-dimensional radon transform to generate a taxonomy of shape for the characterization of abrasive powder particles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1411–1423.

[Lee et al., 1990] Lee, S., Kim, J., and Groen, F. (1990). Translation- rotation- and scale invariant recognition of hand-drawn symbols in schematic diagrams. *International Journal of Pattern Recognition and Artificial Intelligence*, 4(1):1.

[Lee and Hsu, 1992] Lee, S.-H. and Hsu, F.-J. (1992). Spatial Reasoning and Similarity Retrieval of Images Using 2D C-string Knowledge Representation. *Pattern Recognition*, 25(3):305–318.

[Lemire, 2009] Lemire, D. (2009). Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recognition*, 42(9):2169–2180.

[Li et al., 2005] Li, F.-F., Perona, P., and of Technology, C. I. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition*, pages 524–531. IEEE Computer Society.

[Liu et al., 2004] Liu, Y., Wenyin, L., and Jiang, C. (2004). A Structural Approach to Recognizing Incomplete Graphic Objects. In *Proceedings of the IAPR International Conference on Pattern Recognition*.

[Lladós and Kwon, 2004] Lladós, J. and Kwon, Y.-B., editors (2004). *Graphics Recognition, Recent Advances and Perspectives, 5th Intnl. Workshop, GREC 2003, Barcelona, Spain, July 30-31, 2003, Revised Selected Papers*, volume 3088 of *LNCS*. Springer.

[Lladós et al., 2001] Lladós, J., Martí, E., and Villanueva, J. J. (2001). Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1137–1143.

[Lladós et al., 2002] Lladós, J., Valveny, E., Sánchez, G., and Martí, E. (2002). Symbol Recognition: Current Advances and Perspectives. In Blostein, D. and Kwon, Y.-B., editors, *GREC – Algorithms and Applications*, volume 2390 of *Lecture Notes in Computer Science*, pages 104–127. Springer-Verlag.

[Loncaric, 1998] Loncaric, S. (1998). A Survey of Shape Analysis Techniques. *Pattern Recognition*, 31(8):983–1001.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[Luqman et al., 2009] Luqman, M. M., Brouard, T., and Ramel, J.-Y. (2009). Graphic symbol recognition using graph based signature and bayesian network classifier. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1325–1329.

[Macé et al., 2010] Macé, S., Locteau, H., Valveny, E., and Tabbone, S. (2010). A system to detect rooms in architectural floor plan images. In *Proceedings of International Workshop on Document Analysis Systems*, pages 167–174.

[Malon et al., 2008] Malon, C., Uchida, S., and Suzuki, M. (2008). Mathematical symbol recognition with support vector machines. *Pattern Recognition Letters*, 29(9):1326 – 1332.

[Marszalek and Schmid, 2006] Marszalek, M. and Schmid, C. (2006). Spatial weighting for bag-of-features. In *Computer Vision and Pattern Recognition*, pages 2118–2125.

[Mas Romeu et al., 2007] Mas Romeu, J., Sanchez, G., Llados, J., and Lamiroy, B. (2007). An Incremental On-line Parsing Algorithm for Recognizing Sketching Diagrams. In Flavio Bortolozzi and Robert Sabourin, editors, *Proceedings of International Conference on Document Analysis and Recognition*, pages 452–456, Curitiba Brasil.

[Matsakis et al., 2001] Matsakis, P., Keller, J., Wendling, L., Marjamaa, J., and Sjahputera, O. (2001). Linguistic Description of Relative Positions in Images. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 31(4):573 –588.

[Matsakis et al., 2004] Matsakis, P., Keller, J. M., Sjahputera, O., and Marjamaa, J. (2004). The Use of Force Histograms for Affine-Invariant Relative Position Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):1–18.

[Matsakis and Wendling, 1999] Matsakis, P. and Wendling, L. (1999). A New Way to Represent the Relative Position Between Areal Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(7):634–643.

[Medasani and Krishnapuram, 1997] Medasani, S. and Krishnapuram, R. (1997). A fuzzy approach to content-based image retrieval. In *Proc. of FUZZ-IEEE*, pages 1251–1260.

[Mehtre et al., 1995] Mehtre, B. M., Kankanhalli, M. S., Narasimhalu, A. D., and Man, G. C. (1995). Color matching for image retrieval. *Pattern Recognition Letters*, 16(3):325–331.

[Messmer and Bunke, 1998] Messmer, B. and Bunke, H. (1998). Clustering and error-corrrecting matching of graphs for learning and recognition of symbols in engineering drawings. In Hull, J. and Taylor, S., editors, *Document Analysis Systems II*, pages 102–117. World Scientific.

[Michalski et al., 1986] Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (1986). *Machine Learning: An Artificial Intelligence Approach.* 2nd edition.

[Michalski and Tecuci, 1994] Michalski, R. S. and Tecuci, G. (1994). *Machine Learning: A Multistrategy Approach.* 4th edition.

[Mikolajczyk and Schmid, 2004] Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill.

[Mitra, 2002] Mitra, D. (2002). A Class of Star-Algebras for Point-Based Qualitative Reasoning in Two-Dimensional Space. In *Fifteenth Intl. Florida Artificial Intelligence Research Society Conference*, pages 486–491.

[Miyajima and Ralescu, 1994] Miyajima, K. and Ralescu, A. (1994). Spatial Organization in 2D Segmented Images: Representation and Recognition of Primitive Spatial Relations. *Fuzzy Sets and Systems*, 2(65):225–236.

[Monagan and Roosli, 1993] Monagan, G. and Roosli, M. (1993). Appropriate base representation using a run graph. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 623–626.

[Moravec, 1977] Moravec, H. P. (1977). Towards automatic visual obstacle avoidance. In *Proceedings of International Joint Conference on Artificial Intelligence*.

[Morris, 2003] Morris, A. (2003). A framework for modeling uncertainty in spatial databases. *Transactions in GIS*, 7:83–101.

[Muggleton et al., 1994] Muggleton, S., Luc, and Raedt, D. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19:629–679.

[Mundy and Zisserman, 1992] Mundy, J. L. and Zisserman, A. (1992). *Geometric Invariance in Computer Vision*. MIT Press.

[Myers et al., 1995] Myers, G. K., Mulgaonkar, P. G., huei Chen, C., Decurtins, J. L., and Chen, E. (1995). Verification-based approach for automated text and feature extraction from raster-scanned maps. In *Graphics Recognition, Springer, Lecture Notes in Computer Science Series*, pages 190–203.

[Nagy, 2000] Nagy, G. (2000). Twenty years of document image analysis in pami. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):38–62.

[Nemhauser and Wolsey, 2004] Nemhauser, G. L. and Wolsey, L. A. (2004). *Integer and Combinatorial Optimization*. Wiley-Interscience, New York.

[Nguyen et al., 2008] Nguyen, T.-O., Tabbone, S., and Terrades, O. R. (2008). Symbol descriptor based on shape context and vector model of information retrieval. In *Proceedings of International Workshop on Document Analysis Systems*, pages 191–197.

[Nienhuys-Cheng and Wolf, 1997] Nienhuys-Cheng, S.-H. and Wolf, R. d. (1997). *Foundations of Inductive Logic Programming*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Okazaki et al., 1988] Okazaki, A., Tsunekawa, S., Kondo, T., Mori, K., and Kawamoto, E. (1988). An automatic circuit diagram reader with loop-structure-based symbol recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):331–341.

[Otsu, 1979] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62 –66.

[Paek and John, 1998] Paek, S. and John, R. (1998). Detecting image purpose in world-wide web documents. In *Proc. IS&T/SPIE Symposium on Electronic Imaging: Science and Technology - Document Recognition*, pages 151–158.

[Papadias and Theodoridis, 1997] Papadias, D. and Theodoridis, Y. (1997). Spatial relations, minimum bounding rectangles, and spatial data structures. *International Journal of Geographical Information Science*, 11(2):111–138.

[Patel et al., 2010] Patel, A., Ramakrishnan, G., and Bhattacharya, P. (2010). Incorporating linguistic expertise using ilp for named entity recognition in data hungry indian languages. In *Proceedings of the 19th international conference on Inductive logic programming*, pages 178–185, Berlin, Heidelberg. Springer-Verlag.

[Pavlidis, 1977] Pavlidis, T. (1977). *Structural Pattern Recognition*. Springer-Verlag, Berlin-Heidelberg-New York.

[Persoon and Fu, 1986] Persoon, E. and Fu, K.-S. (1986). Shape discrimination using fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(3):388 –397.

[Peuquet and CI-Xiang, 1987] Peuquet, D. and CI-Xiang, Z. (1987). An algorithm to determine the directional relationship between arbitrarily-shaped polygons in the plane. *Pattern Recognition*, 20(1):65–74.

[Pham and Smeulders, 2006] Pham, T. V. and Smeulders, A. W. M. (2006). Learning spatial relations in object recognition. *Pattern Recognition Letters*, 27(14):1673–1684.

[Philbin et al., 2007] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition*.

[Plotkin, 1971] Plotkin, G. (1971). *Automatic Methods of Inductive Inference*. PhD thesis, Edinburgh University.

[Psyllos et al., 2010] Psyllos, A. P., Anagnostopoulos, C.-N., and Kayafas, E. (2010). Vehicle logo recognition using a sift-based enhanced matching scheme. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):322–328.

[Pullar and Egenhofer, 1988] Pullar, D. and Egenhofer, M. (1988). Towards formal definitions of topological relations among spatial objects. In *In: D. Marble, editor, The Third International Symposium on Spatial Data Handling*, pages 225–242.

[Raedt and Kersting, 2008] Raedt, L. D. and Kersting, K. (2008). Probabilistic inductive logic programming. In *Probabilistic Inductive Logic Programming*, volume 4911, pages 1–27.

[Rebelo et al., 2010] Rebelo, A., Capela, G., and Cardoso, J. S. (2010). Optical recognition of music symbols: A comparative study. *International Journal on Document Analysis and Recognition*, 13(1):19–31.

[Rendek et al., 2004] Rendek, J., Masini, G., Dosch, P., and Tombre, K. (2004). The search for genericity in graphics recognition applications: Design issues of the qgar software system. In Marinai, S. and Dengel, A., editors, *Proceedings of International Workshop on Document Analysis Systems*, volume 3163 of *Lecture Notes in Computer Science*, pages 366–377. Springer-Verlag.

[Rendek and Wendling, 2008] Rendek, J. and Wendling, L. (2008). On determining suitable subsets of decision rules using choquet integral. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(2):207–232.

[Renz and Mitra, 2004] Renz, J. and Mitra, D. (2004). Qualitative direction calculi with arbitrary granularity. In *Proceedings of the Pacific Rim International Conferences on Artificial Intelligence*, pages 65–74.

[Renz and Nebel, 1998] Renz, J. and Nebel, B. (1998). Spatial reasoning with topological information. In *Spatial Cognition, An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, pages 351–372, London, UK. Springer-Verlag.

[Retz-Schmidt, 1988] Retz-Schmidt, G. (1988). Various Views on Spatial Prepositions. *AI Magazine*, pages 95–104.

[Reynolds et al., 2006] Reynolds, A. P., Richards, G., Iglesia, B., and Rayward-Smith, V. J. (2006). Clustering rules: A comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms*, 5:475–504.

[Richardson and Domingos, 2006] Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine Learning*, 62:107 – 136.

[Rohlf and Fisher, 1968] Rohlf, F. J. and Fisher, D. L. (1968). Test for hierarchical structure in random data sets. *Systematic Zool.*, 17:407–412.

[Rosenfeld, 1974] Rosenfeld, A. (1974). Adjacency in digital pictures. *Information and Control*, 26(1):24–33.

[Rousseeuw, 1987] Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65.

[Roy et al., 2008] Roy, P. P., Vazquez, E., Lladós, J., Baldrich, R., and Pal, U. (2008). A system to segment text and symbols from color maps. In Liu, W., Lladós, J., and Ogier, J.-M., editors, *Graphics Recognition. Recent Advances and New Opportunities*, volume 5046 of *Lecture Notes in Computer Science*, pages 245–256. Springer.

[Rusiñol and Lladós, 2008] Rusiñol, M. and Lladós, J. (2008). Word and symbol spotting using spatial organization of local descriptors. In *Proceedings of International Workshop on Document Analysis Systems*, pages 489–496.

[Russell and Norvig, 2010] Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition.

[Ruta and Gabrys, 2000] Ruta, D. and Gabrys, B. (2000). An overview of classifier fusion methods. *Computing and Info. Sys.*, 7(1):1 – 10.

[Saitta et al., 2007] Saitta, S., Raphael, B., and Smith, I. F. (2007). A bounded index for cluster validity. In *Proceedings of International conference on Machine Learning and Data Mining in Pattern Recognition*, pages 174–187, Berlin, Heidelberg. Springer-Verlag.

[Salmon, 2008] Salmon, J.-P. (2008). *Reconnaissance de Symboles Complexes*. Thèse de doctorat, Institut National Polytechnique de Lorraine.

[Salmon et al., 2007] Salmon, J. P., Wendling, L., and Tabbone, S. (2007). Improving the recognition by integrating the combination of descriptors. *International Journal on Document Analysis and Recognition*, 9(1):3–12.

[Samet and Soffer, 1996a] Samet, H. and Soffer, A. (1996a). Marco: Map retrieval by content. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:783–798.

[Samet and Soffer, 1996b] Samet, H. and Soffer, A. (1996b). Marco: Map retrieval by content. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:783–798.

[Schmitt et al., 2008] Schmitt, E., Bombardier, V., and Wendling, L. (2008). Improving fuzzy rule classifier by extracting suitable features from capacities with respect to the choquet integral. *IEEE, Systems, Man, and Cybernetics, Part B*, 38(5):1195–1206.

[Sebastian et al., 2001] Sebastian, T. B., Klein, P. N., and Kimia, B. B. (2001). Recognition of shapes by editing shock graphs. In *Proceedings of International Conference on Computer Vision*, pages 755–762.

[Sebastian et al., 2003] Sebastian, T. B., Klein, P. N., and Kimia, B. B. (2003). On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125.

[Smith, 2010] Smith, E. H. B. (2010). An analysis of binarization ground truthing. In *Proceedings of International Workshop on Document Analysis Systems*, pages 27–34. ACM.

[Sokal and Rohlf, 1962] Sokal, R. R. and Rohlf, F. J. (1962). The Comparison of Dendrograms by Objective Methods. *Taxon*, 11(2):33–40.

[Song et al., 2002] Song, J., Su, F., Tai, C.-L., and Cai, S. (2002). An Object-Oriented Progressive-Simplification Based Vectorization System for Engineering Drawings: Model, Algorithm, and Performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1048–1060.

[Song et al., 2009] Song, L., Cheng, X., Guo, Y., Liu, Y., and Ding, G. (2009). Contentex: A framework for automatic content extraction programs. In *IEEE International Conference on Intelligence and Security Informatics*, pages 188 –190.

[Sun and Chen, 2011] Sun, S.-K. and Chen, Z. (2011). Robust logo recognition for mobile phone applications. *Journal of Infornamtion Science and Engineering*, 27(2):545–559.

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

[Tabbone et al., 2003] Tabbone, S., Wendling, L., and Tombre, K. (2003). Matching of graphical symbols in line-drawing images using angular signature information. *International Journal on Document Analysis and Recognition*, 6(2):115–125.

[Teh and Chin, 1988] Teh, C.-H. and Chin, R. T. (1988). On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):496–513.

[Tenenbaum, 2008] Tenenbaum, J. B. (2008). Building theories of the world: Human and machine learning perspectives. In *ILP*, volume 5194 of *Lecture Notes in Computer Science*, page 1. Springer.

[Terrades et al., 2007a] Terrades, O. R., Tabbone, S., and Valveny, E. (2007a). A review of shape descriptors for document analysis. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 227–231.

[Terrades et al., 2007b] Terrades, O. R., Valveny, E., and Tabbone, S. (2007b). On the combination of ridgelets descriptors for symbol recognition. In *GREC*, pages 40–50.

[Terrades et al., 2009] Terrades, O. R., Valveny, E., and Tabbone, S. (2009). Optimal classifier fusion in a non-bayesian probabilistic framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1630–1644.

[Tombre, 1998] Tombre, K. (1998). Analysis of engineering drawings: State of the art and challenges. In Tombre, K. and Chhabra, A. K., editors, *Graphics Recognition, Algorithms and Systems*, volume 1389 of *Lecture Notes in Computer Science*, pages 257–264. Springer.

[Tombre, 2010] Tombre, K. (2010). Graphics recognition - what else? In Ogier, J.-M., Liu, W., and Lladós, J., editors, *Graphics Recognition. Achievements, Challenges, and Evolution*, volume 6020 of *Lecture Notes in Computer Science*, pages 272–277. Springer.

[Tombre et al., 1998] Tombre, K., Ah-Soon, C., Dosch, P., Habed, A., and Masini, G. (1998). Stable, robust and off-the-shelf methods for graphics recognition. *Proceedings of the IAPR International Conference on Pattern Recognition*, 1:406.

[Tombre and Lamiroy, 2008] Tombre, K. and Lamiroy, B. (2008). Pattern recognition methods for querying and browsing technical documentation. In *Proceedings of Iberoamerican Congress on Pattern Recognition*, pages 504–518.

[Torsello and Hancock, 2004] Torsello, A. and Hancock, E. R. (2004). A skeletal measure of 2d shape similarity. *Computer Vision and Image Understanding*, 95(1):1–29.

[Valveny and Martí, 2000] Valveny, E. and Martí, E. (2000). Hand-drawn symbol recognition in graphic documents using deformable template matching and a bayesian framework. In *Proceedings of the IAPR International Conference on Pattern Recognition*, pages 2239–2242.

[Valveny and Martí, 2003] Valveny, E. and Martí, E. (2003). A model for image generation and symbol recognition through the deformation of lineal shapes. *Pattern Recognition Letters*, 24(15):2857–2867.

[Valveny et al., 2007] Valveny, E., Tabbone, S., Terrades, O. R., and Philippot, E. (2007). Performance characterization of shape descriptors for symbol representation. In *Graphics Recognition, Springer, Lecture Notes in Computer Science Series*, pages 278–287.

[Vandenbrande and Requicha, 1993] Vandenbrande, J. H. and Requicha, A. A. G. (1993). Spatial Reasoning for the Automatic Recognition of Machinable Features in Solid Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1269–1285.

[Ventura and Schettini, 1994] Ventura, A. D. and Schettini, R. (1994). Graphic Symbol Recognition using a Signature Technique. In *Proceedings of the IAPR International Conference on Pattern Recognition*, volume 2, pages 533–535.

[Wang and Keller, 1999] Wang, X. and Keller, J. (1999). Human-Based Spatial Relationship Generalization Through Neural/Fuzzy Approaches. *Fuzzy Sets and Systems*, 101:5–20.

[Watanabe, 1985] Watanabe, S. (1985). *Pattern Recognition: Human and Mechanical*. John Wiley & Sons, ISBN: 0471808156, New York.

[Wendling et al., 2008] Wendling, L., Rendek, J., and Matsakis, P. (2008). Selection of suitable set of decision rules using choquet integral. pages 947–955.

[Wenyin, 2003] Wenyin, L. (2003). On-line graphics recognition: State-of-the-art. In *Graphics Recognition, Springer, Lecture Notes in Computer Science Series*, pages 291–304.

[Winston, 1975] Winston, P. (1975). *The psychology of computer vision.* McGraw-Hill, New York.

[Worboys, 1995] Worboys, M. (1995). *GIS - A computing perspective.* Taylor and Francis.

[Xiaogang et al., 2004] Xiaogang, X., Zhengxing, S., Binbin, P., Xiangyu, J., and Wenyin, L. (2004). An online composite graphics recognition approach based on matching of spatial relation graphs. *International Journal on Document Analysis and Recognition*, 7(1):44–55.

[Yang et al., 2007] Yang, R., Lu, T., and Cai, S. (2007). A dynamic-rule-based framework of engineering drawing recognition and interpretation system. In *ICIC: International Conference on Advanced intelligent computing theories and applications*, pages 1006–1017, Berlin, Heidelberg. Springer-Verlag.

[Yang, 2005] Yang, S. (2005). Symbol recognition via statistical integration of pixel-level constraint histograms: A new descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):278–281.

[Yu, 1995] Yu, B. (1995). Automatic understanding of symbol-connected diagrams. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 803–806.

[Yu, 2007] Yu, Y. (2007). Online sketched symbol recognition. Master's thesis, City University of Hong Kong.

[Yuen et al., 1998] Yuen, P. C., Feng, G.-C., and Tang, Y. Y. (1998). Printed chinese character similarity measurement using ring projection and distance transform. *International Journal of Pattern Recognition and Artificial Intelligence*, 12(2):209–221.

[Zadeh, 1965] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8:338–353.

[Zhang and Lu, 2002] Zhang, D. and Lu, G. (2002). Shape-based image retrieval using generic fourier descriptor. *Signal Processing: Image Communication*, 17:825–848.

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

[Zhang and Lu, 2005] Zhang, D. and Lu, G. (2005). Study and evaluation of different fourier methods for image retrieval. *Image and Vision Computing*, 23(1):33–49.

[Zhang et al., 2006] Zhang, W., Wenyin, L., and Zhang, K. (2006). Symbol recognition with kernel density matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2020–2024.

[Zhang et al., 2009] Zhang, X., Li, Z., Zhang, L., Ma, W.-Y., and Shum, H.-Y. (2009). Efficient indexing for large scale visual search. In *Proceedings of International Conference on Computer Vision*, pages 1103–1110.

[Zheng et al., 2005] Zheng, Y., Li, H., and Doermann, D. S. (2005). A parallel-line detection algorithm based on hmm decoding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):777–792.

[Zhu and Yuille, 1996] Zhu, S. C. and Yuille, A. L. (1996). FORMS: A Flexible Object Recognition and Modelling System. *International Journal of Computer Vision*, 20(3):187–212.

Authorisation

Nancy-Université
INPL

AUTORISATION DE SOUTENANCE DE THESE
DU DOCTORAT DE L'INSTITUT NATIONAL
POLYTECHNIQUE DE LORRAINE

o0o

VU LES RAPPORTS ETABLIS PAR :

Monsieur Thierry PAQUET, Professeur, LITIS EA, Université de Rouen, Mont Saint Aignan
Monsieur Jean-Yves RAMEL, Professeur, Université de François Rabelais, Tours

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur K.C. Santosh

à soutenir devant un jury de l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE,
une thèse intitulée :

"Graphics Recognition Using Spatial Relations and Shape Analysis"

en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : **" Informatique "**

Fait à Vandoeuvre, le 22 novembre 2011
Le Président de l'I.N.P.L.,
F. LAURENT

La Vice-Présidente
du Conseil d'Administration de l'INPL

Christine ROIZARD

Institut National Polytechnique de Lorraine
inpl@inpl-nancy.fr | www.inpl-nancy.fr

2 av. de la Forêt de Haye | BP 3
54501 Vandoeuvre-lès-Nancy Cedex
FRANCE
Tél. : +33(0)3 83 59 59 59
Fax : +33(0)3 83 59 59 55

Abstract

In the current state-of-the-art, symbol recognition usually means recognising isolated symbols. However, isolated symbol recognition methods are not always suitable for solving real-world problems. In case of composite documents that contain textual and graphical elements, one needs to be able to extract and formalise the links that exist between the images and the surrounding text, in order to exploit the information embedded in those documents.

Related to this context, we first introduce a method for graphics recognition based on dynamic programming matching of the Radon features. This method allows to exploit the Radon Transform property to include both boundary and internal structure of shapes without compressing the pattern representation into a single vector that may miss information. The method outperforms all major set of state-of-the-art of shape descriptors but remains mainly suited for isolated symbol recognition only. We therefore integrate it in a completely new approach for symbol recognition based on the spatio-structural description of a 'vocabulary' of extracted visual primitives. The method is based on spatial relations between pairs of labelled vocabulary types (some of which can be characterised with the previously mentioned descriptor), which are further used as a basis for building an attributed relational graph (ARG) to describe symbols. Thanks to our labelling of attribute types, we avoid the general NP-hard graph matching problem. We provide a comprehensive comparison with other spatial relation models as well as state-of-the-art approaches for graphics recognition and prove that our approach effectively combines structural and statistical descriptors together and outperforms them significantly.

In the final part of this thesis, we present a Bag-Of-Features (BOFs) approach using spatial relations where every possible pair of individual visual primitives is indexed by its topological configuration and the visual type of its components. This provides a way to retrieve isolated symbols as well as significant known parts of symbols by applying either an isolated symbol as a query or a collection of relations between the important visual primitives. Eventually, it opens perspectives towards natural language based symbol recognition process.

Keywords — Radon Features, Dynamic Programming, Shape Descriptors, Visual Vocabulary, Spatial Relations, Spatial-Bag-of-Features, Graphics Recognition.

Résumé

Dans l'état de l'art actuel, la reconnaissance de symboles signifie généralement la reconnaissance des symboles isolés. Cependant, ces méthodes de reconnaissance de symboles isolmés ne sont pas toujours adaptés pour résoudre les problèmes du monde réel. Dans le cas des documents composites qui contiennent des éléments textuels et graphiques, on doit être capable d'extraire et de formaliser les liens qui existent entre les images et le texte environnant, afin d'exploiter les informations incorporées dans ces documents.

Liés à ce contexte, nous avons d'abord introduit une méthode de reconnaissance graphique basée sur la programmation dynamique et la mise en correspondance de caractéristiques issues de la transormée de Radon. Cette méthode permet d'exploiter la propriété de cette transformée pour inclure à la fois le contour et la structure interne des formes sans utiliser de techniques de compression de la représentation du motif dans un seul vecteur et qui pourrait passer à côté d'informations importantes. La méthode surpasse en performances les descripteurs de forme de l'état de l'art, mais reste principalement adapté pour la reconnaissance de symboles isolés seulement. Nous l'avons donc intégrée dans une approche complètement nouvelle pour la reconnaissance de symboles basé sur la description spatio-structurelle d'un «vocabulaire» de primitives visuelles extraites. La méthode est basée sur les relations spatiales entre des paires de types étiquetés de ce vocabulaire (dont certains peuvent être caractérisés avec le descripteur mentionné précédemment), qui sont ensuite utilisées comme base pour construire un graphe relationnel attribué (ARG) qui décrit des symboles. Grâce à notre étiquetage des types d'attribut, nous évitons le problème classique NP-difficile d'appariement de graphes. Nous effectuons une comparaison exhaustive avec d'autres modèles de relations spatiales ainsi qu'avec l'état de l'art des approches pour la reconnaissance des graphismes afin de prouver que notre approche combine efficacement les descripteurs statistiques structurels et globaux et les surpasse de manière significative.

Dans la dernière partie de cette thèse, nous présentons une approche de type sac de caractéristiques utilisant les relations spatiales, où chaque paire possible primitives visuelles est indexée par sa configuration topologique et les types visuels de ses composants. Ceci fournit un moyen de récupérer les symboles isolés ainsi que d'importantes parties connues de symboles en appliquant soit un symbole isolée comme une requêten soit une collection de relations entre les primitives visuelles. Finalement, ceci ouvre des perspectives vers des processus de reconnaissance de symboles fondés sur le langage naturel.

Mots-clés — descripteur de Radon, programmation dynamique, descripteurs de forme, vocabulaire visuel, relations spatiales, sac de caractéristiques spatiales, reconnaissance graphique.