



HAL
open science

Reconnaissance et localisation de symboles dans les documents graphiques : approches basées sur le treillis de concepts

Ameni Boumaiza

► **To cite this version:**

Ameni Boumaiza. Reconnaissance et localisation de symboles dans les documents graphiques : approches basées sur le treillis de concepts. Autre [cs.OH]. Université de Lorraine, 2013. Français. NNT : 2013LORR0028 . tel-01749418

HAL Id: tel-01749418

<https://hal.univ-lorraine.fr/tel-01749418v1>

Submitted on 29 Mar 2018

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

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



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Reconnaissance et Localisation de Symboles dans les Documents Graphiques : Approches basées sur le Treillis de Concepts

THÈSE

présentée et soutenue publiquement le 20 Mai 2013

pour l'obtention du

Doctorat de l'université de Lorraine

(spécialité informatique)

par

Ameni BOUMAIZA

Composition du jury

<i>Rapporteurs :</i>	Jean-Marc OGIER	Professeur de l'Université de la Rochelle
	Josep LLADOS	Professeur de l'Universidad Autonoma de Barcelona
<i>Examineurs :</i>	Elisa FROMONT	Maître de Conférence de l'Université de Saint-Etienne
	Jean-Paul HATON	Professeur de l'Université de Lorraine
	Véronique EGLIN	Maître de Conférence de l'Université de Lyon
<i>Directeur de thèse :</i>	Salvatore TABBONE	Professeur de l'Université de Lorraine

"Soyons reconnaissants aux personnes qui nous donnent du bonheur ; elles sont les charmants jardiniers par qui nos âmes sont fleuries."

Marcel Proust

J'adresse mes vifs remerciements à Monsieur Jean-Marc OGIER et Monsieur Josep LLADOS pour avoir accepté d'être rapporteurs de ce travail. Je tiens également à remercier Madame Elisa FROMONT, Madame Véronique EGLIN et Monsieur Jean-Paul HATON de m'avoir fait l'honneur de participer à ce jury de thèse.

Je tiens à exprimer mes remerciements les plus sincères à Monsieur Salvatore TABBONE, mon directeur de thèse, pour m'avoir conseillée, encouragée et soutenue tout au long de la thèse avec patience et disponibilité, et pour la confiance qu'il m'a accordée.

Enfin, les mots les plus simples étant les plus forts, j'adresse toute mon affection à ma famille, et en particulier à ma maman Mabrouka qui m'a fait comprendre que la vie n'est pas faite que de problèmes qu'on pourrait résoudre grâce à des formules mathématiques et des algorithmes, à mon père Khemaïes pour tout ce qu'il a fait pour me soutenir et m'offrir tout ce que j'en ai besoin depuis que j'étais petite. Malgré mon éloignement depuis de nombreuses années, leur intelligence, leur confiance, leur tendresse, leur amour me portent et me guident tous les jours. Merci pour avoir fait de moi ce que je suis aujourd'hui. Je tiens à remercier mes deux âmes sœurs, mes deux belles petites sœurs qui étaient à mes côtés des les moments difficiles. C'est vrai que je suis la sœur aînée mais tchoutchou w ammoula montraient une maturité qui m'a aidée à surmonter toutes les difficultés. Une pensée à mon cher frère Maher qui était toujours à mes côtés pour m'aider et me soutenir. Je vous aime. Que Dieu vous bénisse.

J'adresse toute ma gratitude à tous mes ami(e)s (Nabil, Khaled, Hana, Salim, Nihel, Mehdi, Mohamed, Ghazi, Moutie, Bilel, Wahiba, Kort, Walid...) et à toutes les personnes qui m'ont aidé dans la réalisation de ce travail. Je remercie toutes les personnes formidables que j'ai rencontrées par le biais de l'INRIA. Merci pour votre support et vos encouragements.

Je remercie tous les membres et ex-membres de l'équipe QGAR pour le climat sympathique dans lequel ils m'ont permis de travailler. Les nombreuses discussions que j'ai pu avoir avec chacun m'ont beaucoup apporté.

Une pensée pour terminer ces remerciements à tous ceux que je ne nomme pas, mais qui se reconnaîtront.

*Je dédie cette thèse
à ma mère Mabrouka,
à mon père Khemaïes,
à mon frère Maher,
à mes sœurs Amel et Kaouther.*

Table des matières

Chapitre 1	
Introduction Générale	1
1.1 Cadre Générale	1
1.1.1 La vision par ordinateur	1
1.1.2 Reconnaissance de Formes	2
1.1.3 La Reconnaissance de Symboles Graphiques	3
1.1.4 Localisation de Symboles dans les Documents Graphiques	3
1.1.5 Analyse Formelle de Concepts	4
1.2 Contributions	5
1.2.1 Reconnaissance d'Objets Graphiques	6
1.2.2 Localisation de Symboles dans les Documents Graphiques	7
1.3 Organisation du mémoire	7
1.3.1 Partie 1 : <i>État de l'Art</i>	8
1.3.2 Partie 2 : <i>Reconnaissance de Symboles Graphiques</i>	8
1.3.3 Partie 3 <i>Localisation de Symboles dans les Documents Graphiques</i>	8

Partie I État de l'Art	11
-------------------------------	-----------

Chapitre 1 Reconnaissance de Symboles	13
1.1 Introduction	13
1.2 Segmentation	15
1.3 Description des primitives	18
1.3.1 Approches Statistiques	18

1.3.2	Discussion	24
1.3.3	Approches Structurelles	25
1.3.4	Discussion	27
1.3.5	Conclusion sur la description des primitives	27
1.4	Apprentissage et Classification	28
1.4.1	Principe de la classification	29
1.4.2	k -plus proche voisin	30
1.4.3	k -means	30
1.4.4	Arbre de Décision	31
1.4.5	Les machines à vecteur de support	31
1.4.6	Classifieur de Bayes	32
1.4.7	Conclusion	33
Chapitre 2 Treillis de Galois et Analyse Formelle des Concepts		35
2.1	Introduction	36
2.2	Étude Théorique de la structure du Treillis de Galois	36
2.3	Fondements Mathématiques	37
2.3.1	Préliminaires et Définitions	37
2.3.2	Un exemple illustratif d'un contexte Formel	40
2.4	Algorithmes de Construction du Treillis de Galois	42
2.4.1	Phase d'apprentissage	42
2.4.2	Phase de Classement	49
2.4.3	Complexité des Algorithmes	49
2.5	Les règles d'association	49
2.6	Systèmes à Base du Treillis de Galois	51
2.6.1	GRAND	51
2.6.2	Complexité	52
2.6.3	LEGAL et LEGAL-E	52
2.6.4	Principe et Construction du pseudo-treillis	52
2.6.5	Complexité	53
2.6.6	GALOIS	53
2.6.7	RULEARNER	54
2.6.8	Apprentissage et Classement	54
2.6.9	CIBLe	54
2.6.10	Complexité	54
2.6.11	CLNB & CLNN	55
2.6.12	Complexité et Comparaison expérimentale	55

2.6.13	Comparaison des systèmes à base du Treillis de Galois	55
2.6.14	Discussion	56
2.7	Conclusion	57

Partie II	Reconnaissance de Symboles Graphiques	59
------------------	--	-----------

Chapitre 1	Reconnaissance de Symboles Graphiques	61
1.1	Introduction	62
1.2	Approche Proposée	63
1.2.1	Représentation de symboles	64
1.2.2	L'approche sac de mots	67
1.3	Le Treillis de Galois et la Classification	68
1.3.1	Phase d'apprentissage	68
1.3.2	Construction du treillis de Galois	72
1.3.3	Comment attribuer une classe à un objet ?	75
1.3.4	Discussion	76
1.4	Résultats Expérimentaux	76
1.4.1	Protocole d'expérimentation	76
1.4.2	Mesures de performances	77
1.4.3	Résultats et Discussions	79
1.4.4	Évaluation de la taille du treillis de concepts en fonction du nombre d'attributs	81
1.4.5	Comparaison avec quelques classifieurs usuels	85
1.4.6	Évaluation de l'étape de représentation de symboles	85
1.4.7	Comportement de la taille du treillis de concepts en fonction de la taille du vocabulaire	94
1.5	Conclusion	94

Chapitre 2 Treillis de Galois Fréquent pour la reconnaissance de symboles	97
2.1 Introduction	97
2.2 Approche proposée	98
2.3 Résultats Expérimentaux	105
2.3.1 Comparaison du TGF avec les autres approches	106
2.3.2 Invariance aux transformations : Bases GREC_{scale} , GREC_{rot} et $\text{GREC}_{scale+rot}$	112
2.3.3 Invariance aux déformations ou occlusions : Bases GREC_{degr1} , GREC_{degr2} et GREC_{dist1}	112
2.3.4 Évaluation des techniques de sélection de caractéristiques	114
2.3.5 Évaluation du taux de reconnaissance du treillis de Galois en fonction de la taille du dictionnaire	114
2.4 Conclusion	115
Chapitre 3 Application à la Reconnaissance d'Images	119
3.1 Introduction	119
3.2 État de l'Art	121
3.3 Le Formalisme du Treillis de Galois	123
3.4 Formulation du Problème	123
3.5 Approche Proposée	124
3.6 Résultats Expérimentaux	126
3.6.1 Comparaison avec les classifieurs de l'état de l'art	127
3.7 Évaluation des performances	133
3.7.1 Évaluation des performances du treillis en fonction du nombre d'attributs	133
3.7.2 Évaluation du taux de reconnaissance en fonction de la taille de l'échantillon d'apprentissage	134
3.8 Étude de la Complexité du treillis	134
3.9 Conclusion	135

Partie III Localisation de symboles dans les documents graphiques **137**

Chapitre 1 État de l'Art sur les Approches de Localisation de Symboles dans les Documents	139
1.1 Introduction	139

1.2	Localisation de Symboles dans les Documents Graphiques	140
1.2.1	Approches Structurelles	141
1.2.2	Approches Pixelaires	148
1.2.3	Discussion	151
1.3	Conclusion	152
Chapitre 2 Étude Comparative des Approches Pixelaires pour la Représentation et la Localisation de Symboles dans les Documents Graphiques		155
2.1	Introduction	156
2.2	Formulation du problème	157
2.2.1	Le détecteur de Harris (1988)	158
2.2.2	Le Détecteur Harris-Laplace (2004)[Mikolajczyk and Schmid2001]	159
2.2.3	Le Détecteur Hessian-Laplace	160
2.2.4	Le Détecteur SIFT (DoG) [Lowe1999]	160
2.2.5	Le Détecteur <i>MSER</i> [Matas <i>et al.</i> 2004]	161
2.2.6	Discussion	162
2.2.7	Le Détecteur <i>pseudo-aléatoire</i>	163
2.3	Indexation du document graphique avec un vocabulaire visuel	163
2.3.1	Construction du Vocabulaire Visuel	164
2.3.2	Correspondance entre le Vocabulaire Visuel	164
2.3.3	Indexation du Document Graphique	165
2.3.4	Localisation de symboles dans les documents graphiques	166
2.4	Résultats Expérimentaux	168
2.4.1	Comparaison des Détecteurs de Points d'Intérêt	169
2.5	Discussion	171
2.6	Conclusion	171
2.7	Impact de l'étape de représentation sur les résultats de la localisation	172
2.7.1	Méthode a	172
2.7.2	Méthode b	173
2.7.3	Méthode c	173
2.7.4	Méthode d	173
2.8	Conclusion	182
Chapitre 3 Localisation de Symboles dans les Documents Graphiques en se basant sur le Treillis de Galois		183
3.1	Introduction	183
3.2	Approche Proposée	184

3.2.1	Description de l'approche	185
3.2.2	Extraction des primitives	187
3.2.3	Construction du Vocabulaire Visuel	188
3.2.4	Sélection des primitives	188
3.2.5	Détection de symboles dans les documents graphiques	189
3.3	Résultats Expérimentaux	191
3.3.1	Impact de l'étape de sélection des attributs et des concepts sur la taille du treillis	194
3.4	Conclusion	194

Partie IV	Conclusion Générale	197
------------------	----------------------------	------------

Chapitre 1	Conclusion	199
-------------------	-------------------	------------

1.1	Publications Scientifiques	202
-----	--------------------------------------	-----

Glossaire	203
------------------	------------

Bibliographie	205
----------------------	------------

Table des figures

1.1	Contexte des différentes thématiques abordées dans le cadre de la thèse.	2
1.2	Exemples de quelques documents graphiques.	4
1.1	Technique de segmentation de symboles basée sur les contours actifs : un contour actif peut se dilater et tenter de remplir une forme, il sera alors situé à l'intérieur de celle-ci au démarrage de l'algorithme. À chaque itération, l'algorithme va tenter de trouver un meilleur positionnement pour le contour pour minimiser les dérives par rapport aux contraintes utilisées. L'algorithme s'arrêtera lorsqu'il ne sera plus possible d'améliorer le positionnement ou simplement quand le nombre maximum d'itérations aura été atteint.	16
1.2	Opérateur Laplacien pour la détection des contours à différentes tailles de masque.	17
1.3	La squelettisation de symboles graphiques.	17
1.4	Illustration graphique de la transformée de Radon d'une fonction $f(x, y)$. La transformée de Radon est une transformation de l'image de l'espace (x, y) à l'espace des paramètres (θ, ρ) et peut être représentée mathématiquement par une droite intégrante de (x, y) tout le long de toutes les lignes L paramétrées par (θ, ρ) et représentées dans l'espace de l'image (x, y)	20
1.5	Définition de la transformée de Radon pour plusieurs images de différentes échelles et orientations (extrait à partir de [Tabbone <i>et al.</i> 2006]).	21
1.6	Descripteur Générique de Fourier : la première image est l'image du symbole d'origine, la deuxième étape décrit le passage de l'image et la transformée de Fourier dans le domaine polaire et la troisième étape décrit le descripteur générique de Fourier.	22
1.7	Parties réelles de la fonction de base de ART (extrait à partir de [Ricard <i>et al.</i> 2004]).	22
1.8	Les étapes de calcul du contexte de forme. (a) et (b) Représentation des échantillons des points de contours de deux formes. (c) Schéma d'histogrammes log-polaires utilisés dans le calcul des contextes de forme. 5 pages sont utilisés pour $\log r$ et 12 pages pour θ . (d), (e) et (f) Contextes de forme des points marqués par \circ et \triangleleft en (a) et par \diamond en (b) (reprise de [Belongie <i>et al.</i> 2002]).	23
1.9	Calcul du descripteur SIFT.	24
1.10	À gauche : dans le cas d'une région homogène, toutes les valeurs sont relativement faibles. Au milieu : en présence de fréquences dans la direction x , la valeur de $ dx $ est élevée, mais les autres valeurs restent faibles. Si l'intensité augmente progressivement dans la direction x , les deux valeurs $\sum dx$ et $ dx $ sont élevées (cet exemple est extrait de [Bay <i>et al.</i> 2006]).	25
1.11	Représentation d'une image sous forme de graphe.	25
1.12	Graphe de voisinage des points d'intérêt d'un symbole graphique.	26

1.13	Représentation d'une image par un graphe de lignes (reprise de [Delalandre <i>et al.</i> 2008]).	26
1.14	Différents niveaux de détail avec leur graphe correspondant (reprise de [Fonseca and Jorge2003]).	27
1.15	Un exemple d'un dendrogramme où la position de deux objets au sein d'un groupe ou entre les groupes n'est pas liée à leur proximité dans l'espace des données. La longueur/hauteur de chaque branche est égale à la distance moyenne entre les groupes. En traçant une ligne verticale/horizontale à une hauteur donnée de l'arbre, on peut identifier k groupes (http://neamar.fr/Res/Fouille_Sociale/fouille_donnees.pdf).	29
1.16	Classification avec les k -plus proche voisin (http://julien.pinquier.free.fr/These/these/node29.html).	30
1.17	Classification avec les k -means (http://www.mathworks.com/matlabcentral/fileexchange/24616-kmeans-clustering).	31
1.18	Exemple d'un arbre de décision (http://imss-www.upmf-grenoble.fr/prevert/Prog/Tris/ArbreDecision.html).	31
1.19	Machine à vecteur de support (http://kokzard.blogspot.fr/2011/10/jfjdkshfkjsldf.html).	32
2.1	Représentation du (a) diagramme de Hasse construit en se basant sur l'algorithme de Godin [Godin and Missaoui1994] et du (b) treillis d'héritage de l'exemple présenté dans la Table.2.1. Le diagramme de Hasse est un graphe composé des nœuds appelés aussi concepts contenant deux parties : l' <i>extension</i> représente l'ensemble des objets partageant les mêmes caractéristiques de l' <i>intension</i> . Le premier concept se situant en haut du graphe contient : $\{\{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8\}, \emptyset\}$. Dans le treillis d'héritage les concepts ne contiennent pas toute l'information car les parties <i>extension</i> et <i>intension</i> sont héritées d'un niveau à un autre.	41
2.2	Le Treillis de concepts du contexte de l'exemple des "planètes" généré en utilisant l'algorithme de Godin.	44
2.3	Le Treillis de concepts du contexte de l'exemple des "planètes" généré en utilisant l'algorithme de Nourine et Raynaud.	46
2.4	Le treillis de concepts du contexte de l'exemple des "planètes" généré en utilisant l'algorithme de Norris.	47
2.5	Le diagramme de Hasse du contexte de l'exemple des "planètes" généré en utilisant l'algorithme de Bordat.	48
1.1	Les différentes étapes de la chaîne de traitement proposée dans le cadre de la reconnaissance des symboles graphiques.	63
1.2	Illustration de l'architecture du système proposé pour la reconnaissance de symboles graphiques.	64
1.3	Descripteur SURF appliqué sur quelques échantillons de symboles isolés.	67
1.4	Vocabulaire visuel ayant une taille $ \Omega =270$ mots visuels construit sur les échantillons de symboles graphiques de la base GREC2003.	68
1.5	Table binarisée du contexte $\mathcal{K}=(\mathcal{O}, \mathcal{A}, \mathcal{R})$.	71
1.6	Comparaison entre les deux algorithmes de construction du treillis de concepts : Nourine et Norris en terme du nombre de concepts et le temps nécessaire pour leur génération.	72
1.7	Un exemple illustratif du processus incrémental d'insertion d'un nouvel objet lors de la phase d'apprentissage et de construction du treillis de Galois.	73



1.8	Un exemple illustratif de l'étape de la mise à jour des classes des nœuds du treillis suite à l'insertion d'un nouvel objet lors de la phase d'apprentissage.	74
78figure.2.1.9		
1.10	La représentation graphique d'un treillis de Galois pour un sous-ensemble de 100 symboles et 60 attributs.	83
1.11	Exemple illustratif d'un processus de classification d'un symbole requête. (a) Le treillis de Galois représentant les relations entre 15 symboles et 9 attributs. (a) et (b) Mise à jour du treillis de concepts suite à l'introduction de l'objet requête, (c) Navigation dans l'arborescence de bas vers le haut en vue de la détermination des concepts dont la partie <i>intension</i> correspond à la partie <i>intension</i> de l'objet requête, (d) Les concepts retenus comme candidats résultant de la phase de la classification.	86
1.12	Comparaison des taux de <i>Précision/Rappel</i> du treillis de Galois et les classifieurs usuels de l'état de l'Art. Dans cette expérimentation, nous avons utilisé 9 modèles de symboles, chaque modèle contient 50 symboles. Le nombre de symboles d'apprentissage utilisés dans cette expérimentation est 302 symboles bruités et 98 symboles pour l'échantillon du test. Dans (a) la taille du vocabulaire est $ \Omega =30$, (b) $ \Omega =50$ et (c) $ \Omega =80$	87
1.13	Comparaison des taux de <i>Précision/Rappel</i> du treillis de Galois+BoF et du treillis de Galois+signatures numériques. Dans cette expérimentation, nous avons utilisé 9 modèles de symboles, chaque modèle contient 50 symboles. Le nombre de symboles d'apprentissage utilisés dans cette expérimentation est 302 symboles bruités et 98 symboles pour l'échantillon du test. (a) $ \Omega =30$, (b) $ \Omega =50$ et (c) $ \Omega =80$	88
1.14	Évaluation du taux de classification du treillis de Galois en fonction de la taille de l'échantillon d'apprentissage en utilisant la combinaison des descripteurs suivants : Moments de Zernike (Z), GFD (G) et la <i>R</i> -signature (R).	88
1.15	Sélection des caractéristiques du descripteur SIFT extrait d'un symbole graphique en se basant sur la méthode d'ACP. À gauche, nous avons l'ensemble des vecteurs SIFTS représentant un symbole graphique. À droite, l'ensemble des variables sélectionnées par la méthode ACP représentant les vecteurs SIFTS les plus discriminants : il s'agit de 5 groupes de variables principales.	90
1.16	Évaluation des performances du treillis de Galois en fonction de la taille du vocabulaire. Dans cette expérimentation, nous avons utilisé 5 modèles de symboles, chaque modèle contient 30 symboles. Le nombre de symboles d'apprentissage utilisés dans cette expérimentation est 102 symboles bruités et 48 symboles pour l'échantillon du test. La taille du vocabulaire est $ \Omega =50$	94
2.1	Idée de filtrage du dictionnaire visuel en fonction de <i>l'informativité</i> et de la <i>pertinence</i> des mots visuels apparaissant dans deux modèles de symboles graphiques. \circ , \circ , \circ et \diamond représentent des mots visuels appartenant au vocabulaire Ω	99
2.2	L'architecture de notre système proposé pour la reconnaissance de symboles graphiques basé sur la structure TGF. Les étapes sont les suivantes : (1) Extraction et description des primitives à partir des symboles, (2) Quantification des descripteurs visuels et construction du vocabulaire visuel, (3) Filtrage des motifs pertinents et fréquents à partir du vocabulaire, (4) Construction de la table binaire modélisant les relations de $\{Présence/Absence\}$ entre symboles et attributs, (5) Construction du treillis de Galois et reconnaissance du symbole requête et (6) Reconnaissance du symbole requête.	101

2.3	Approche adoptée pour le filtrage du vocabulaire visuel en se basant sur les probabilités normalisées (dans cet exemple, le seuil de comparaison est fixé à 50%).	102
2.4	Le vocabulaire visuel et la technique de seuillage utilisée pour le partitionnement du dictionnaire visuel : $\Omega = \Omega_{Freq} \cup \Omega_{NonFreq}$	103
2.5	Évolution du taux de reconnaissance en utilisant le Treillis de Galois combiné à Ω et Ω_{Freq} en fonction du nombre de caractéristiques. Dans cette expérimentation, la base de symboles GREC2003 est partitionnée en un ensemble d'apprentissage constitué de 1000 instances de symboles et un ensemble de test comportant 500 symboles. $ \Omega $ varie entre 50 et 500 mots visuels.	105
2.6	Comparaison entre les taux de reconnaissance des classifieurs suivants : $TGF_{\Omega_{Freq}}$, $TG_{\Omega_{NonFreq}}$ et TG_{Ω} en fonction de θ (le seuil à partir duquel nous pouvons parler de la notion de <i>pertinence</i> donc possibilité de filtrage du vocabulaire sans perte de discrimination). Dans cette expérimentation, nous avons choisi une taille du vocabulaire $\ \Omega\ $ entre 50 et 500 mots visuels. Un mot visuel w_i est considéré comme <i>pertinent</i> et $w_i \in \Omega_{Freq}$ si sa probabilité d'apparition dans un modèle de symboles k dépasse le seuil θ , sinon $w_i \in \Omega_{NonFreq}$	107
2.7	Évaluation du taux de reconnaissance en utilisant le TG_{Ω} , $TGF_{\Omega_{Freq}}$, le classifieur Bayésien et le classificateur SVM (dans cette expérimentation, $\theta=2/3$, $\ \text{NbSymb}\ =500$ et Ω_{Freq} est l'ensemble des mots visuels fréquents. Nous avons testé les classifieurs sur 6 modèles de symboles : (a) $\Omega=50$ mots visuels et (b) $\Omega=100$ mots visuels. Le noyau utilisé pour le classifieur SVM est linéaire.	107
2.8	Évaluation du taux de reconnaissance (%) des différents classificateurs en fonction de la taille du vocabulaire visuel.	108
2.9	Comparaison entre la taille du $TGF_{Symboles, \Omega_{Freq}}$ et le $TG_{Symboles, \Omega}$ en fonction de la taille du dictionnaire visuel.	111
2.10	Courbes de <i>Précision/Rappel</i> pour la reconnaissance de symboles. Dans cette expérimentation, nous avons utilisé les bases de modèles de symboles suivantes : (a) $GREC_{scale}$, (b) $GREC_{rot}$ et (c) $GREC_{scale+rot}$ en vue d'évaluer les performances des descripteurs vs notre approche (pour la description des bases voir le chapitre précédent). Nous avons effectué les tests de recherche des symboles similaires à une requête avec les 50 modèles de chaque base. Le nombre moyen d'occurrences de chaque symbole dans ces trois bases varie entre 0 et 13 occurrences.	113
2.11	Courbes <i>Précision/Rappel</i> pour la reconnaissance de symboles. Dans cette expérimentation, nous avons utilisé les bases de modèles de symboles suivantes : (a) $GREC_{degr1}$, (b) $GREC_{dist1}$ et (c) $GREC_{dist2}$ en vue d'examiner le comportement des descripteurs pour les symboles déformés. Nous avons effectué les tests de recherche des symboles similaires à une requête avec les 50 modèles de chaque base. Le nombre moyen d'occurrences de chaque symbole dans ces bases varie entre 0 et 11 occurrences.	113
2.12	Évaluation des techniques de sélection de caractéristiques dans le cadre de la reconnaissance des symboles graphiques en se basant sur le treillis de Galois.	114
2.13	Comparaison des performances obtenues par les trois structures du treillis de Galois suivants : $TGF_{\Omega_{Freq}}$, $TG_{\Omega_{NonFreq}}$ et le TG_{Ω} en fonction de la taille du vocabulaire visuel utilisé (a) $\ \Omega\ =50$, (b) : $\ \Omega\ =100$, (c) : $\ \Omega\ =300$, (d) : $\ \Omega\ =400$, (e) : $\ \Omega\ =500$, (f) : $\ \Omega\ =800$	116
3.1	L'architecture proposée pour la chaîne de traitement de reconnaissance d'images en se basant sur le treillis de Galois _{image}	120

3.2	Exemples des résultats obtenus dans le cadre d'un système de reconnaissance d'images.	122
3.3	Les différentes étapes de la construction du treillis de Galois _{image} basé sur la représentation par sac de mots.	124
3.4	Échantillons des images de la base Caltech101.	127
3.5	Le taux de reconnaissance des classifieurs usuels en comparaison avec le treillis de Galois, en fonction du nombre de symboles de l'ensemble d'apprentissage. Le nombre d'images d'apprentissage varie entre 0 et 30 images. Les courbes montrent les performances des classifieurs en utilisant un seul descripteur (le descripteur SIFT est utilisé dans cette expérimentation) dans le cadre de la classification d'images provenant des bases Caltech101 : (a) Aeroplane, (b) Person, (c) Cars, (d) Boat, (e) Birds. Notre approche n'a pas besoin d'étape préalable de configuration de paramètres à l'encontre du SVM et le réseau Bayésien. En ce qui concerne le classifieur SVM, nous avons utilisé le noyau linéaire obtenu comme suit : $\mathcal{K}(x, x_i)=x*x_i$. Pour le k -ppv nous avons choisi $k=4$	128
3.6	Le taux de reconnaissance des classifieurs usuels en comparaison avec le treillis de Galois, en fonction du nombre de symboles de l'ensemble d'apprentissage. Le nombre d'images d'apprentissage varie entre 0 et 30 images. Les courbes montrent les performances des classifieurs en utilisant un seul descripteur (le descripteur SIFT est utilisé dans cette expérimentation) dans le cadre de la classification d'images provenant des bases Caltech256 : (a) Échantillon _{apprentissage} =5, (b) Échantillon _{apprentissage} =10, (c) Échantillon _{apprentissage} =20 et (d) Échantillon _{apprentissage} =30. Notre approche n'a pas besoin d'étape préalable de configuration de paramètres à l'encontre du SVM et le réseau Bayésien. En ce qui concerne le classifieur SVM, nous avons utilisé le noyau linéaire obtenu comme suit : $\mathcal{K}(x, x_i)=x*x_i$. Pour le k -ppv nous avons choisi $k=4$	130
3.7	Évaluation du taux de reconnaissance du treillis de concepts en fonction de la taille du vocabulaire visuel. Le nombre d'images d'apprentissage varie entre 0 et 30 images. Les courbes montrent les performances des classifieurs en utilisant un seul descripteur (le descripteur SIFT est utilisé dans cette expérimentation) pour la classification d'images provenant des bases Caltech 256 : (a) Aeroplane, (b) Person, (c) Cars, (d) Boat.	133
3.8	Évaluation du taux de reconnaissance retourné par le treillis de Galois en fonction de la taille de l'ensemble d'apprentissage. Le nombre d'images test est 15 images. La taille du vocabulaire utilisée dans cette expérimentation $ \Omega =150$	134
1.1	Quelques exemples de documents graphiques où les symboles sont enfouis dans le document.	140
1.2	Symbole requête et la segmentation correspondante du document graphique (extrait de [Müller and Rigoll2000]).	141
1.3	Symbole requête et les résultats du processus de localisation (extrait de [Müller and Rigoll2000]).	141
1.4	Les différents types de relations existantes entre deux segments pour la construction de la signature vectorielle : la colinéarité (C), le recouvrement (R), le parallélisme (P), la jonction "T" (T) et la jonction "V" (V) (extrait de [Dosch and Lladós2003]).	142
1.5	Quelques résultats de localisation de symbole dans des dessins techniques obtenus par la méthode du calcul des signatures vectorielles. Les régions d'intérêt susceptibles de contenir le symbole requête sont présentées par un rectangle contenant le nom du symbole(s) détecté (extrait de [Dosch and Lladós2003]).	143

1.6	(a)Un dendrogramme, (b)Les partitions trouvées (extrait de [Zuwala and Tabbone2006]).	143
1.7	Pré-traitements du document (a) 1 ^{ère} étape : binarisation du document, (b) 2 ^{ème} étape : squelettisation du document, (c) 3 ^{ème} étape : localisation des points de jonctions, (d) 4 ^{ème} étape : décomposition en chaînes de points (extrait de [Zuwala and Tabbone2006]).	144
1.8	Une partie du graphe de jonction définie sur la Figure.1.7 (extrait de [Zuwala and Tabbone2006]).	144
1.9	Exemple de symboles non traités par la méthode proposé par [Zuwala and Tabbone2006].	145
1.10	Quelques exemples de résultats de localisation obtenus par la méthode de [Rusiñol and Lladós2005].	145
1.11	Les étapes de représentation des symboles. (a) Symbole original (b) Les composantes connectées (c) Les contours de chaque région (d) Approximation polygonale des régions fermées avec le nombre de segments composant chaque polygone (extrait de [Rusiñol and Lladós2007]).	146
1.12	Résultats du processus de localisation de symboles dans les documents graphiques [Rusiñol and Lladós2007] : (a) Symboles requêtes : un lavabot et un bidet (b) Image originale, (c) Les régions correspondantes au lavabot sont marquées en vert et les régions correspondantes au bidet sont marquées en rouge.	146
1.13	(a) Un document, (b) Décomposition du document en arbre X-Y, (c) Arbre X-Y du document, (d) Image requête, (e) Décomposition en arbre X-Y, (f) Arbre X-Y de l'image requête (extrait de [Zanibbi and Yu2011]).	147
1.14	(a) Appariement entre T_a et T_b , (b) Un appariement exacte (extrait de [Zanibbi and Yu2011]).	148
1.15	Graphe de proximité ($k=5$) calculé à partir des points d'intérêt extraits. Chaque point d'intérêt possède un index dans la table de hachage. En se basant sur le graphe et ces indexes, une information sur l'organisation spatiale des descripteurs locaux similaires peut être déduite (extrait de [Rusiñol and Lladós2008]).	149
1.16	(a) Représentation pyramidale de l'image, (b) Représentation pyramidale du modèle, (c) cette figure montre les différentes représentations nécessaires à chaque niveau de la pyramide pour la recherche de la corrélation de gradient. Le modèle est donné en (a). La version du modèle de fenêtres est représentée en (b), avec la fonction de fenêtrage représentée en (c), (d) Surface de corrélation au niveau le plus haut de l'image, les deux extrema les plus forts (blancs) correspondent aux positions de deux occurrences du modèle (extrait à partir de [MacLean and Tsotsos2008]).	149
1.17	Résultats de localisation de portes dans les documents de la méthode proposée par [Escalera <i>et al.</i> 2009].	150
1.18	Résultats de localisation de quelques symboles dans les documents graphiques de la méthode proposée par [Weber and Tabbone2012].	151
1.19	Quelques Résultats de localisation de symboles obtenus avec la méthode [Nguyen <i>et al.</i> 2009].	151
2.1	Approche contour : un point d'intérêt peut être un point d'une courbure.	158
2.2	Approche photométrique.	158
2.3	Approche modèle.	158
2.4	L'idée principale du détecteur de Harris : (a) "flat" : aucun changement de directions, (b) contour : pas de changement tout au long du contour. (c) changements significatifs dans toutes les directions.	159
2.5	Recherche des maxima locaux de R.	159
2.6	Recherche des maxima locaux de Harris-Laplace pour la sélection de l'échelle caractéristique.	160

2.7	Le détecteur de points d'intérêt utilisé dans [Lowe1999] : (a) Construction de la pyramide d'échelles. (b) Détection des extrema des images de DoG : les maxima et minima du DoG sont détectés en comparant un point (marqué par X) avec les 26 points dans la région de voisinage 3x3x3 (marqués par des cercles) de l'échelle courante et de deux échelles adjacentes (extrait de [Lowe1999]).	161
2.8	Application du détecteur <i>MSER</i> sur quelques images (http://www-master.ufr-info-p6.jussieu.fr/specialite/ima/bima/cours7.2p.pdf).	162
2.9	Exemples de quelques indices visuels délimités par des ellipses rouges présentant 4 classes de mots visuels.	164
2.10	Les étapes de la chaîne de traitement pour la localisation de symboles dans les documents graphiques.	165
2.11	Une représentation de deux documents (D_1, D_2) et une requête (Q) dans un espace de 3 termes (t_1, t_2, t_3).	166
2.12	Quelques exemples de symboles dans les bases de test (a) Symboles ayant subi des transformations d'échelle $GREC_{scale}$, (b) Symboles ayant subi des rotations $GREC_{rot}$, (c) Symboles ayant subi des transformations d'échelle et de rotation $GREC_{scale+rot}$, (d) Symboles dégradés niveau 1 de la base $GREC_{degr1}$, (e) Symboles dégradés niveau 2 de la base $GREC_{degr2}$, (f) Symboles détériorés niveau 1 de la base $GREC_{dist1}$, (g) Symboles détériorés niveau 2 de la base $GREC_{dist2}$. . .	170
2.13	Évaluation des détecteurs en fonction de l'impact du nombre de points d'intérêt sur le taux de reconnaissance de symboles. Dans cette expérimentation, nous avons utilisé le classifieur <i>7-ppv</i> en vue de reconnaître les 7 occurrences semblables à la requête.	171
2.14	Exemples de symboles requêtes interrogés lors du processus de localisation. . . .	172
2.15	Localisation des symboles requêtes dans des plans architecturaux en se basant sur le descripteur SIFT. Un appariement est réalisé entre le symbole modèle et le plan architectural.	174
2.16	Localisation des symboles requêtes dans des plans architecturaux en se basant sur le descripteur SURF. Un appariement est réalisé entre le symbole modèle et le dessin électronique.	175
2.17	Localisation de symboles requêtes dans des plans architecturaux en se basant sur le descripteur SURF. Un appariement est trouvé entre le symbole modèle et le plan architectural.	176
2.18	Résultats de localisation de la requête.2.14(j) dans un plan architectural de la base [Delalandre <i>et al.</i> 2007] (résultat extrait de [Nguyen <i>et al.</i> 2009]).	177
2.19	Localisation de quelques symboles requêtes dans des documents graphiques : (a) interrogation de la requête de la Figure.2.14(i), (b) interrogation de la requête de la Figure.2.14(b), (c) interrogation de la requête de la Figure.2.14(g), (d) interrogation de la requête de la Figure.2.14(j) et (e) l'interface de "Symbol Spotting" mise en place.	178
3.1	Illustration des différentes étapes de la chaîne de traitement proposée dans le cadre de la conception d'un système de localisation de symboles dans les documents graphiques.	186
3.2	Un ensemble de mots visuels et les points d'intérêt qui leur sont associés.	190
3.3	Exemples de symboles requêtes interrogés lors du processus de localisation. . . .	192

3.4	Résultats de détection de quelques symboles dans un plan architectural. (a) Projection des points d'intérêt associés aux mots visuels d'un plan architectural, (b) Localisation d'un symbole requête de la Figure.3.3(b), (c) Localisation d'un symbole requête de la Figure.3.3(g).	192
3.5	Évaluation de quelques réponses du système de localisation lors de l'interrogation de 3 requêtes différentes. Req1 est le symbole de la Figure.3.3(f), Req2 est le symbole de la Figure.3.3(c) et Req3 est le symbole de la Figure.3.3(e).	193
3.6	Treillis de Galois $\{Objets, Attributs\}$ avant/après la sélection. Les objets entourés par  sont ceux éliminés suite à l'étape de la sélection des objets et les attributs entourés par  sont ceux éliminés après la sélection des attributs.	195

Introduction Générale

"Language is more than just words. It is any signal (or lack of signal when one is expected), where a signal is a deliberate action that is intended to convey a message".

Clark Herbert

1.1 Cadre Générale

1.1.1 La vision par ordinateur

Omniprésents, la relation *homme-machine* est encore une définition ardue à cerner. Les ordinateurs réalisent dans le temps des tâches récurrentes. Ils aident ainsi l'homme à manipuler d'énormes quantités de données, souvent même plus rapidement et plus précisément que lui. Malgré cela, la capacité des ordinateurs demeure limitée lorsqu'il s'agit d'extraire automatiquement des informations d'images ou de vidéos, qui représentent pourtant des volumes de données extrêmement importants.

La vision par ordinateur est un domaine qui inclut des méthodes d'acquisition, de traitement, d'analyse et de compréhension des images afin de produire de l'information numérique ou symbolique. Un axe de recherche contribuant au développement de ce domaine consiste à reproduire les capacités de la vision humaine par voie électronique afin de percevoir et de comprendre une image. Il s'agit de développer des algorithmes qui reproduisent une des capacités les plus étonnantes du cerveau humain à savoir la déduction des propriétés du monde purement externe au moyen de la lumière qui nous revient des divers objets qui nous entourent. Nous pouvons déterminer par exemple la distance qui nous sépare de ces objets, comment ces derniers sont orientés par rapport à notre vision et la relation qui les lie aux autres objets. Nous pouvons ainsi déterminer d'une façon fiable leurs couleurs et leurs textures et par la suite les reconnaître facilement. La vision par ordinateur considérée comme un sous thème participant à l'intelligence artificielle, peut aussi se diviser en différents domaines comme illustré dans la Figure. 1.1 : la reconnaissance de formes, la reconstruction 3D, l'analyse du mouvement et l'analyse de la couleur. Ces applications ne constituent pas une liste exhaustive mais donnent une idée des objectifs de la vision par ordinateur.

Nos travaux se situent au coeur de cet axe de recherche qui a connu un fort essor avec la montée en flèche du nombre d'images numériques en raison du développement de la vidéosurveillance et de la démocratisation des appareils photos numériques, des téléphones et des réseaux permettant

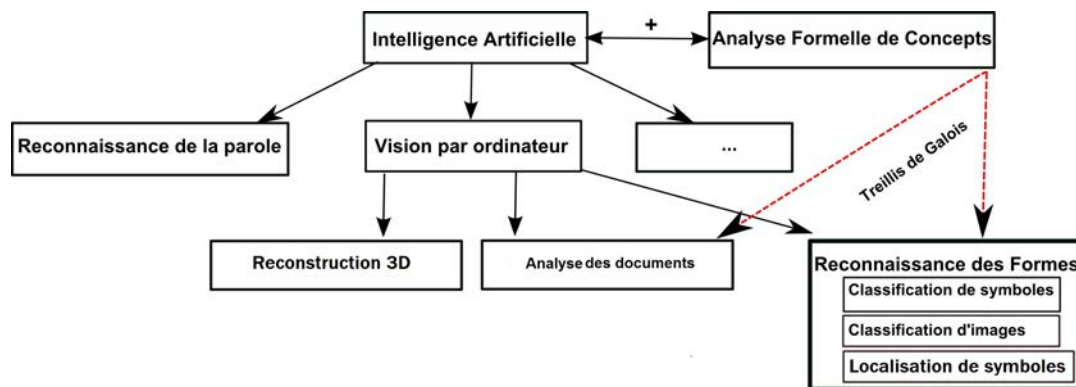


FIGURE 1.1 – Contexte des différentes thématiques abordées dans le cadre de la thèse.

leur échange. Cette thèse a été élaborée et financée dans le cadre du projet européen EUREKA SCANPLAN dont l'objectif est le traitement des documents graphiques et plus particulièrement les plans architecturaux. Nous nous intéressons ici à plusieurs sous-disciplines à savoir la reconnaissance de symboles et leur localisation dans les documents graphiques (dans ce qui suit nous utilisons le mot *symbole* et *objet* indifféremment). Ce type de techniques combine des approches de traitement d'images et d'apprentissage automatique dont le but est de mener à bien cette tâche de reconnaissance des objets présents dans les images. L'amélioration des capacités des ordinateurs à effectuer cette tâche offrirait une multitude de possibilités en termes d'applications. Les efforts consentis dans ce contexte commencent à porter leurs fruits et des applications industrielles commencent à voir le jour. Toutes ces applications, et bien d'autres, expliquent l'effort conduit par la communauté de la vision par ordinateur pour répondre à ce problème. Les travaux présentés dans ce manuscrit participent à cet effort.

1.1.2 Reconnaissance de Formes

La reconnaissance de formes est l'une des thématiques les plus actives dans le domaine de la vision par ordinateur et qui nous intéresse dans ces travaux de thèse. La reconnaissance de symboles est l'une des principaux thèmes de la reconnaissance graphique (voir Figure.1.1). Elle a été intensivement le centre des recherches dans les dernières décennies. Cette thématique a été utilisée dans un grand nombre de domaines comme le domaine de l'ingénierie, l'architecture, la modélisation, la musique, la cartographie, etc. Elle consiste à reconnaître des objets à partir d'un vaste ensemble de classes.

Dans les dernières années, il y a un intérêt croissant dans le développement de bonnes méthodes de reconnaissance de formes. La représentation de la forme est une tâche difficile en raison des distorsions que peut subir un objet, comme les occlusions, les déformations ou le bruit.

L'interprétation automatique des documents graphiques exige des processus capables de reconnaître l'alphabet correspondant à des symboles métiers. Les dessins, les cartes et les diagrammes utilisent des notations graphiques dépendantes de leur domaine d'applications (architecture, dessin). D'une manière très générale, un symbole peut être défini comme une entité graphique significative dans un domaine spécifique d'application (table, toilette, évier). Un plan architectural est constitué de différents types de symboles selon que l'on représente des portes, des fenêtres, des tables. Donc l'élément clef de toute application de reconnaissance de formes concerne la représentation des informations contenues dans les images. Ces informations permettent de caractériser, d'indexer et de faire des recherches sur les images. De nombreuses méthodes de représentation

ont été proposées.

1.1.3 La Reconnaissance de Symboles Graphiques

Elle peut être définie essentiellement comme un processus en deux étapes : l'extraction des signatures et la reconnaissance. L'extraction des signatures peut être réalisée en se basant sur des approches statistiques ou des approches structurelles et syntaxiques. La plupart des méthodes statistiques sont basées sur la répartition des pixels. Elles sont généralement utilisées avec les classifieurs probabilistes ou connexionnistes. D'autre part, les approches syntaxiques et structurelles sont généralement basées sur une description des primitives élémentaires qui sont extraites à partir des symboles (description basique, relations, organisation spatiale, ...). La reconnaissance de formes est généralement classée selon le type de procédure d'apprentissage utilisée pour générer la valeur de sortie.

L'apprentissage supervisé suppose qu'un ensemble de données d'apprentissage (l'ensemble d'apprentissage) a été fourni, constitué par un ensemble d'instances qui ont été correctement étiquetées manuellement avec la sortie correcte. Une procédure d'apprentissage génère ensuite un modèle qui tente de répondre à deux objectifs parfois contradictoires : effectuer aussi bien que possible sur les données d'apprentissage, et de généraliser aussi bien que possible les nouvelles données.

L'apprentissage non supervisé, d'autre part, prend les données d'apprentissage qui n'ont pas été étiquetées manuellement, et tente de trouver des modèles inhérents qui peuvent ensuite être utilisés pour déterminer la valeur de sortie correcte pour les instances de nouvelles données. Une combinaison des deux a récemment été explorée (apprentissage semi-supervisé) utilisant une combinaison de données étiquetées et non étiquetées (généralement un petit ensemble de données étiquetées combiné avec une grande quantité de données non étiquetées). Nous allons passer maintenant à une deuxième thématique traitée durant cette thèse. Il s'agit de la localisation de symboles graphiques dans les documents graphiques.

1.1.4 Localisation de Symboles dans les Documents Graphiques

Une autre thématique intéresse la communauté autant que la reconnaissance des graphiques. Il s'agit de la localisation de symboles dans les documents graphiques. De nos jours, de nombreux documents sont encore stockés au format papier. Une numérisation des documents et leur organisation dans les bibliothèques numériques s'avèrent nécessaires afin de résoudre les problèmes de gain de place et de sauvegarde. Cependant, l'accessibilité aux informations dans les documents numérisés reste un problème ouvert. La gestion d'une grande quantité de données numériques suscite un vif intérêt.

Dans ce travail, nous présentons une méthode pour récupérer les zones des documents graphiques susceptibles de contenir des occurrences d'un certain symbole requête. Cette tâche s'est avérée très complexe et un problème difficile à résoudre. La localisation est liée à l'indexation et à la recherche, et elle a été un sujet d'intérêt très émergent. Les techniques de localisation s'appuient sur différentes méthodes de reconnaissance de formes géométriques, comme les caractéristiques, les composantes connexes, ou la représentation structurelle de symboles.

Dans ce contexte, les symboles ne sont plus isolés de leur contexte. Notre intérêt porté sur ce type d'approches revient au fait que durant les dernières décennies, les contributions proposées dans la littérature se sont focalisées plus autour du problème de reconnaissance de symboles que le problème de détection de symboles graphiques en contexte. Néanmoins, de nombreux domaines ont montré un besoin croissant de l'indexation automatique de documents et de la récupération

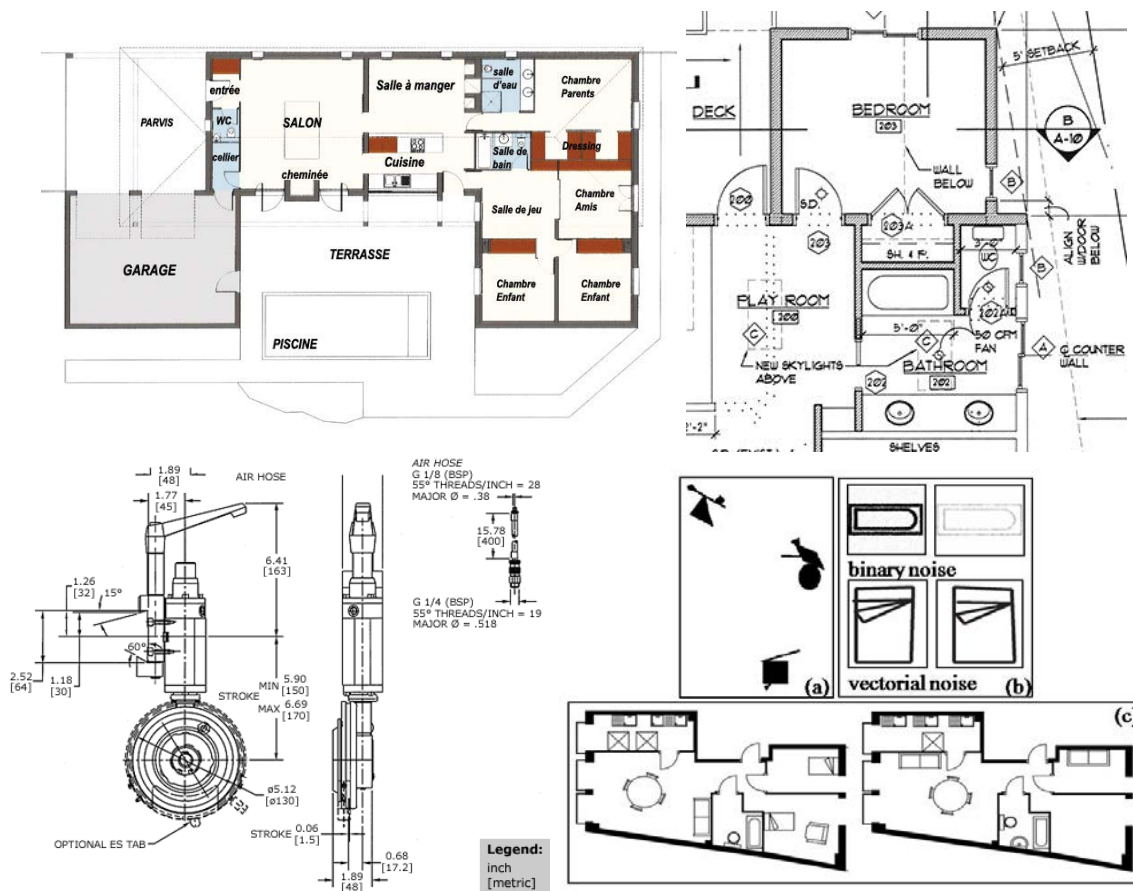


FIGURE 1.2 – Exemples de quelques documents graphiques.

de l'information dans de grandes collections de données. Certains de ces domaines sont basés sur des bases de documents très volumineuses comme par exemple les bases de livres dans les musées ou les bibliothèques, les catalogues de produits ou services dans le domaine du web ou encore les plans architecturaux dans le domaine de l'architecture.

Il en découle ainsi un besoin urgent d'un outil d'accès direct à l'information permettant à l'utilisateur d'une base de documents, de spécifier une requête en termes d'objets graphiques composant le document et d'en retrouver les résultats de sa recherche dans les plus brefs délais et à une précision élevée. La mise en place d'un système de détection de symboles en contexte s'avère nécessaire dans ce cadre. Une telle problématique est plus complexe que celle de la reconnaissance de symboles isolés puisqu'elle a besoin en plus de la phase de reconnaissance d'une phase de localisation.

1.1.5 Analyse Formelle de Concepts

Dans nos travaux de thèse, l'originalité des approches proposées réside dans l'alliance entre le domaine de l'Analyse Formelle de Concepts (AFC) par l'utilisation de la structure du treillis et le domaine de la vision par ordinateur. Cette combinaison s'est avérée très intéressante et prometteuse dans le cadre de la reconnaissance et de la localisation de symboles dans leur contexte. Depuis les années 2000, un intérêt croissant est porté à l'Analyse Formelle des Concepts (AFC) dans de nombreux domaines de l'informatique, aussi bien en fouille de données et en représen-

tation des connaissances, que dans le domaine des ontologies ou encore des bases de données. L'Analyse Formelle de Concepts est une branche de la théorie des treillis basée sur la formalisation de la notion de concepts et de regroupement conceptuel. Elle permet, entre autres, la construction du treillis de concepts. Le treillis des concepts est un graphe possédant la propriété du treillis qui se définit à partir de données organisées sous forme d'une table binaire $\{\text{Objets } x \text{ } \text{Attributs}\}$. Les sommets du graphe sont des concepts - un concept est un regroupement maximal d'objets possédant des attributs en commun. Le treillis ainsi composé de concepts reliés par inclusion, fournit une représentation des données très intuitive. Ce dernier s'est avéré être un cadre théorique intéressant pour la fouille de données puisqu'il permet la génération de concepts (regroupement conceptuel) et de règles d'association. La structure du treillis de concepts est liée principalement au domaine de la fouille du texte où elle constitue une puissante structure de données. La fouille de données, sous toutes ses facettes, permet certainement de découvrir, dans un ensemble volumineux de données, des connaissances fort intéressantes qu'un être humain aurait été dans l'impossibilité de trouver par lui-même. Toutefois, si les connaissances produites sont à leur tour en très grande quantité, elles resteront difficiles à appréhender et pourront en ce sens ne pas être transmises aux personnes intéressées. Les treillis de concepts ne sont pas à l'abri de ce problème.

L'Analyse Formelle de Concepts est un domaine de recherche très large. Dans ce manuscrit, nous allons nous concentrer essentiellement sur les éléments qui permettent de définir et de construire un treillis de concepts ou bien de générer seulement les concepts. Pour situer le problème, la construction du treillis de concepts et son exploitation représentent la seconde étape du processus de la découverte de connaissances (knowledge discovery) qui comporte trois étapes :

1. Prétraitement des données
2. Construction du treillis
3. Interprétation des résultats

Le prétraitement des données consiste principalement à sélectionner l'ensemble des données à traiter, traiter les données manquantes ou incohérentes et effectuer des transformations de données. Le prétraitement dans notre cas concerne la segmentation et la représentation des symboles et des documents graphiques. Le prétraitement des données et la construction du treillis font l'objet de ce document. L'interprétation des résultats représente un domaine à part et n'est pas décrite dans ce rapport. Elle fait appel aux techniques de visualisation permettant d'afficher les résultats d'une manière compréhensible par l'être humain.

1.2 Contributions

Dans ce travail de thèse, nous proposons plusieurs contributions originales s'inscrivant dans le cadre de la résolution des problèmes de reconnaissance et de localisation des symboles graphiques en contexte. L'originalité des approches proposées réside dans la proposition d'une alliance intéressante entre l'Analyse Formelle de Concepts et la vision par ordinateur. Pour ce faire, nous nous sommes confrontés à l'étude du domaine de l'AFC et plus précisément l'adaptation de la structure du treillis de concepts et son utilisation comme étant l'outil majeur de nos travaux. Notre incursion dans le domaine de l'Analyse Formelle de Concepts et plus précisément notre choix de la structure du *treillis de Galois* appelé aussi *treillis de concepts*, est motivée par les nombreux avantages présentés par cet outil. Le principal avantage du *treillis de concepts* est l'aspect symbolique qu'il offre. Il présente un espace de recherche concis, précis et souple facilitant ainsi la prise de décision. Notre première contribution s'inscrit dans le cadre de la reconnaissance de symboles graphiques.

1.2.1 Reconnaissance d'Objets Graphiques

Classification de Symboles en se basant sur un treillis de Galois

Notre première contribution réside dans l'exploitation de la représentation par sac de mots combinée au treillis de Galois. La communauté de l'analyse des documents a fait émerger l'idée d'intégrer cette structure dans ce cadre. Nous proposons une nouvelle approche pour la reconnaissance de symboles graphiques en combinant un treillis de concepts avec une représentation par sac de mots. Les mots visuels définissent les propriétés représentatives d'un symbole graphique qui seront modélisées dans le treillis de Galois. En effet, l'algorithme de classification est fondé sur le treillis de Galois où les *intensions* de ses concepts représentent des mots visuels. L'utilisation des mots visuels comme des primitives permet d'évaluer le classifieur avec une approche symbolique qui n'a plus besoin de l'étape de discrétisation primordiale pour sa construction. Notre méthode est comparée aux approches classiques, sans sac de mots et à plusieurs classifieurs usuels, évaluées sur différentes bases de symboles. Nous montrons la pertinence et la robustesse de notre proposition pour la classification de symboles graphiques. Cette approche a mis en exergue la nécessité d'allier une approche symbolique avec le formalisme du treillis de concepts dans le cadre de la reconnaissance d'objets graphiques. Cette méthode s'est avérée très performante et a montré une aptitude élevée dans le contexte de la classification. La première approche proposée nous a emmenés à nous poser deux questions fondamentales : *Quel sera l'impact d'une étape de sélection d'attributs du vocabulaire sur les performances du classifieur ?*, *Y- aura t-il aussi un impact sur les performances du treillis ?*

Classification de Symboles en se basant sur un Treillis de Galois Fréquent

Dans une deuxième contribution, nous proposons une approche basée sur une nouvelle méthode à base d'un treillis de concepts de caractéristiques fréquentes que nous appelons *le Treillis de Galois Fréquent* utilisé dans un cadre d'apprentissage supervisé. Notons que cette appellation est différente du principe du treillis proposé dans [Waiyamaï and Lakhal2000]. En effet, le terme "fréquent" est attribué au treillis dans notre cas faisant référence aux types de données qui y sont organisées. Ces données sont les plus pertinentes parmi l'ensemble des caractéristiques extraites à partir de la base des objets graphiques. L'originalité de ce travail réside dans l'idée de construire un treillis de Galois non pas en termes de toutes les caractéristiques visuelles des symboles, mais en termes des caractéristiques les plus pertinentes ou en d'autres termes les mots visuels ayant le pouvoir discriminant le plus élevé après filtrage du vocabulaire. Notre approche consiste premièrement, à créer un vocabulaire visuel, contenant les mots visuels les plus informatifs, à partir de vecteurs caractéristiques définis sur une base d'apprentissage de symboles. Le treillis de Galois obtenu en fonction des motifs fréquents est entraîné et utilisé comme un classificateur. Ceci, nous permet d'aboutir à un prototype nommé *le Treillis de Galois Fréquent* combiné à un dictionnaire visuel et servant comme classificateur supervisé de symboles.

Application à la Classification d'Images

Motivé par les bonnes performances d'une telle structure, nous avons choisi de l'appliquer au contexte de la classification des images. Les évaluations mises en place dans les deux premières contributions déjà exposées, ont permis d'ouvrir la voie à tester le formalisme du treillis de concepts dans un contexte plus général. Pour ce faire et à l'égard des raisons évoquées auparavant, nous avons choisi de l'appliquer dans le cadre de la reconnaissance des images comme application réelle à ce classifieur. La classification des images reste un défi majeur pour la communauté de

vision par ordinateur. Pour nous positionner par rapport aux travaux de la littérature, nous avons choisi de nous comparer aux approches de l'état de l'art qui sont les plus adéquates à notre approche tout en tentant d'utiliser les mêmes protocoles d'expérimentations adoptées dans ces travaux. Cette application montre la généricité de notre système proposé car il suffit d'*instancier* les objets pour pouvoir varier les applications en partant d'une approche de reconnaissance d'objets graphiques segmentés à une approche de classification d'images.

1.2.2 Localisation de Symboles dans les Documents Graphiques

Notre intérêt porté sur ce type d'approches revient au fait que durant les dernières décennies, les contributions proposées dans la littérature tournent autour du problème de la reconnaissance de symboles et laissent de côté le problème de détection de symboles graphiques en contexte. Hors, de nombreuses applications ont montré un besoin croissant de l'indexation automatique de documents et de récupération de l'information dans de grandes collections de données. Certaines de ces applications se basent sur des documents trop volumineux comme par exemple les bases de livres dans les musées ou les bibliothèques, les catalogues de produits ou les services dans le domaine du web ou encore les plans architecturaux dans le domaine de l'architecture.

La première contribution dans ce contexte s'inscrit dans le cadre de l'évaluation de l'impact des étapes de segmentation et de description des documents graphiques sur les résultats de localisation de symboles en contexte. Pour cela, nous avons choisi d'évaluer un ensemble de détecteurs de points d'intérêt en les intégrant dans des approches pixelaires de localisation s'inspirant des travaux au sein du laboratoire dans l'équipe QGAR lors des précédentes thèses [Nguyen *et al.*2009]. Nos objectifs étaient principalement orientés pour répondre aux questions suivantes : *Comment fournir une description fine d'un document graphique ? Quel descripteur permet de représenter les documents graphiques le plus fidèlement possible ?*

Dans ce même contexte et comme nous l'avons largement détaillé, notre objectif est de bénéficier des capacités de la structure du treillis de concepts en l'intégrant dans diverses applications. À l'égard de ces raisons et dans un même contexte, nous avons présenté une approche originale pour la détection de symboles dans leur contexte basée sur la structure du treillis de concepts. Nous avons choisi d'intégrer la structure du treillis de concepts dans le contexte de la localisation puisqu'il représente une structure pouvant organiser les primitives découlant du document d'une façon hiérarchique tout en bénéficiant de l'aspect sémantique offert par un tel outil. Nous avons choisi une alliance intéressante entre les mots visuels caractérisant les documents graphiques et les points d'intérêt qui leur sont associés. Pour satisfaire les objectifs du système de détection de symboles dans les documents, il nous est paru nécessaire de modéliser cette alliance en se basant sur le treillis de concepts et l'intégrer dans une même chaîne de traitement. La combinaison entre ces deux ensembles a pour objectif principal la satisfaction du processus de détection proposé. Dans cette contribution, le treillis de concepts est aussi utilisé comme un outil de sélection de primitives, ensuite une recherche de mots visuels du symbole modèle s'avère nécessaire précédant l'étape de localisation. Une projection des points d'intérêts correspondant aux mots visuels retenus, combinée à une technique de *template matching* aboutit à la localisation des zones du document susceptibles de contenir les occurrences du symbole requête.

1.3 Organisation du mémoire

Nous avons présenté une description des thématiques abordées dans ce manuscrit en décrivant les principes de la recherche d'informations par le contenu, la reconnaissance de formes et la localisation de symboles en contexte ainsi que les motivations de notre incursion dans le domaine

de l'Analyse Formelle de Concepts. Ce manuscrit s'articule autour de trois grandes parties elles-mêmes scindées respectivement en deux et trois chapitres chacune. La première partie intitulée État de l'Art et composée de deux chapitres, introduit quelques techniques de la reconnaissance de formes et de l'Analyse Formelle de Concepts (AFC). La deuxième partie se focalise sur les contributions apportées dans le cadre de la reconnaissance des objets graphiques. La troisième partie concerne la problématique de localisation de symboles. Elle est composée de trois chapitres. Dans ce qui suit, nous présentons une brève description du contenu de ce manuscrit.

1.3.1 Partie 1 : *État de l'Art*

Dans le premier chapitre, nous débutons par une analyse des différentes étapes composant le processus de classification en détaillant les étapes de segmentation et de représentation des primitives caractérisant les objets graphiques. Dans ce cadre, nous distinguons deux types d'approches de description des primitives à savoir les approches statistiques et les approches structurelles. Ensuite, nous présentons un ensemble de classifieurs usuels utilisés en reconnaissance de formes. Nous avons mis l'accent sur les classifieurs qui nous ont servis tout au long de ce travail pour l'évaluation de notre approche et son positionnement par rapport aux autres travaux et qui nous ont parus les plus corrélés à ce que nous proposons.

Comme il est de coutume de positionner les travaux de recherche par rapport à l'existant, le chapitre 2 dresse une panoplie de systèmes basés sur la structure du treillis de concepts et qui sont les plus corrélés à nos travaux. Nous proposons une introduction formelle du treillis de concepts suivie d'une description détaillée des aspects algorithmiques utilisés dans les systèmes de la littérature à base du treillis.

1.3.2 Partie 2 : *Reconnaissance de Symboles Graphiques*

Le premier chapitre est structuré comme suit : nous détaillons l'ensemble des étapes de la chaîne de traitement proposée pour la reconnaissance d'objets graphiques. Les résultats du système de reconnaissance proposé sont présentés ainsi qu'une comparaison avec les approches de l'état de l'art. Dans le deuxième chapitre, nous présentons le contexte de la deuxième contribution et nous présentons une vision globale de l'approche proposée basée sur la conception d'un treillis de Galois à base de caractéristiques fréquentes. Nous décrivons la stratégie adoptée pour renforcer le critère discriminatif du vocabulaire visuel construit à partir de la base de symboles et son impact sur les taux de classification obtenu par le treillis de concepts.

Dans le troisième chapitre, nous présentons les évaluations mises en place dans ces deux contributions, et qui ont permis d'ouvrir la voie à tester le formalisme du treillis de concepts dans un contexte plus général à savoir la classification des images qui reste un défi majeur pour la communauté de vision par ordinateur.

1.3.3 Partie 3 *Localisation de Symboles dans les Documents Graphiques*

Comme nous venons de l'expliquer, un autre axe a été exploré dans ces travaux de thèse à savoir la localisation de symboles dans les documents graphiques. Dans ce contexte, les contributions apportées par ce travail sont les suivantes :

Le premier chapitre sera consacré à l'étude des approches de localisation de symboles en contexte tout en présentant un état de l'art des travaux réalisés ces dernières années. Nous présentons une panoplie de quelques techniques utilisées dans la littérature et les contributions de la communauté pour la résolution du problème de localisation. Nous organisons cet état de l'art en deux

parties différentes où nous décrivons le principe des approches de localisation existantes dans la littérature d'une part structurelles et d'autre part pixellaires, suivie par une discussion et une analyse des approches détaillées.

Dans le deuxième chapitre, nous proposons une évaluation de quelques approches pixelaires de localisation en vue de tester l'impact de l'étape de représentation des documents sur les taux de détection dans le cadre d'un effort continu pour la résolution des systèmes de localisation. Nous effectuons une étude des différents détecteurs de l'état de l'art afin de se prononcer pour la meilleure méthode garantissant de bons taux de localisation. Cette étude a permis non seulement d'obtenir des détecteurs de bonne qualité, mais aussi une meilleure compréhension du fonctionnement du système de localisation et une amélioration des travaux de l'état de l'art par l'amélioration de l'étape cruciale dans la chaîne de localisation à savoir l'étape de représentation des primitives des documents graphiques. Une autre conclusion que nous avons pu retenir est qu'il est difficile de se prononcer pour telle ou telle méthode de segmentation car elles ont toutes des avantages ainsi que des inconvénients et dépendent du contexte de leurs applications.

Notre dernière contribution s'intéresse au problème de localisation en introduisant la structure du treillis de concepts qui s'est avérée performante dans le cadre de la reconnaissance de symboles graphiques. Nous avons choisi de l'intégrer dans le contexte de la localisation puisqu'elle représente une structure pouvant organiser les primitives découlant du document d'une façon hiérarchique tout en bénéficiant de l'aspect sémantique offert par un tel outil. Nous avons choisi une alliance intéressante entre les mots visuels découlant des documents graphiques (plans architecturaux dans notre cas) et les points d'intérêt qui leurs sont associés. La combinaison entre ces deux ensembles a pour objectif principal la résolution du problème de détection proposé.

Première partie

État de l'Art

1

Reconnaissance de Symboles

"La perception est un processus qui fait intervenir nos connaissances préalables pour rassembler et interpréter les stimuli que nos sens enregistrent. Les deux aspects les plus pertinents de la perception en psychologie cognitive concernent la reconnaissance de formes et l'attention.

La reconnaissance de formes consiste à identifier un arrangement complexe de stimuli sensoriel, tel qu'une lettre de l'alphabet, un visage humain ou une scène complexe".

Margaret W. Maltin

Sommaire

1.1	Introduction	13
1.2	Segmentation	15
1.3	Description des primitives	18
1.3.1	Approches Statistiques	18
1.3.2	Discussion	24
1.3.3	Approches Structurelles	25
1.3.4	Discussion	27
1.3.5	Conclusion sur la description des primitives	27
1.4	Apprentissage et Classification	28
1.4.1	Principe de la classification	29
1.4.2	k -plus proche voisin	30
1.4.3	k -means	30
1.4.4	Arbre de Décision	31
1.4.5	Les machines à vecteur de support	31
1.4.6	Classifieur de Bayes	32
1.4.7	Conclusion	33

1.1 Introduction

Le succès de la fouille de données et des techniques de recherche d'informations dans des bases de données structurées et semi-structurées, a récemment émergé dans le domaine de vision

par ordinateur, y compris la localisation des objets, la recherche, la classification et la reconnaissance de symboles graphiques. Comme nous l'avons développé dans l'introduction générale, durant nos travaux de thèse nous proposons d'apporter plusieurs contributions pour la résolution de divers problèmes de reconnaissance et de détection de symboles en contexte. Notre objectif dans cette partie sur l'état de l'art est mené avec le souci de mettre en relief les thématiques qui ont été abordées tout au long de ce travail de thèse. Ces domaines sont la reconnaissance d'objets graphiques d'une part (ce chapitre), et le domaine des treillis de concepts qui s'inscrit dans le cadre de l'Analyse Formelle de Concepts AFC d'autre part (chapitre 2).

Fruit de plusieurs décennies de recherche, la reconnaissance de formes est, aujourd'hui, un domaine très intéressant et utile pour le traitement de tâches complexes et pour la prise de décisions dans des systèmes d'intelligence artificielle comme par exemple, la recherche d'images par le contenu [Smeulders *et al.*2000, Berretti *et al.*2001, Veltkamp *et al.*2001], la reconnaissance de visages [Liu *et al.*2009, Zhao *et al.*2003] et la localisation de symboles dans des documents graphiques [Nguyen *et al.*2009], [Zuwala and Tabbone2006].

La reconnaissance de formes est aujourd'hui devenue une partie intégrante de la plupart des systèmes d'intelligence artificielle, citons à titre d'exemple le domaine de l'ingénierie, la musique, l'architecture... Ces domaines possèdent un riche patrimoine de documents graphiques ce qui a fait émerger des besoins urgents de systèmes de reconnaissance de symboles qui sont devenus le centre d'intérêt de la communauté de l'analyse de documents.

Dans la littérature, plusieurs travaux décrivent la problématique de la reconnaissance d'images, et plus particulièrement de symboles issus de documents graphiques : [Tombre and Chhabra1998], [Cordella and Vento2000], [Blostein and Kwon2002], [Ogier and Trupin2003], [Lladós and Kwon2004].

La reconnaissance de symboles est une tâche importante dans de nombreuses applications réelles et elle a connu un bond en avant depuis une dizaine d'années. Cela comprend trois étapes :

1. Une étape de segmentation qui consiste à isoler la forme à reconnaître ou à se focaliser sur des zones d'intérêt quand les symboles ne sont pas isolés.
2. Une étape de représentation/description de la forme segmentée.
3. Une étape de classification tirant bénéfice des étapes précédentes et qui consiste à assigner une classe à un nouvel objet en se basant sur ses caractéristiques déjà extraites.

Un choix d'une bonne technique de représentation de symboles a une influence directe sur les résultats d'un système de reconnaissance ou de classification. Un descripteur de formes doit garantir une bonne représentation des primitives issues de l'image et il est préférable que le descripteur ait un pouvoir discriminant assez élevé. Dans la littérature, il existe principalement deux catégories de descripteurs à savoir les approches statistiques et les approches structurelles. Une étude assez détaillée de ces deux types de descripteurs sera présentée dans la section suivante.

Une fois l'étape de description des caractéristiques des symboles graphiques établie, nous passons à l'étape de classification ou de reconnaissance. La classification est une des tâches centrales de l'étape de la fouille de données dans le processus d'extraction de connaissances dans les bases de données. Le problème de la classification est traité dans plusieurs communautés de recherche qui se découvrent et s'enrichissent mutuellement : statistiques, reconnaissances de formes, apprentissage automatique, réseaux de neurones et raisonnement à base de cas. Le terme classification en français désigne à la fois les termes anglais classification (classification supervisée) et clustering (classification non supervisée). Nous nous focalisons sur la classification supervisée qui est un processus comprenant deux phases : apprentissage et classement. La phase d'apprentissage consiste à construire un modèle (ou classifieur) qui décrit un ensemble prédéterminé de classes d'exemples. La phase de classement consiste à utiliser le modèle pour assigner une classe à un nouvel exemple. Dans la littérature, de nombreuses méthodes de classification supervisée s'ap-

puient sur des classifieurs usuels comme par exemple les réseaux bayésiens, le k -plus proches voisins, arbres de décision, réseaux de neurones ou treillis de concepts.

Dans la suite de ce chapitre, nous débutons la première partie par une analyse des différentes étapes composant le processus de reconnaissance de symboles en détaillant les étapes de segmentation et de représentation des primitives caractérisant les objets. Dans ce cadre, nous distinguons deux types d'approches de description des primitives des objets à savoir les approches statistiques et les approches structurelles. Ensuite, nous présentons un ensemble de classifieurs usuels utilisés en reconnaissance de formes. Nous avons mis l'accent sur les classifieurs qui nous ont servi tout au long de ce travail pour l'évaluation de notre approche et son positionnement par rapport aux autres travaux. Parmi les classifieurs usuels de l'état de l'art, nous présentons l'arbre de décision, les k -plus proches voisins, les machines à vecteur de support et le réseau Bayésien.

1.2 Segmentation

L'analyse de la littérature a montré que l'étape de segmentation des objets graphiques est primordiale dans les systèmes de reconnaissance. Il s'agit d'une étape critique dans le processus de la classification possédant un impact direct sur les performances des classifieurs. Pour atteindre des performances de classification acceptables, il est nécessaire d'avoir recours à une étape de description des primitives assez robuste. Un constat global que nous pouvons faire est que malgré la prolifération des méthodes de représentation des primitives, cette étape reste un problème ouvert.

Dans ce qui suit, nous présentons quelques approches de représentation utilisées dans la littérature que nous avons utilisées durant nos travaux de thèse soit à titre de comparaison soit comme méthode de base des approches proposées.

L'objectif de l'étape de segmentation d'une image consiste à la décomposer en régions homogènes ou à extraire les contours/squelettes d'un symbole par exemple. Une région sera considérée comme homogène si elle a des caractéristiques communes par exemple. Pour segmenter une image, des techniques de segmentation ou de squelettisation, plus proches de l'analyse d'images sont également applicables pour extraire des informations des données. Commençons par un panorama de quelques techniques de segmentation les plus répandues dans la littérature.

Contours Actifs

Le principe des contours actifs [Kass *et al.*1988] consiste à déplacer les points pour les rapprocher des zones de fort gradient tout en conservant des caractéristiques comme la courbure du contour ou la répartition des points sur le contour ou d'autres contraintes liées à la disposition des points. Au démarrage de l'algorithme, le contour actif est disposé uniformément autour de l'objet à détecter puis il va se rétracter pour en épouser au mieux ses formes (voir l'exemple dans la Figure.1.1).

Détection de contours : Opérateur Laplacien

Les opérateurs de gradient définissent le contour dans une image comme étant le maximum du gradient dans la direction orthogonale au contour. Tandis que le passage par zéro de la dérivée seconde d'une rupture d'intensité permet également de mettre en relief le contour. Le calcul de Laplacien permet de déterminer la dérivée seconde comme suit :

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial x} \frac{\partial f}{\partial x} + \frac{\partial}{\partial y} \frac{\partial f}{\partial y} \quad (1.1)$$

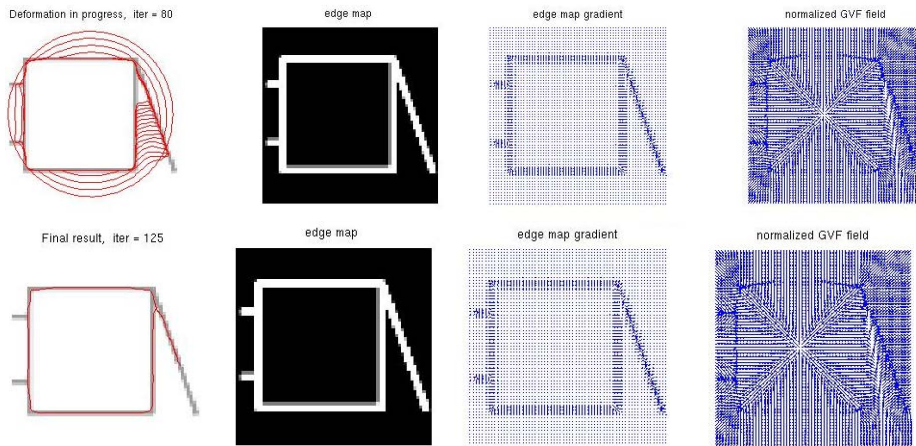


FIGURE 1.1 – Technique de segmentation de symboles basée sur les contours actifs : un contour actif peut se dilater et tenter de remplir une forme, il sera alors situé à l’intérieur de celle-ci au démarrage de l’algorithme. À chaque itération, l’algorithme va tenter de trouver un meilleur positionnement pour le contour pour minimiser les dérivées par rapport aux contraintes utilisées. L’algorithme s’arrêtera lorsqu’il ne sera plus possible d’améliorer le positionnement ou simplement quand le nombre maximum d’itérations aura été atteint.

qui peut s’écrire ainsi :

$$\nabla^2 f = \nabla_x(f(x + 1, y) - f(x, y)) + \nabla_y(f(x, y + 1) - f(x, y)) \quad (1.2)$$

Le laplacien est alors défini comme suit :

$$\nabla^2 f = (f(x + 1, y) - f(x - 1, y)) + f(x, y + 1) - f(x, y - 1) - 4f(x, y) \quad (1.3)$$

Un filtrage avec les masques 3*3 permet d’appliquer l’opération de calcul de Laplacien à une image.

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Une fois que l’image est filtrée au moyen de ce filtre, les passages par zéro sont détectés en gardant uniquement les passages les plus marqués. En effet, la double dérivation peut rendre la technique particulièrement sensible au bruit. Une solution pour remédier à ce problème est de ne pas considérer le bruit, traduit généralement par des oscillations autour de zéro, comme un contour. Un seuil S peut être utilisé afin de ne prendre en compte que les passages par zéro d’amplitude relativement élevée correspondant à des vrais contours de l’image (voir Figure.1.2).

Squelettisation

Pour permettre la squelettisation, l’image doit tout d’abord être binarisée. Ensuite, une liste d’opérateurs morphologiques d’érosion sont appliqués pour l’amincissement séquentiel de l’image par l’élément structurant jusqu’à idempotence. Le squelette est une représentation synthétique de la forme qui fournit des segments de largeur unitaire. Parmi les diverses méthodes existantes pour extraire un squelette, seules celles capables de travailler en 8-connexité sont susceptibles de fournir correctement les points d’intersection entre deux segments [Zhang and Wang1994].

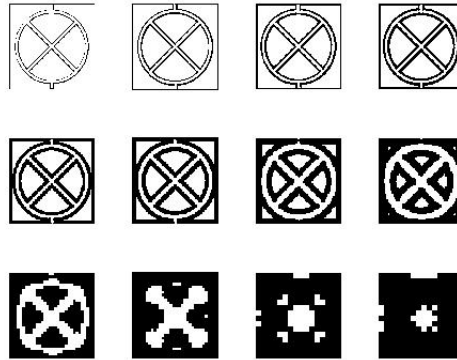


FIGURE 1.2 – Opérateur Laplacien pour la détection des contours à différentes tailles de masque.

Malheureusement, le squelette, est très sensible au bruit (déformation du contour, présence d'un trou, ...). Des exemples d'extraction de squelettes des symboles sont présentés dans la Figure.1.3. Après ce survol de techniques de segmentation des images les plus répandues dans la littérature, passons maintenant à l'étape primordiale dans la chaîne de traitement du processus de la reconnaissance à savoir l'étape de description des primitives résultantes de l'étape de segmentation. Dans ce qui suit, nous allons présenter un ensemble d'approches de représentation des primitives.

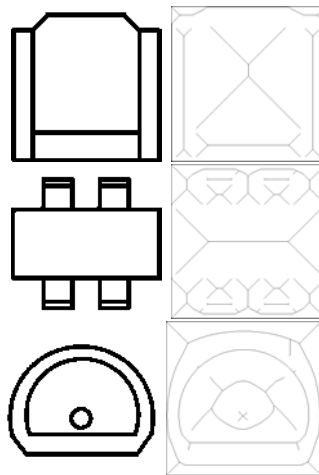


FIGURE 1.3 – La squelettisation de symboles graphiques.

Points d'intérêt

Dans la littérature, l'étape d'extraction des caractéristiques se base le plus souvent sur des détecteurs de points d'intérêt classiques où un descripteur local est calculé au voisinage de ces points. Des études récentes montrent la robustesse de ces méthodes dans les applications de traitement de symboles graphiques [Mikolajczyk and Schmid2004]. Récemment, il y a une tendance à l'utilisation des points clés ou les points d'intérêt locaux pour la recherche de symboles et pour la classification. Les points d'intérêt sont les primitives saillantes qui contiennent l'information

locale riche d'un symbole quelconque, et peuvent être détectés automatiquement en utilisant différents détecteurs. Pour cela, nous avons expérimenté plusieurs algorithmes de détection de points dans la dernière partie de ce manuscrit afin de se prononcer pour telle ou telle technique et pouvoir l'adapter dans notre contexte. Pour plus de détails sur ces approches, le lecteur peut se rendre au chapitre.3.5.

1.3 Description des primitives

Dans le cadre de la représentation des primitives des objets graphiques, nous distinguons deux types d'approches à savoir *les approches statistiques* et *les approches structurelles*.

1.3.1 Approches Statistiques

Une signature *statistique* est une description de l'objet en fonction de son contenu pixelaire. Concrètement, un vecteur de caractéristiques (un ensemble de valeurs numériques) est calculé en se basant sur certaines propriétés de l'objet. Un objet est alors représenté par un vecteur x de dimension n : $x=(x_1, x_2, \dots, x_n) \in R^n$.

Généralement, il existe deux catégories de descripteurs de formes : les descripteurs de formes à base de *contour* et les descripteurs de forme à base de *région*.

Parmi les descripteurs de forme basés sur les contours, nous pouvons citer les descripteurs de Fourier [Hu1962, Kauppinen *et al.*1995, Zahn and Roskies1972]. Ces descripteurs sont limités au fait qu'ils se basent uniquement sur l'information contenue dans le contour de l'image.

Néanmoins, les moments géométriques comme les moments de Legendre, les moments de Zernike représentent des descripteurs de formes à base de régions. Ces descripteurs exploitent l'information contenue dans la région interne de l'objet plutôt que de se contenter de l'information dans des points restreints de l'image.

Signature de Fourier-Mellin

Les applications basées sur la signature de Fourier sont nombreuses comme par exemple la recherche et la reconstruction de l'image [Derrode and Ghorbel2001], l'estimation du mouvement ou encore la reconnaissance de caractères de différentes tailles. La transformée de Fourier-Mellin est définie pour une image qui est décrite par une fonction réelle et positive $f(r, \theta)$ en coordonnées polaires comme suit :

$$\mathcal{M}_{pq} = \int_{r=0}^{+\infty} r^{p-1} e^{-iq\theta} f(r, \theta) dr d\rho \quad (1.4)$$

$$p = \sigma + iv \in C, q \in Z \quad (1.5)$$

Les invariants de Fourier-Mellin sont donnés par la formule suivante :

$$I_{pq} = \mathcal{M}_{pq}[\mathcal{M}_{0\theta}]^{-\frac{p}{\theta}} [\mathcal{M}_{1\theta}]^{-q} [\mathcal{M}_{1\theta}]^q \quad (1.6)$$

avec p un coefficient radial et q un coefficient circulaire. Ces moments sont invariants à la translation, à la rotation et au changement d'échelle.

Moments de Zernike

Ces moments reposent sur l'utilisation des polynômes orthogonaux et ont été mis en place par [Teh and Chin1988]. Les moments de Zernike sont calculés comme suit :

$$\mathbf{V}_{nm}(x, y) = \mathbf{V}_{nm}(\rho \cos(\theta), \rho \sin(\theta)) = \mathbf{R}_{nm}(\rho) \exp(jm\theta) \quad (1.7)$$

et

$$\mathbf{R}_{nm}(\rho) = \sum_{s=0}^{\frac{(n-|m|)}{2}} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s} \quad (1.8)$$

avec ρ est la longueur du vecteur partant de l'origine vers le pixel (x, y) , θ est l'angle entre ce vecteur et l'axe des abscisses dans le sens inverse des aiguilles d'une montre, n et m sont des entiers :

$$x^2 + y^2 = 1 \quad (1.9)$$

$$x = r \cos(\theta) \quad (1.10)$$

$$y = r \sin(\theta) \quad (1.11)$$

Signature à base d'ondelettes

La transformée en ondelettes est un outil mathématique particulièrement performant dans le domaine de la reconnaissance de formes. La transformée en ondelettes discrète consiste à caractériser une fonction de carré intégrable par ses produits scalaires avec une famille dénombrable de fonctions de carré intégrable : les ondelettes.

Il existe deux catégories d'ondelettes à savoir : les fonctions d'ondelette appelées *ondelettes mères* notées $\psi(x)$ et les fonctions d'échelle appelées *ondelettes pères* notées $\phi(x)$. La dilatation et la translation des *ondelettes mères* et des *ondelettes pères* entraînent respectivement *les ondelettes filles et fils*. La mathématicienne *Ingrid Daubechies* a défini deux contraintes sur *les ondelettes de Daubechies* [Daubechies1992] à savoir (a) l'orthogonalité de la base d'ondelettes et (b) la compacité du support de l'*ondelette-mère*. Ces ondelettes sont aussi des fonctions à n moments nuls.

$$\forall k \in \{0, \dots, n\} \langle f | t^k \rangle = \int t^k f(t) dt = 0 \quad (1.12)$$

En 1909 Alfred Haar a développé la première ondelette qui fait partie du groupe *d'ondelettes de Daubechies*, dont les fonctions d'échelle vérifient des propriétés intéressantes.

$$\Phi_{Haar}(x) = \begin{cases} 1 & \text{si } 0 \leq x < 1 \\ 0 & \text{sinon} \end{cases}$$

$$\Psi_{Haar}(x) = \begin{cases} 1 & \text{si } 0 \leq x < \frac{1}{2} \\ -1 & \text{si } \frac{1}{2} \leq x < 1 \\ 0 & \text{sinon} \end{cases} \quad (1.13)$$

Signature de Radon

La transformée de Radon a été introduite par Johann Radon (1917), qui a également fourni une formule pour la transformée inverse. En outre, Radon propose des formules dans trois dimensions pour la transformée. Cette transformée est une représentation de la forme par un ensemble de projections suivant différentes orientations. La transformée de Radon consiste à projeter

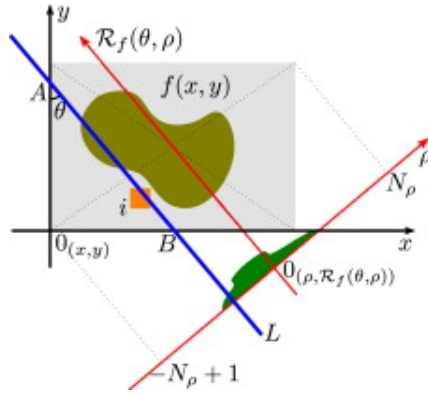


FIGURE 1.4 – Illustration graphique de la transformée de Radon d’une fonction $f(x, y)$. La transformée de Radon est une transformation de l’image de l’espace (x, y) à l’espace des paramètres (θ, ρ) et peut être représentée mathématiquement par une droite intégrante de (x, y) tout le long de toute les lignes L paramétrées par (θ, ρ) et représentées dans l’espace de l’image (x, y) .

l’image sur une droite ($\Delta : \rho = x\cos(\theta) + y\sin(\theta)$) avec ρ est la distance entre Δ et l’origine du repère et θ est l’angle entre l’axe des abscisses et la normale à Δ (voir Figure.1.4).

Soit $f(x, y)$ une image, la transformée de Radon est définie par la double intégrale suivante :

$$T_{Rf}(\rho, \theta) = \iint_{x,y} f(x, y)\delta(x\cos\theta + y\sin\theta - \rho)dx dy \quad (1.14)$$

avec $-\infty < \rho < \infty$, $0 \leq \theta \leq \pi$ et δ correspond à la fonction Dirac telle que $\rho(x)=1$ si $x=0$ et 0 sinon. Intuitivement cela revient à intégrer la fonction f le long d’une droite pour tout l’espace de paramètres (ρ, θ) . Dans [Tabbone *et al.*2006], les auteurs introduisent une adaptation originale de la transformée (dite *R-signature*) afin de définir un descripteur invariant à la rotation, à la translation et au changement d’échelle. La *R-signature* consiste à convertir la transformée de Radon 2D en 1D par une intégration des informations radiales selon les coordonnées angulaires (voir Figure.1.5).

$$\mathcal{R}_f(\theta) = \int_{-\infty}^{+\infty} T_{Rf}^2(\rho, \theta)d\rho \quad (1.15)$$

Cette transformée est invariante à la translation, aux changements d’échelle et à la rotation si les permutations sont incluses dans son calcul de distance (voir Figure.1.5). Afin de corriger le déphasage dans l’espace de Radon qui peut être engendré suite à la rotation de l’objet, il faut intégrer dans la mesure de distance entre deux signatures, la comparaison avec les différentes permutations possibles des valeurs de la signature cela revient en d’autres termes à réaliser une permutation cyclique de la signature. Une autre solution proposée afin de garantir l’invariance de la transformée de Radon à la rotation, est de calculer sa transformée de Fourier.

Descripteur Générique de Fourier

Le principe de ce descripteur consiste à échantillonner le contour sur un nombre de points préfixé afin de calculer la Transformée de Fourier Discrète (voir Figure.1.6). Comme première étape, l’image d’entrée $f(r, \theta)$ où :

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}, \theta = \arctan \frac{y - y_c}{x - x_c} \quad (1.16)$$

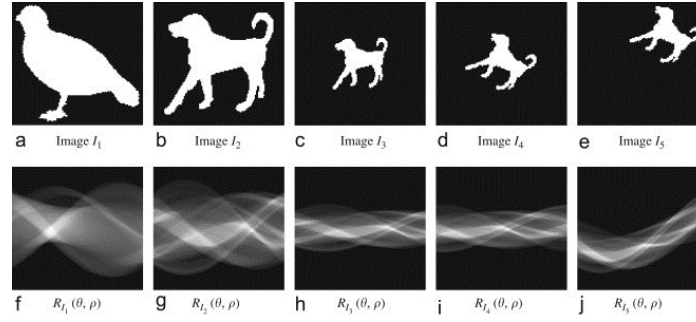


FIGURE 1.5 – Définition de la transformée de Radon pour plusieurs images de différentes échelles et orientations (extrait à partir de [Tabbone *et al.*2006]).

où (x_c, y_c) sont les coordonnées du centre de gravité. Ensuite, une transformation de l'image polaire échantillonnée en trame dans le système des coordonnées cartésiennes est réalisée. Ces images sont obtenues par un échantillonnage circulaire de l'objet/surface dans l'image. Alors une transformation de Fourier 2D sur l'image transformée est calculée :

$$FP_2(\rho, \phi) = \sum_r \sum_i f(r, \theta_i) \exp[-j2\pi(\frac{r}{R}\rho + \frac{i}{T}\phi)] \quad (1.17)$$

où $0 \leq \rho < R$ et $\theta_i = i(\frac{2\pi}{T})$ ($0 \leq i < T$); $0 \leq \rho < R$, $0 \leq \phi < T$. R et T sont respectivement, les résolutions radiale et angulaire. Les coefficients fréquentiels normalisés ainsi obtenus forment les descripteurs. Il est alors possible de calculer des descripteurs selon la courbure ou selon la distance au centre de gravité. D.Zhang et G.Lu ont montré que cette seconde solution est plus performante [Zhang and Lu2003]. Le contour étant normalisé à M points, la distance au centre de gravité est établie pour chaque point (équation.1.18). Seul le module est conservé vu son invariance en rotation (équation.1.19). Le vecteur d'attribut conservé a ses coefficients normalisés par $|F_0|$ pour s'affranchir aux changements d'échelle (équation.1.20).

$$R(t) = \sqrt{(x(t) - x_c(t))^2 + (y(t) - y_c(t))^2} \quad (1.18)$$

$$F_i = \frac{1}{M} \sum_{t=0}^{M-1} R(t) \exp(\frac{-2j\pi it}{M}), i = 0, 1, \dots, M-1 \quad (1.19)$$

$$f_R = \frac{|F_1|}{|F_0|} \quad \frac{|F_2|}{|F_0|} \quad \dots \quad \frac{|F_{\frac{M}{2}}|}{|F_0|} \quad (1.20)$$

ART (Angular Radium Transform)

Le descripteur ART [Fang and Qiu2003] est un descripteur de forme basé sur la région de la norme MPEG-7. L'ART est un complexe de transformation orthogonale unitaire défini sur une unité de disque qui se compose de la sinusoïde complète orthogonale des fonctions de base en coordonnées polaires. Les coefficients du descripteur ART, F_{nm} d'ordre n et m , sont définis par :

$$F_{nm} = \int_0^{2\pi} \int_0^1 V_{nm}(\rho, \theta) f(\rho, \theta) \rho d\rho d\theta \quad (1.21)$$

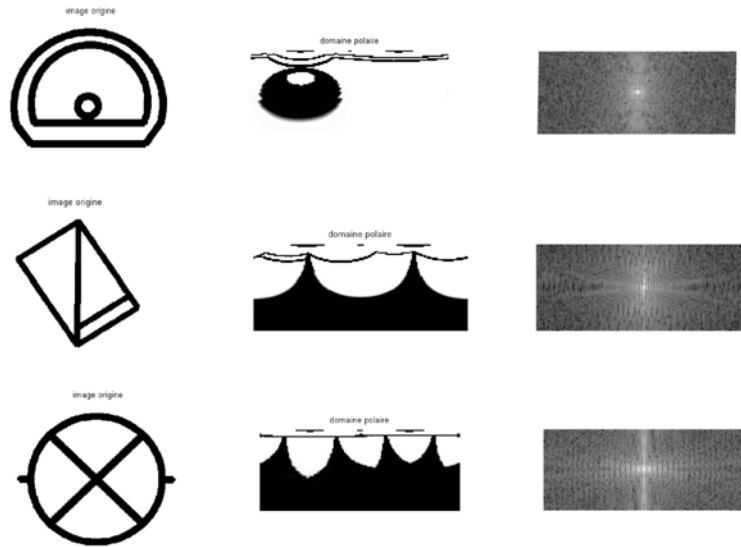


FIGURE 1.6 – Descripteur Générique de Fourier : la première image est l’image du symbole d’origine, la deuxième étape décrit le passage de l’image et la transformée de Fourier dans le domaine polaire et la troisième étape décrit le descripteur générique de Fourier.

où $f(\rho, \theta)$ est une fonction en coordonnées polaires et $v_{nm}(\rho, \theta)$ est la fonction de base de ART qui sont séparables le long des directions radiales et angulaires :

$$V_{nm}(\rho, \theta) = A_m(\theta)R_n(\rho) \tag{1.22}$$

Afin d’atteindre l’invariance par rotation, une fonction exponentielle est utilisée pour la fonction de base angulaire. La fonction radiale de base est définie par une fonction cosinus :

$$\begin{cases} A_m(\theta) = \frac{1}{2\pi} \exp(jm\theta) \\ R_n(\rho) = \begin{cases} 1 & n = 0 \\ 2\cos(\pi n\rho) & n \neq 0 \end{cases} \end{cases} \tag{1.23}$$

Le descripteur ART est défini comme un ensemble de niveaux normalisés de l’ensemble de

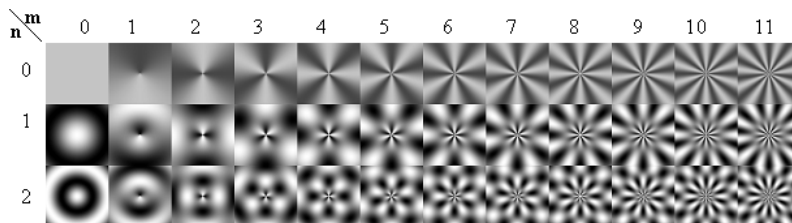


FIGURE 1.7 – Parties réelles de la fonction de base de ART (extrait à partir de [Ricard *et al.*2004]).

coefficients ART. L’invariance par rotation est obtenue à l’aide de la grandeur des coefficients. Pour les coefficients de normalisation échelle ART, sont divisées par la valeur du coefficient ART d’ordre $n = 0, m = 0$. Pour diminuer la taille du descripteur, la quantification est appliquée

à chaque coefficient en utilisant quatre bits par coefficient. Comme proposé dans MPEG-7, la distance entre deux formes décrites par le descripteur ART est calculée en utilisant la Norme L1 :

$$d_{ART}(Q, I) = \sum_{k=0}^{n.m} \|ART_Q[i] - ART_I[i]\| \quad (1.24)$$

Contexte de Forme

L'idée de base des contextes de forme est illustrée dans la Figure.1.8. Une forme est représentée par un ensemble discret de points échantillonnés à partir de la forme interne et des contours externes de la forme. Ceux-ci peuvent être obtenus sous forme de positions des points de contour, donnant ainsi un ensemble de points $IP = \{p_1, \dots, p_n\}$, $p_i \in \mathbb{R}^2$, de n points. Le contexte de forme est obtenu par la distribution des points de contour dans la région d'un point $p_i \in IP$. Les

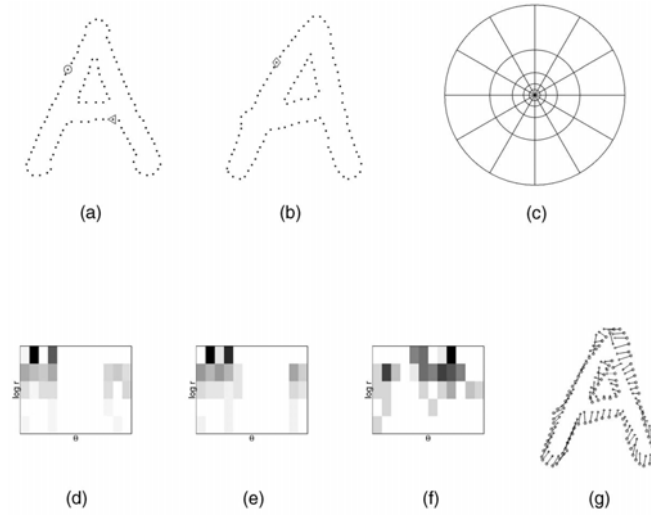


FIGURE 1.8 – Les étapes de calcul du contexte de forme. (a) et (b) Représentation des échantillons des points de contours de deux formes. (c) Schéma d'histogrammes log-polaires utilisés dans le calcul des contextes de forme. 5 plages sont utilisés pour $\log r$ et 12 plages pour θ . (d), (e) et (f) Contextes de forme des points marqués par \circ et \triangleleft en (a) et par \diamond en (b) (reprise de [Belongie *et al.*2002]).

coordonnées des $n-1$ autres points sont déterminées dans un système de coordonnées log-polaire utilisant p_i comme origine :

$$q = (\log(r_q), \theta_q), \forall q \neq p_i \wedge q \in IP \quad (1.25)$$

où r_q représente la distance entre q et p_i , θ_q est l'angle entre $\overrightarrow{p_i q}$ et l'axe horizontal. Le contexte de forme h_i du point p_i est défini par :

$$h_i^k = q \neq p_i : (q_{p_i}) \in \text{bin}(k) \quad (1.26)$$

où h_i^k est le nombre de points de contours de la $k^{\text{ième}}$ classe de l'histogramme et $\text{bin}(k) = \{(r_q, \theta_q) : r_q \in [r_k, r_k + \Delta r_k] \wedge \theta_q \in [\theta_k, \theta_k + \Delta \theta_k]\}$

Descripteur SIFT

Dans nos travaux, nous avons eu recours au descripteur SIFT (Scale Invariant Feature Transform) [Lowe2004]. Son principe consiste à calculer un histogramme d'orientations de gradients. Plus précisément, le voisinage d'un patch P d'une image I est décomposé en 16 blocs de 4*4 pixels. Ensuite, un histogramme d'orientation de gradients est formé au niveau de chaque bloc en discrétisant l'orientation en 8 bins. La détermination de l'histogramme est effectuée comme suit :

$$H[\Theta^i(x, y)] = H[\Theta^i(x, y)] + \|\vec{G}(x, y)\| \cdot M_\sigma(x, y) \quad (1.27)$$

où $\Theta^i(x, y)$ est l'orientation du gradient discrétisée au pixel (x, y) , $\|\vec{G}(x, y)\|$ est le module du gradient au pixel (x, y) et $M_\sigma(x, y)$ est la pondération résultante au niveau du pixel (x, y) de l'application d'un masque Gaussien centré sur le point d'intérêt et d'écart type σ .

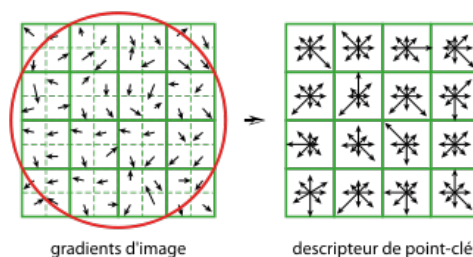


FIGURE 1.9 – Calcul du descripteur SIFT.

Descripteur SURF

Le descripteur SURF décrit l'intensité de pixels dans un voisinage autour de chaque point d'intérêt. La réponse en x et en y des ondelettes de Haar est calculée dans un voisinage de 6s où s est l'échelle où le point d'intérêt a été trouvée. À partir de ces valeurs, l'orientation dominante de chaque point d'intérêt est calculée en faisant glisser une fenêtre d'orientation. Pour calculer le descripteur, un carré de taille 20s orienté selon l'orientation dominante est extrait. Cette zone est subdivisée en 4x4 carrés. Pour chacune de ces sous-régions, les ondelettes de Haar sont calculées sur 5x5 points. Soit dx et dy la réponse à l'ondelette de Haar, 4 valeurs sont calculées pour chacune des sous-régions : $\sum dx$, $\sum dy$, $\sum |dx|$ et $\sum |dy|$ (voir Figure.1.10). Au final, chacun des points extraits à l'étape précédente est décrit par un vecteur composé de 4x4x4 valeurs soit 64 dimensions.

1.3.2 Discussion

Ce panorama de descripteurs statistiques montre que la mise en place de ce type de représentations est facile car elles sont calculées directement sur l'image. La plupart de ces descripteurs sont invariants aux transformations géométriques (rotation, échelle et translation) et robustes aux occlusions qu'un symbole peut subir. Un constat global est qu'aucun descripteur ne peut fournir une solution miracle pour les problèmes de la communauté de vision. Pour cela, il nous a paru nécessaire de les adapter au contexte de leur utilisation afin de répondre à un objectif bien déterminé.

Dans le cadre de cette thèse, nous avons utilisé plusieurs descripteurs statistiques soit dans le cadre de nos approches proposées comme par exemple le descripteur SIFT, soit dans le cadre de l'évaluation et de comparaison comme les moments de Zernike, le descripteur de Fourier, le

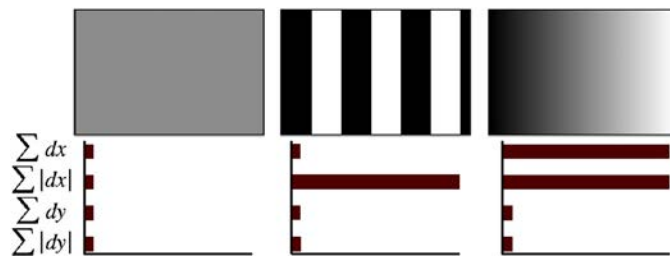


FIGURE 1.10 – À gauche : dans le cas d’une région homogène, toutes les valeurs sont relativement faibles. Au milieu : en présence de fréquences dans la direction x , la valeur de $|dx|$ est élevée, mais les autres valeurs restent faibles. Si l’intensité augmente progressivement dans la direction x , les deux valeurs $\sum dx$ et $|dx|$ sont élevées (cet exemple est extrait de [Bay *et al.*2006]).

contexte de forme et la R -signature car ils se sont avérés performants. Passons maintenant au deuxième type d’approches de description des primitives à savoir *les approches structurelles*.

1.3.3 Approches Structurelles

Dans les approches structurelles, les symboles graphiques sont décomposés en un ensemble de primitives élémentaires comme une région (composante connexe, région de voronoi...), un segment, un arc de cercle, vecteurs... Il s’agit d’une représentation de la *structure* de la forme. Les approches de reconnaissance basées sur l’utilisation des signatures structurelles intègrent des approches syntaxiques basées sur la théorie du langage formel [Fu *et al.*2004a] et les approches basées sur les graphes. La description des relations entre ces différentes primitives a fréquemment la forme des graphes ou encore des approches syntaxiques basées sur l’utilisation de grammaires. Dans la théorie du langage formel, une grammaire \mathcal{G} est constituée d’un alphabet d’éléments terminaux (éléments élémentaires d’une grammaire) et un alphabet d’éléments non terminaux (ensemble d’éléments composés des éléments terminaux). Par analogie à la grammaire qui représente une phrase par une suite de mots, une forme est composée par un ensemble de primitives (voir l’exemple présenté dans la Figure.1.11). L’utilisation des approches syntaxiques s’avèrent

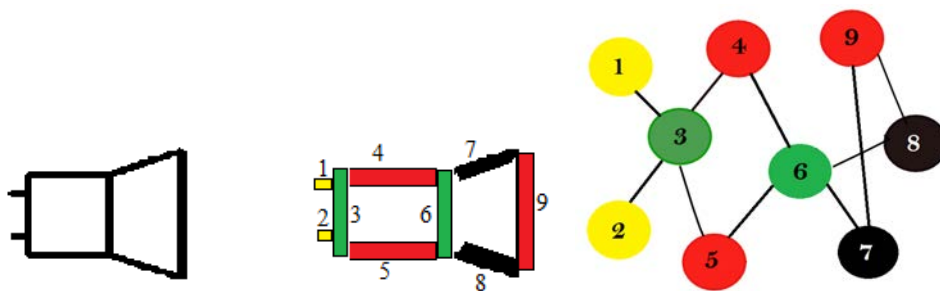


FIGURE 1.11 – Représentation d’une image sous forme de graphe.

peu utilisables vu la capacité limitée d’expression et leur non-adaptation pour l’apprentissage [Tanaka1995]. En outre, pour pouvoir définir les primitives de la grammaire, une connaissance a priori sur la base de données est exigée. Contrairement à ces approches, les approches basées sur les graphes sont les plus utilisées vu qu’elles ne nécessitent pas une étape de définition préalable de primitives. Parmi les structures de données, les graphes sont généralement utilisés pour la description structurelle des symboles. Les graphes sont particulièrement bien adaptés pour re-

présenter les informations structurelles et les relations entre primitives. Ils peuvent représenter des simples liaisons entre des points d'intérêt du symbole ou aussi des graphes d'adjacence entre les lignes, mais peuvent aussi décrire les relations spatiales entre primitives. Parmi les signatures structurelles sous forme de graphes, nous distinguons :

- le graphe de vecteurs : dans ces approches, les *nœuds* du graphe sont des vecteurs (arcs de cercle, segment,...) et les *arêtes* sont les relations topologiques entre ces vecteurs.
- le graphe de régions : dans ces approches, les *nœuds* sont des régions et les *arêtes* sont les relations topologiques entre ces régions.

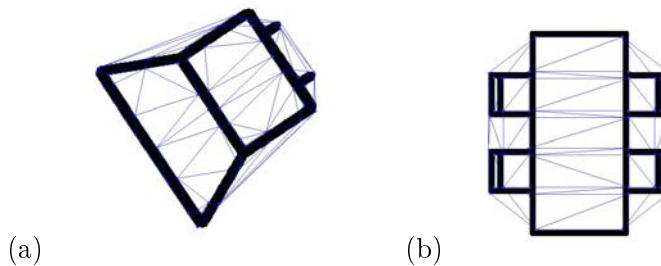


FIGURE 1.12 – Graphe de voisinage des points d'intérêt d'un symbole graphique.

Certains types de graphes comme les graphes attribués (Attributed Relational Graphs ARGs) utilisés dans [Luqman *et al.*2010], représentent un bon support pour la représentation structurelle de symboles graphiques : les nœuds et les arêtes, avec leurs attributs associés, décrivant respectivement les composants d'intérêt (les régions, les points, les arcs de cercle, les segments, ...) ainsi que les interconnexions entre eux (voir un exemple dans la Figure.1.12 qui illustre un graphe de voisinage décrivant les interconnexions entre les points d'intérêt d'une image). Comme approche structurelle, M. Delalandre [Delalandre *et al.*2004] propose de décomposer une image correspondant à un symbole en graphe de lignes (voir Figure.1.13). Les relations entre les segments de l'image sont de deux types : parallélisme ou chevauchement. [Fonseca *et al.*2005]

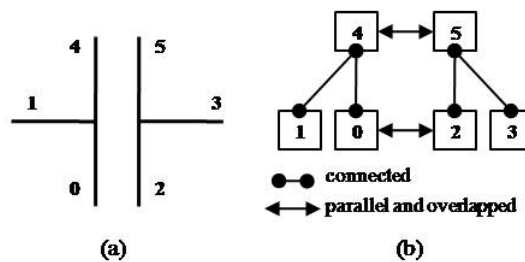


FIGURE 1.13 – Représentation d'une image par un graphe de lignes (reprise de [Delalandre *et al.*2008]).

propose d'utiliser des polygones comme primitives. Un graphe de topologie dont les nœuds sont des polygones et les arcs sont des relations topologiques représente un document technique. Ces relations topologiques garantissent l'invariance à la translation et à la rotation. Trois relations topologiques sont utilisées : la disjonction, l'inclusion et l'adjacence. Dans le graphe de topologie, la relation d'inclusion est représentée par un arc vertical et la relation d'adjacence par un arc

horizontal. Un exemple est illustré dans la Figure.1.14 présentant un document en trois niveau de détails.

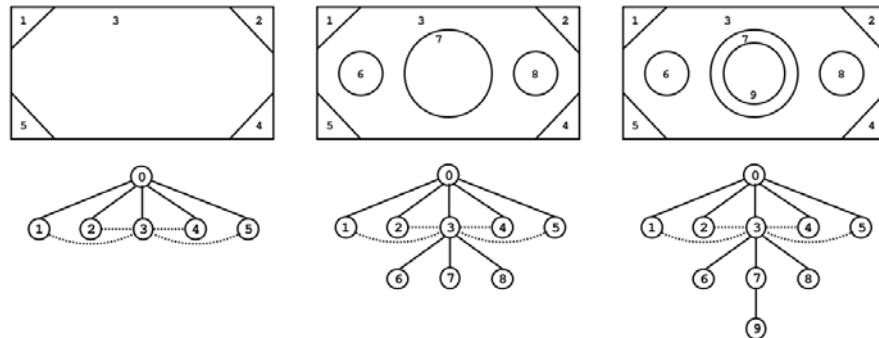


FIGURE 1.14 – Différents niveaux de détail avec leur graphe correspondant (reprise de [Fonseca and Jorge2003]).

1.3.4 Discussion

Un constat global que nous pouvons faire après cette brève étude est que la littérature est riche en techniques de représentation d'objets graphiques. Les défenseurs des *approches statistiques* mettent en avant la simplicité de modèles sur lesquels elles se basent pour décrire des comportements spécifiques. Un autre point qui justifie cet intérêt croissant pour les *approches statistiques* réside surtout dans leurs robustesses aux bruits qu'un symbole peut subir. Elles se basent sur un ou plusieurs attributs de description de forme. Un autre point important est qu'elles offrent une bonne description des objets graphiques qui est déterminante pour atteindre des performances de reconnaissance stables et satisfaisantes.

Les défenseurs des *approches structurelles*, considèrent qu'elles permettent de traiter des objets structurés faisant intervenir des relations complexes entre sous-parties contrairement aux *approches statistiques* exploitant traditionnellement des représentations "plates" (vectorielles). En plus, elles utilisent une décomposition de l'image en objets primitifs et permet de décrire une partie de l'organisation de ces primitives les unes par rapport aux autres et d'exploiter par la suite, des structures de représentation permettant de stocker ces structures comme par exemple les graphes d'adjacences.

1.3.5 Conclusion sur la description des primitives

Dans cet état de l'art, nous avons souhaité présenter les notions concernant la reconnaissance de formes. Les approches de calcul de descripteurs sont nombreuses et il est difficile de se prononcer pour telle ou telle méthode car les performances de ces approches que ce soit les approches statistiques ou structurelles dépendent du contexte de leurs applications. Dans notre approche, nous ne cherchons pas à inventer une nouvelle méthode de description des primitives mais d'adapter les meilleures techniques qui offrent un bon compromis entre la précision des résultats de reconnaissance et la complexité du système mis en place.

Comme nous l'avons déjà indiqué, du fait de la grande diversité des méthodes de représentation des primitives, il nous paraît indispensable de combiner plusieurs approches de description. Les méthodes statistiques sont relativement simples et peu coûteuses en temps de calcul. À l'encontre des méthodes structurelles qui sont plus complexes et plus coûteuses. Au regard des objectifs fixés

dans l'introduction de ce manuscrit, et dans le but de concevoir des systèmes de reconnaissance et de localisation assez robustes, il nous est paru intéressant d'utiliser plusieurs descripteurs car il est difficile de trancher entre telle ou telle méthode. Nous avons donc fait le choix tout au long de cette thèse, a priori de diversifier le choix des descripteurs en vue de rendre la description des images de plus en plus fine.

Les raisons de ce choix sont de deux ordres essentiellement. La première raison résulte du fait que dans certains cas, un seul descripteur ne fournit pas le résultat escompté. Pour pallier à cette problématique, nous voulons bénéficier de la diversification des descripteurs utilisés en fonction du cadre traité allant parfois à une combinaison de plusieurs approches dans une même chaîne de traitement. La deuxième raison concerne le type des symboles utilisés dans nos travaux qui sont très variés, séparés de leur contexte ou inclus dans des documents graphiques complexes. Tout en tentant de bien représenter ces objets, nous proposons d'utiliser les techniques de description des primitives les plus adéquates aux types de graphiques manipulés. Hormis le choix d'un descripteur robuste pour la première étape du processus de reconnaissance, il est nécessaire de choisir un bon classifieur pour garantir de très bonnes performances de reconnaissance. Le choix de la structure qui servira comme classificateur possède aussi un impact sur les performances d'un tel système.

La classification supervisée est une tâche d'extraction de données qui consiste à construire un classificateur à partir d'un ensemble d'échantillons qui sont étiquetés par leur classe (phase d'apprentissage), puis de prédire la classe de nouveaux exemples en se basant sur le classificateur (phase de classification). Dans la section suivante, nous passons en revue quelques types de classificateurs utilisés dans la reconnaissance de formes

1.4 Apprentissage et Classification

Le sujet de la reconnaissance de symboles est un sujet en pleine expansion. Bien que de nombreuses approches ont été proposées durant ces dernières décennies, le problème de classification reste un sujet complexe surtout quand les symboles ne sont pas isolés et ne sont pas connectés à d'autres parties de l'image. Une dualité assez connue dans ce cadre est la suivante : *"reconnaître pour segmenter" ou "segmenter pour reconnaître" ?*.

Le succès de la fouille de données et des techniques de recherche d'informations dans des bases de données structurées et semi structurées, a récemment émergé dans le domaine de vision par ordinateur, y compris la localisation des objets, la recherche, la classification et la reconnaissance de symboles graphiques. Les recherches en reconnaissance de documents graphiques s'intéressent à la localisation et à la classification de formes susceptibles d'apparaître dans des documents graphiques. Tous ces documents sont constitués d'un ensemble fini de formes connues a priori (alphabets de symboles). Ces dernières années, la volonté de produire des systèmes capables d'indexer ou de faire des recherches sur les documents entiers semble se développer. La reconnaissance de symboles graphiques dans ces documents s'appuie sur deux étapes principales à savoir l'extraction et la description des primitives graphiques et la phase de classification.

Une grande variété de descripteurs ont été proposés dans la littérature. Un choix d'une bonne technique de représentation de symboles a une influence directe sur les résultats d'un système de reconnaissance ou de classification. D'autre part, le choix de la structure qui servira comme classificateur est déterminante pour tout problème de reconnaissance. Le choix du classifieur dépend de la méthode avec laquelle est abordé le problème de la reconnaissance. Dans la section suivante, nous présentons succinctement différents classifieurs qui sont les plus adaptés à notre application et que nous avons utilisés à titre de comparaison avec notre système.

1.4.1 Principe de la classification

Comme nous l'avons détaillé, l'objectif d'un système de reconnaissance est de classifier une requête et reconnaître la classe ou le modèle qui lui est associé. La classification est alors le processus de formation des classes cohérentes et isolées à partir d'un ensemble d'objets $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ où x décrit un vecteur de caractéristiques de la forme $\{x_1, \dots, x_m\}$, $y_k = 1$ si x_k appartient à la classe 1 et $y_k = -1$ si x_k est dans une autre classe. Un objet y est assigné à l'une de ces classes et son vecteur caractéristique est utilisé pour apprendre le classifieur. La classe d'un objet est obtenue en calculant la fonction d'apprentissage f :

$$y_i = f(x_i); \forall i \in \{1, \dots, n\} \quad (1.28)$$

Dans cette partie, nous allons passer en revue quelques classifieurs usuels utilisés dans le cadre de la reconnaissance de formes. Malgré la prolifération des techniques de classification de symboles dans la littérature, le choix d'un "bon" classifieur reste un sujet ouvert. La phase de classification d'un système de reconnaissance consiste à utiliser le vecteur des primitives résultant de la phase d'extraction des primitives afin d'assigner l'objet à une classe. Dans notre travail, nous évoquons quelques méthodes non supervisées ainsi que les méthodes supervisées vu qu'elles sont les plus communes à notre contexte de travaux.

Plusieurs méthodes de classification supervisée publiées dans la littérature s'appuient sur des techniques différentes : inférence bayésienne [Barrat and Tabbone2010], k-plus proches voisins, arbres de décision [Marée *et al.*2005], réseaux de neurones [Yang *et al.*1993] qui ont déjà été appliqués avec succès à l'apprentissage de nombreux problèmes et de reconnaissance de formes. L'exploration de données techniques a été appliquée pour sélectionner un sous-ensemble optimal de caractéristiques de formes, qui sont ensuite utilisées pour la classification. Les méthodes non

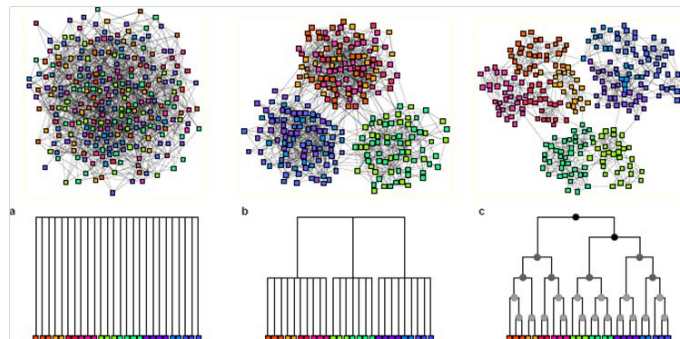


FIGURE 1.15 – Un exemple d'un dendrogramme où la position de deux objets au sein d'un groupe ou entre les groupes n'est pas liée à leur proximité dans l'espace des données. La longueur/hauteur de chaque branche est égale à la distance moyenne entre les groupes. En traçant une ligne verticale/horizontale à une hauteur donnée de l'arbre, on peut identifier k groupes (http://neamar.fr/Res/Fouille_Sociale/fouille_donnees.pdf).

supervisées se composent elles mêmes de deux catégories : les approches *hiérarchiques* et les approches de *partitionnement*. Le côté intéressant de ces approches *hiérarchiques* réside essentiellement dans la possibilité de représenter des relations entre des *sous-ensembles* souvent basés sur la structure d'arbre appelée dendrogramme. La navigation d'un document graphique est par la suite réduite à une simple navigation du dendrogramme (voir Figure.1.15). La limite de ces approches est la complexité élevée de la construction du dendrogramme ainsi que la définition des classes à reconnaître à partir de ce dernier et le seuil utilisé pour le découpage. Passons maintenant à la description des approches dites approches de *partitionnement*.

1.4.2 k -plus proche voisin

Il s'agit de l'algorithme le plus populaire en reconnaissance. Cette méthode a été très étudiée dans la littérature et de nombreuses variantes ont été proposées [Boiman *et al.*2008]. Elle utilise des mesures de similarité pour attribuer une classe à un objet. Habituellement, la distance euclidienne est utilisée comme mesure de similarité entre les objets qui sont rassemblés en nuages de points très rapprochés s'ils appartiennent à la même classe sinon ils sont très éloignés (voir Figure.1.16). La décision de ne retenir qu'une seule classe est basée sur un processus de vote majoritaire. La classe majoritaire sera attribuée à l'objet requête. Bien que cette méthode est la plus répandue dans le contexte de la classification, elle présente un biais car l'ensemble d'apprentissage doit être conservé au cours de l'étape du calcul de la distance avec tous les objets appris pour assigner une classe à l'objet requête.

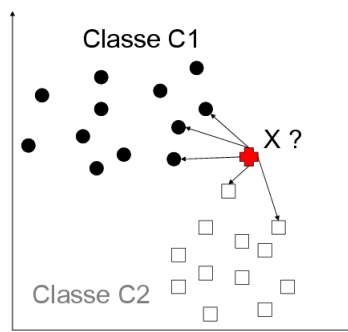


FIGURE 1.16 – Classification avec les k -plus proche voisin (<http://julien.pinquier.free.fr/These/these/node29.html>).

1.4.3 k -means

L'algorithme k -means est sans doute le plus utilisé des algorithmes de regroupement. Il a été mis au point par McQueen en 1967 [MacQueen1967]. A partir de k centroïdes (généralement initialisés aléatoirement), cet algorithme itératif affecte chaque point de l'espace à son plus proche centroïde, puis re-calcule le barycentre de chaque groupe pour déterminer à nouveau les centroïdes. L'algorithme s'arrête lorsqu'un critère de convergence est satisfait, et le partitionnement final détermine le mapping entre chaque point de l'espace et l'un des centroïdes.

Dans la pratique, la construction du vocabulaire visuel consiste à quantifier l'espace de tous les descripteurs locaux décrivant l'objet et de les regrouper en clusters (voir Figure.1.17). Le nombre de clusters représente la taille du dictionnaire visuel. Chaque centre du cluster représente un mot visuel dans le dictionnaire visuel. Un mot n'est plus un descripteur, mais représente un cluster. Commençons par un ensemble de données :

$X = \{X_1, X_2, \dots, X_n\}$, où X_i sont des vecteurs de même taille p , on obtient par k -regroupement des groupes de différentes tailles : $\{C_1, C_2, \dots, C_k\}$. Les propriétés qui doivent être remplies par les partitions sont les suivantes :

1. pas de partition vide,
2. chaque élément de la base appartient exactement à une seule partition.

L'idée principale du partitionnement est que la similarité intra-cluster soit maximisée et la similarité inter-cluster soit réduite au minimum.

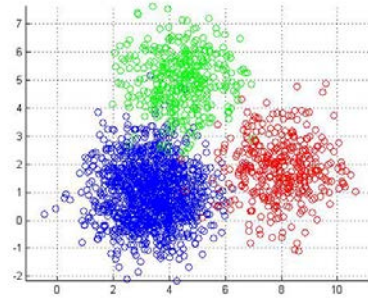


FIGURE 1.17 – Classification avec les *k-means* (<http://www.mathworks.com/matlabcentral/fileexchange/24616-kmeans-clustering>).

1.4.4 Arbre de Décision

L'arbre de décision est l'une des méthodes les plus largement utilisées et pratiques dans le domaine de la classification et surtout en classification supervisée. Il y a de nombreux algorithmes pour construire des arbres de décision. Dans notre travail, nous avons utilisé les plus populaires : à savoir C4.5 (Quinlan, 1993) et J48. Un arbre de décision peut être considéré comme un ensemble de données structurées en forme d'un arbre. La pertinence de la classification est déterminée par la réduction de l'entropie qui décrit la pureté des échantillons. Les nœuds de l'arbre sont appelés *nœuds de décision* car chaque nœud contient un test qui permet de partitionner les objets (voir Figure.1.18). Un avantage majeur de l'arbre de décision est leur simple structure, qui permet une bonne interprétation des données (la plupart des caractéristiques importantes sont à proximité du nœud racine) ainsi qu'une bonne visualisation de ces dernières. Cependant, les arbres de décision présentent quelques inconvénients : d'une part, les algorithmes de construction des arbres ne sont pas optimaux et il est difficile de retenir une heuristique relative à la construction, d'autre part, un mauvais choix de l'ensemble d'échantillonnage peut entraîner une erreur assez conséquente sur la classification.

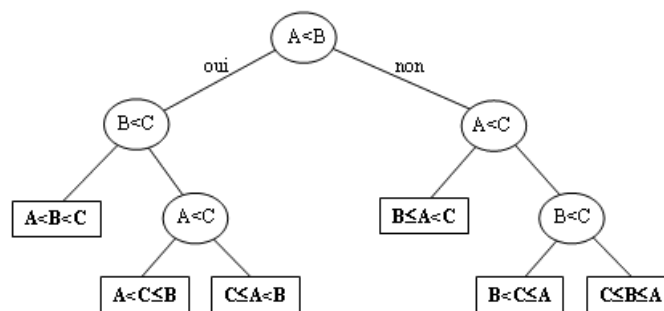


FIGURE 1.18 – Exemple d'un arbre de décision (<http://imss-www.upmf-grenoble.fr/prevert/Prog/Tris/ArbreDecision.html>).

1.4.5 Les machines à vecteur de support

Les machines à vecteur de support (SVM) ont été inventées en 1979 à partir des considérations théoriques de Vladimir Vapnik [Bi and Vapnik2003] sur le développement d'une théorie

statistique de l'apprentissage : *la Théorie de Vapnik-Chervonenkis*. Fondamentalement, les SVMs séparent deux classes différentes par le biais d'hyperplans. Si les classes sont séparables par des hyperplans, une fonction optimale peut être déterminée à partir des données empiriques. Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais Support Vector Machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression. Les SVMs sont une généralisation des classifieurs linéaires. Les SVMs ont rapidement été adoptés pour leur capacité pour traiter des données de grandes dimensions, leur faible nombre de paramètres et leurs robustesse en pratique. Cette méthode a prouvé son efficacité dans le domaine de la classification et s'est avérée très performante dans de nombreuses applications [Bosch *et al.*2007], [Grauman and Darrell2005]. Cependant, sa version non-linéaire par exemple, dépend d'une fonction noyau donc le choix reste un problème ouvert.

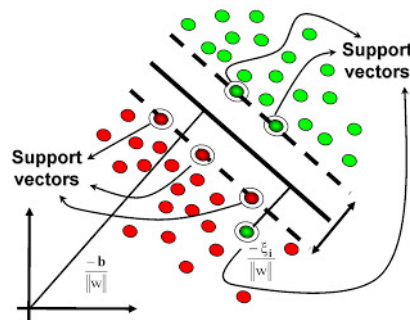


FIGURE 1.19 – Machine à vecteur de support (<http://kokzard.blogspot.fr/2011/10/jfjkdshfkjsldf.html>).

1.4.6 Classifieur de Bayes

Le classifieur bayésien naïf fonctionne comme suit :

- Soit T un ensemble d'échantillons d'apprentissage étiquetés par leur classe. Il existe k classes, C_1, C_2, \dots, C_k . Chaque échantillon est représenté par un vecteur à n dimensions, $X = x_1, x_2, \dots, x_n$, représentant n valeurs des attributs respectivement A_1, A_2, \dots, A_n .
- Étant donné un échantillon X , le classifieur prédit que X appartient à la classe ayant la plus haute probabilité a posteriori, sachant X .
 X appartient à la classe C_i si et seulement si :

$$P(C_i|X) > P(C_j|X) \text{ for } 1 \leq j \leq m, j \neq i \quad (1.29)$$

Ainsi, nous trouvons la classe qui maximise $P(C_i|X)$. La classe C_i où $P(C_i|X)$ est maximale est appelée le maximum a posteriori des hypothèses. Selon le théorème de Bayes, la probabilité $P(C_i|X)$ pourra être calculée en fonction de $P(X|C_i)$, $P(X)$ et $P(C_i)$ comme suit :

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (1.30)$$

et ces probabilités sont estimées à partir des données utilisées. Tant que $P(X)$ est la même pour toutes les classes, seulement $P(X|C_i) P(C_i)$ doit être maximisé. Si la classe des probabilités a priori $P(C_i)$ n'est pas connue, il est communément admis que les classes sont équiprobables, cela veut dire que $P(C_1) = P(C_2) = \dots = P(C_k)$, et nous devons donc

maximiser $P(X|C_i)$. Sinon, nous maximisons $P(X|C_i) P(C_i)$. Notons que la probabilité de la classe a priori peut être estimée comme suit :

$$P(C_i) = \frac{\text{freq}(C_i, T)}{|T|} \quad (1.31)$$

- Compte tenu de jeux de données avec de nombreux attributs, le calcul de $P(X|C_i)$ est coûteux. Afin de réduire la complexité du calcul de $P(X|C_i) P(C_i)$, une hypothèse d'indépendance de classe est prise en compte. Ceci suppose que les valeurs des attributs sont conditionnellement indépendantes les unes des autres, compte tenu de l'étiquette de classe de l'échantillon. Mathématiquement parlant, cela peut se traduire par :

$$P(C_i|X) \approx \prod_{k=1}^n P(x_k|C_i) \quad (1.32)$$

- Afin de déterminer la classe de X, $P(X|C_i) P(C_i)$ est évaluée pour chaque classe C_i . Le réseau Bayésien retient C_i comme la classe de X *si et seulement si* C_i maximise $P(X|C_i) P(C_i)$.

Dans cette partie, nous avons passé en revue quelques classifieurs utilisés en reconnaissance de formes et qui ont été utilisés tout le long de cette thèse à titre d'évaluation comme indiqué dans la Table.1.1. Chaque classifieur présente des avantages mais aussi des inconvénients.

1.4.7 Conclusion

Dans cette partie, nous avons évoqué quelques méthodes de classification qui, à notre avis, sont les plus corrélées à notre travail. Après cette étude, notre constat général, est qu'il est assez difficile de se prononcer pour telle ou telle approche car elles ont toutes des avantages et des inconvénients. Dans cette thèse et pour des raisons qui seront explicitées ultérieurement, nous nous sommes intéressés à un outil qui, à notre connaissance, a été soit, rarement utilisé dans le contexte de la classification d'objets graphiques [Guillas *et al.*2005], soit à notre connaissance jamais utilisé dans le cadre de la localisation de symboles dans les documents graphiques. Cet outil a émergé du domaine de l'Analyse Formelle de Concepts. Un intérêt croissant dans l'Analyse Formelle de Concepts a augmenté l'utilisation de la structure mathématique appelée treillis de Galois [Kaytoue-Uberall *et al.*2010], [Ganter and Wille1999].

Dans le prochain chapitre, nous allons exposer le formalisme du treillis de Galois afin de mettre en lumière ses avantages, ses aspects algorithmiques ainsi que sa complexité. Nous nous sommes attachés à illustrer quelques notions théoriques données par des exemples. Nous avons détaillé aussi les algorithmes de classification associés aux treillis et les plus répandus dans la littérature pour pouvoir les adapter à notre chaîne de traitement de reconnaissance de symboles graphiques. Nous présentons une analyse du formalisme du treillis de Galois (ou treillis de concepts) constituant l'outil majeur de ce travail. Nous exposons ses fondements mathématiques ainsi qu'une description approfondie des algorithmes de construction, d'apprentissage et de classification basés sur cette structure mathématique.

Classifieur	Avantages et Inconvénients
<i>k</i>-plus proche voisin	<ul style="list-style-type: none"> – Il est très simple à mettre en œuvre – Il a prouvé sa robustesse aux données bruitées – Un inconvénient réside dans le fait qu'à chaque nouvel individu à classer, il est nécessaire de parcourir tout l'ensemble d'apprentissage – Temps de calcul des distances assez lourd
k-means	<ul style="list-style-type: none"> – Le plus populaire des méthodes clustering – Un inconvénient réside dans le fait que cet algorithme tend à trouver des classes sphériques de même taille – Dépend de l'initialisation ainsi que le nombre de classes – Possibilité d'obtenir des classes vides car la définition de la classe se fait à partir de son centre qui peut ne pas être un individu de l'ensemble à classer – Complexité NP-difficile
SVM	<ul style="list-style-type: none"> – Montre de très bons résultats en général – Sert comme référence d'évaluation des autres approches – Performant au traitement des problèmes non linéairement séparables – Temps de calcul trop important et exhaustif dans certains cas – Délicatesse du choix de la fonction noyau – Complexité calculatoire dans le cas de la non décomposition du problème
Classifieur de Bayes	<ul style="list-style-type: none"> – Prouve de bons résultats dans le contexte de la classification – Taux d'erreur assez faible – Nécessite un grand nombre de données d'apprentissage
Arbre de décision	<ul style="list-style-type: none"> – Simple à manipuler – Tolérant à différentes distributions de données – Taux de classification élevé – Complexité réduite – L'ordre de traitement des attributs est figé

TABLE 1.1 – Avantages et inconvénients de quelques classifieurs usuels.

Treillis de Galois et Analyse Formelle des Concepts

"Les statisticiens sont comme les artistes, ils tombent amoureux de leurs modèles".

George Box

Sommaire

2.1	Introduction	36
2.2	Étude Théorique de la structure du Treillis de Galois	36
2.3	Fondements Mathématiques	37
2.3.1	Préliminaires et Définitions	37
2.3.2	Un exemple illustratif d'un contexte Formel	40
2.4	Algorithmes de Construction du Treillis de Galois	42
2.4.1	Phase d'apprentissage	42
2.4.2	Phase de Classement	49
2.4.3	Complexité des Algorithmes	49
2.5	Les règles d'association	49
2.6	Systèmes à Base du Treillis de Galois	51
2.6.1	GRAND	51
2.6.2	Complexité	52
2.6.3	LEGAL et LEGAL-E	52
2.6.4	Principe et Construction du pseudo-treillis	52
2.6.5	Complexité	53
2.6.6	GALOIS	53
2.6.7	RULEARNER	54
2.6.8	Apprentissage et Classement	54
2.6.9	CIBLe	54
2.6.10	Complexité	54
2.6.11	CLNB & CLNN	55
2.6.12	Complexité et Comparaison expérimentale	55
2.6.13	Comparaison des systèmes à base du Treillis de Galois	55
2.6.14	Discussion	56
2.7	Conclusion	57

2.1 Introduction

Le succès de la fouille de données et des techniques de recherche d'informations au sein des bases de données structurées et semi-structurées a récemment émergé dans le domaine de vision par ordinateur. Comme nous l'avons déjà signalé dans le chapitre précédent, les approches classiques de reconnaissance de symboles graphiques utilisent des classifieurs classiques comme l'algorithme de plus proche voisin, les arbres de décision ou les réseaux bayésiens qui ont déjà été appliqués avec succès dans le cadre des problèmes de reconnaissance de formes.

Néanmoins, l'intérêt croissant pour l'Analyse Formelle de Concepts (AFC) depuis 2000, soit dans le domaine de la fouille de données, soit dans le domaine de la représentation des connaissances a fait émerger une nouvelle structure mathématique appelée le treillis de Galois. Même si l'AFC semble abstraite et théorique, elle a prouvé son applicabilité dans plusieurs domaines comme le génie logiciel et le web. Ces applications se caractérisent souvent par des informations volumineuses où il s'agit d'extraire des connaissances exceptionnelles. Pour explorer ce volume de données, l'AFC se base sur la structure de données appelée treillis de concepts. Le treillis est un graphe dans lequel les nœuds, appelés aussi concepts établissent des liens entre $\{Objets * Propriétés\}$. Le treillis de concepts organise l'information d'un ensemble d'objets partageant des propriétés communes.

Dans la suite, nous allons présenter quelques préliminaires concernant cette structure qui fera l'objet des travaux de cette thèse. Nous accordons ensuite une large partie pour la description des algorithmes de construction, d'apprentissage et de classification basés sur le treillis de concepts.

2.2 Étude Théorique de la structure du Treillis de Galois

Le succès de la structure du treillis de Galois dans le domaine de la fouille du texte, nous emmène à mieux explorer cet outil et l'intégrer dans des chaînes de traitements dédiées à la classification d'objets graphiques par exemple. La structure du treillis a été introduite par Birkhoff [Birkhoff1967] comme une structure algébrique munie de deux opérateurs appelés *borne inférieure* et *borne supérieure*. Le treillis des concepts, introduit dans les années 1990 par Wille [Ganter and Wille1999], est quant à lui une conséquence immédiate d'un résultat fondamental de la théorie des treillis qui établit l'existence d'une bijection entre un treillis et le treillis de Galois défini à partir d'une connexion de Galois entre deux ensembles (objets et attributs).

La théorie des treillis définit ainsi un lien bijectif entre treillis, tables binaires $\{Objets * Attributs\}$ (via le treillis de concepts), et systèmes de règles implicatives (via le treillis des fermés). La classification basée sur les treillis est une façon de désigner le fait que des hypothèses résultant d'une phase d'abstraction des données sont organisées sous la forme d'un treillis [Nguifo and Njiwoua2005]. Tout système de classification utilisant cette approche, recherche dans cet espace les propriétés les plus pertinentes pour le concept à apprendre [Barbut and Monjardet1970].

La structure du treillis permet de restituer le concept décrit par les données dans toute sa complexité et sa diversité [Nguifo and Njiwoua2005].

Les treillis de concepts formels sont une structure mathématique permettant de représenter les classes non disjointes sous-jacentes à un ensemble d'objets (exemples, instances) décrits à partir d'un ensemble d'attributs (propriétés, descripteurs, items ou motifs). Une classe matérialise un concept. Ce concept peut être défini formellement par une *extension* (exemples du concept) ou par une *intension* (abstraction du concept). Le treillis de concepts est un graphe possédant la propriété du treillis qui se définit à partir de données organisées sous forme d'une table binaire $\{Objets * Attributs\}$. Les sommets du graphe sont des concepts - un concept est un regroupement

maximal d'objets possédant des attributs en commun. Le treillis ainsi composé de concepts reliés par inclusion, fournit une représentation des données très intuitive.

Nous nous intéressons dans cette partie aux méthodes basées sur les treillis de concepts. Nos travaux durant cette thèse portent sur l'utilisation des outils de l'AFC dans le cadre de la reconnaissance de formes. Ce travail aborde une méthode de classification supervisée, en utilisant la structure du treillis de Galois. Cette structure dérive du domaine de la fouille de données où la plupart des travaux de classification utilisant les treillis de Galois reposent sur des stratégies de sélection qui consiste à sélectionner ou choisir les concepts qui comportent l'information la plus pertinente à partir de l'énorme quantité de données disponibles.

Généralement, l'étape de classification est ensuite assurée par un classificateur classique, telle que le k -plus proches voisins, le réseau bayésien ou encore sur une stratégie de navigation dans le treillis similaire à la navigation dans un arbre de décision.

2.3 Fondements Mathématiques

Le principe du treillis de concepts consiste à regrouper des objets non étiquetés partageant des propriétés communes dans des classes. Une caractéristique importante de méthodes de classification conceptuelle, c'est que toute classe de sortie, en plus d'être caractérisée par sa description *extension* (l'ensemble des objets couverts par la classe), est également caractérisée par une partie *intension* qui désigne la description de ces objets. Une deuxième caractéristique, bien que n'étant pas de définition de regroupement conceptuel, n'en est pas moins souvent souhaitable, c'est que les classes de sortie sont disposées dans une hiérarchie basée sur une relation de *{généralisation/spécification}*.

Une autre caractéristique souvent souhaitable, c'est que le processus de formation des classes est progressif, cela veut dire que le traitement du $n^{i\text{ème}}$ objet ne nécessite pas de traitement de l'étendue des $n-1$ objets déjà traités. Toutes ces caractéristiques rendent la structure du treillis de concepts appropriée au processus de classification de symboles graphiques et peut être une motivation pour la communauté d'explorer cette structure et l'adapter à divers problèmes (fouille de données, classification de symboles, la reconnaissance d'images, ...).

Avant de passer à l'étude des aspects algorithmiques du treillis de concepts, il est nécessaire de présenter les fondements mathématiques du treillis de Galois. Pour l'élaboration de cette partie, nous nous sommes basés sur le livre de B. Ganter & R. Wille, *Formal Concept Analysis* [Ganter and Wille1999]. Nous nous sommes efforcés de détailler le principe du treillis avec des exemples illustratifs pour qu'un non-mathématicien puisse assimiler les notions de l'AFC avec peu de connaissances.

2.3.1 Préliminaires et Définitions

Les treillis de concepts formels (ou treillis de Galois) sont une structure mathématique permettant de représenter les classes non disjointes sous-jacentes à un ensemble d'objets (exemples, instances, tuples ou observations) décrits à partir d'un ensemble d'attributs (propriétés, descripteurs ou items). Ces classes non disjointes sont aussi appelées concepts formels, hyper-rectangles ou ensembles fermés.

Une classe matérialise un concept (à savoir une idée générale que l'on a d'un objet). Ce concept peut être défini formellement par une *extension* (exemples du concept) ou par une *intension* (abstraction du concept). Les principales définitions concernant les correspondances et treillis de Galois ont été présentées dans la littérature [Bouthinon *et al.*2006]. D'autres travaux portant sur les treillis de Galois redéfinis dans le champ de l'Analyse Formelle de Concepts, montrent la

robustesse d'une telle structure dans le domaine de la fouille du texte et l'analyse formelle des concepts. Notre objectif sera donc de faire migrer cette structure mathématique du contexte de la fouille du texte et de l'adapter au contexte de la reconnaissance de symboles. Nous exposons dans ce chapitre, les notions les plus primordiales pour le formalisme des treillis de concepts et nous détaillons nos contributions utilisant le treillis de concepts dans les chapitres suivants.

Définition 1 (Ensembles ordonnés et treillis) Un ensemble ordonné est un ensemble muni d'une relation d'ordre \leq . Un ensemble ordonné (\mathcal{M}, \leq) est un treillis *si et seulement si* tout couple d'éléments (x, y) de \mathcal{M} a un plus petit majorant (ou supremum) $(x \vee y)$, et un plus grand minorant (ou infimum) $(x \wedge y)$.

Définition 2 (Correspondance de Galois) Soient $m1 : \mathcal{P} \rightarrow \mathcal{Q}$ et $m2 : \mathcal{Q} \rightarrow \mathcal{P}$ des fonctions définies sur deux ensembles ordonnés $(\mathcal{P}, \leq_{\mathcal{P}})$ et $(\mathcal{Q}, \leq_{\mathcal{Q}})$. $(m1, m2)$ est une correspondance de Galois si $\forall p, p1, p2$ de \mathcal{P} et pour tout $q, q1, q2$ de \mathcal{Q} :

- C1 : $p1 \leq_{\mathcal{P}} p2 \Rightarrow m1(p2) \leq_{\mathcal{Q}} m1(p1)$
- C2 : $q1 \leq_{\mathcal{Q}} q2 \Rightarrow m2(q2) \leq_{\mathcal{P}} m2(q1)$
- C3 : $p \leq_{\mathcal{P}} m2(m1(p))$ et $q \leq_{\mathcal{Q}} m1(m2(q))$

Illustration 1 Soit deux ensembles ordonnés : (\mathcal{G}, \preceq) et $(\mathcal{P}(\mathcal{I}), \subseteq)$:

- \mathcal{G} est un langage dont un terme est un sous-ensemble d'un ensemble d'attribut $\mathcal{A} = \{x1, x2, x3, y3, y4, y5, y6, y7, y8\}$. Ici $c1 \preceq c2$ signifie que le terme $c1$ est moins spécifique que le terme $c2$ (par exemple, $\{a3, a4\} \preceq \{a3, a4, a6\}$),
- \mathcal{I} est un ensemble d'instances = $\{o1, o2, o3, o4, o5, o6, o7, o8\}$.

Soient *int* et *ext* deux fonctions telles que *int* : $\mathcal{P}(\mathcal{I}) \rightarrow \mathcal{G}$ et *ext* : $\mathcal{G} \rightarrow \mathcal{P}(\mathcal{I})$ avec :

- *int*($e1$) est l'ensemble des *attributs* communs à toutes les instances de $e1$.
- *int*($e1$) est appelée l'*intension* de $e1$.
- *ext*($c1$) = $\{i \in \mathcal{I} \text{ tels que } i \text{ isa } c1 \text{ où } \textit{isa} \text{ est l'appartenance usuelle d'une instance à un terme.}$
- *ext*($c1$) est donc l'ensemble des *instances* qui ont tous les attributs de $c1$.

Un exemple est décrit dans la Table.2.1 : chaque ligne i représente l'*intension* *int*(i) d'une instance de \mathcal{I} et chaque colonne j représente l'*extension* *ext*(j) d'un attribut de \mathcal{A} .

- *int* et *ext* forment une correspondance de Galois sur \mathcal{G} et $\mathcal{P}(\mathcal{I})$.

Définition 4 (Treillis de Galois) Soient $m1 : \mathcal{P} \rightarrow \mathcal{Q}$ et $m2 : \mathcal{Q} \rightarrow \mathcal{P}$ deux fonctions définies sur les treillis $(\mathcal{P}, \leq_{\mathcal{P}})$ et $(\mathcal{Q}, \leq_{\mathcal{Q}})$, telles que $(m1, m2)$ est une correspondance de Galois.

Soit $G = (p, q)$, où p est un élément de \mathcal{P} et q un élément de \mathcal{Q} , tels que $p = m2(q)$ et $q = m1(p)$. Soit \leq la relation définie par : $(p1, q1) \leq (p2, q2)$ ssi $q1 \leq_{\mathcal{Q}} q2$. (G, \leq) est un treillis appelé treillis de Galois.

Nous utiliserons si nécessaire la notation complète $G(\mathcal{P}, m1, \mathcal{Q}, m2)$. En classification supervisée, l'apprentissage se ramène dans le cas général à un problème de recherche dans l'espace des hypothèses. Le terme concept peut être employé dans les sens <unité de connaissances à apprendre> et <structure mathématique>. Dans le cas où cela peut prêter à ambiguïté, nous utiliserons concept formel pour désigner concept au sens mathématique.

Définition 5 Un treillis de concepts ou treillis de Galois est un ensemble ordonné, où deux éléments quelconques possèdent un plus petit majorant (appelé Borne Supérieure ou **BS**) et un

TABLE 2.1 – La table binarisée d’un treillis de concepts décrivant la relation : $\mathcal{R}(o_i, a_j)=1$ si le $j^{\text{ème}}$ attribut a_j caractérise le $i^{\text{ème}}$ objet o_i .

	x1	x2	x3	y3	y4	y5	y6	y7	y8
o1	✓			✓	✓	✓			
o2		✓			✓			✓	
o3		✓		✓			✓	✓	
o4			✓		✓	✓			✓
o5	✓			✓	✓	✓		✓	
o6	✓	✓		✓	✓	✓			
o7	✓			✓	✓	✓		✓	
o8	✓			✓	✓	✓			✓

plus grand minorant (appelé Borne Inférieure ou **BI**).

Définition 6 Un contexte \mathcal{K} est un triplet $(\mathcal{O}, \mathcal{A}, \mathfrak{S})$ où \mathcal{O} est un ensemble d’objets, \mathcal{A} est un ensemble d’attributs et \mathfrak{S} est une correspondance de $\{\mathcal{O} \times \mathcal{A}\}$ dans $\{0, 1\}$.

Définition 7 Soit le contexte $\mathcal{K}=(\mathcal{O}, \mathcal{A}, \mathfrak{S})$, nous définissons deux correspondances de $\mathcal{P}(\mathcal{O})$ dans $\mathcal{P}(\mathcal{A})$ et de $\mathcal{P}(\mathcal{A})$ dans $\mathcal{P}(\mathcal{O})$:

$$\begin{aligned} \forall \mathcal{A}' \subseteq \mathcal{O}, \mathcal{A}' &= \{f \in \mathcal{A} \mid \forall o \in \mathcal{A}, \mathfrak{S}(o, f)=1\} \\ \forall \mathcal{B}' \subseteq \mathcal{A}, \mathcal{B}' &= \{o \in \mathcal{O} \mid \forall f \in \mathcal{B}, \mathfrak{S}(o, f)=1\} \end{aligned}$$

\mathcal{A}' est appelé **dual** de \mathcal{A} , similairement \mathcal{B}' est appelé **dual** de \mathcal{B} .

Définition 8 Soit le contexte $\mathcal{K} = (\mathcal{O}, \mathcal{F}, \mathfrak{S})$, le couple $\mathcal{C}=(\mathcal{A}, \mathcal{B})$ est appelé un concept de \mathcal{K} si et seulement si $\mathcal{A}'=\mathcal{B}$ et $\mathcal{B}'=\mathcal{A}$.

Définition 9 \mathcal{A} est appelé l’**extension** du concept \mathcal{C} et \mathcal{B} est appelé l’**intension**. Ceci est défini par $\mathcal{A}=\mathbf{extension}(\mathcal{C})$ et $\mathcal{B}=\mathbf{intension}(\mathcal{C})$.

Considérant une relation d’ordre définie par la relation d’inclusion des intensions, on peut définir un treillis de Galois ou treillis de concepts :

Définition 10 Le treillis complet $\mathcal{G}(\mathcal{K})$ de concepts du contexte \mathcal{K} est appelé le **treillis de concepts**.

Définition 11 Pour l’ensemble des objets $\mathcal{I} \subseteq \mathcal{O}$, est associé un ensemble d’attributs $f(\mathcal{A})$ en relation \mathfrak{S} avec $\mathcal{I} : \{f(\mathcal{I})=x \in \mathcal{A} \mid \forall p \in \mathcal{I}, \mathfrak{S}(p, x)=1\}$.

Réciproquement, pour un ensemble d’attributs $\mathcal{F} \subseteq \mathcal{A}$, un ensemble d’objets est défini $g(\mathcal{F}) \subseteq \mathcal{O}$ en relation \mathfrak{S} avec $\mathcal{F} : \{g(\mathcal{F})=p \in \mathcal{O} \mid \forall x \in \mathcal{F}, \mathfrak{S}(p, x)=1\}$.

Un concept est un couple {objet-attributs} $(\mathcal{I}, \mathcal{F})$ avec $\mathcal{I} \subseteq \mathcal{O}, \mathcal{F} \subseteq \mathcal{A}, f(\mathcal{I})=\mathcal{F}$ et $g(\mathcal{F})=\mathcal{I}$. Deux concepts $(\mathcal{I}_1, \mathcal{F}_1)$ et $(\mathcal{I}_2, \mathcal{F}_2)$ sont en relation s’ils vérifient la propriété d’inclusion : $(\mathcal{I}_1, \mathcal{F}_1) \leq (\mathcal{I}_2, \mathcal{F}_2) \Leftrightarrow \mathcal{I}_1 \supseteq \mathcal{I}_2$ équivalent à $\mathcal{F}_1 \subseteq \mathcal{F}_2$.

Soit \prec la réduction transitive associée à \leq . Le concept minimal $(\mathcal{O}, f(\mathcal{O}))$ relatif à la relation \leq contient tous les objets \mathcal{O} et l’ensemble $f(\mathcal{O})$.

Notons que $f(\mathcal{O})=\emptyset$ lorsque les attributs communs à tous les objets sont supprimés. Réciproquement, le concept maximal est $(g(\mathcal{A}), \mathcal{A})$.

Il existe plusieurs types de représentation graphiques du treillis de concepts. Dans ce qui suit, nous présenterons quelques exemples des diagrammes graphiques utilisés dans la littérature ainsi que les graphiques utilisés dans nos travaux. Nous avons pu tester les deux types de représentations graphiques à savoir *le diagramme de Hasse* et *le treillis d'héritage*. Il s'agit simplement de la façon de modélisation des éléments (objets, attributs). Parfois, ce qui nous intéressait était les concepts et non pas le graphe en sa totalité. Pour ces raisons, nous avons varié la représentation utilisée en essayant d'adapter chaque structure à nos objectifs et en cherchant un bon compromis entre la précision des résultats et la complexité algorithmique du treillis. Passons maintenant à la description de ces deux méthodes.

– **Diagramme de Hasse**

Il s'agit d'un graphe orienté représentant un ensemble partiellement ordonné, dans lequel les éléments de l'ensemble sont représentés par les sommets du graphe, et il y a une arête orientée de x à y *si et seulement si* y couvre x . Les nœuds sont les concepts du treillis. Un nœud du diagramme de Hasse est un couple formé d'un ensemble d'objets (*l'extension*) (voir les lignes de la Table.2.1)) et un ensemble d'attributs (*l'intension*) (voir les colonnes de la Table.2.1)) défini comme l'ensemble de tous les attributs qui sont partagés par tous les objets présents dans *l'extension* du concept (voir Figure.2.1. (a)).

– **Treillis d'héritage**

Ce treillis ne montre que les nouveaux objets (resp. les nouveaux attributs) d'un nœud par rapport à ses descendants (resp. ascendants). Cette représentation permet d'éviter la redondance d'éléments dans l'écriture des nœuds. Tous les nœuds du treillis correspondent à des concepts. Comme remarqué précédemment, si un objet est inclus dans *l'extension* d'un concept, il est inclus dans *l'extension* de tous ses ascendants.

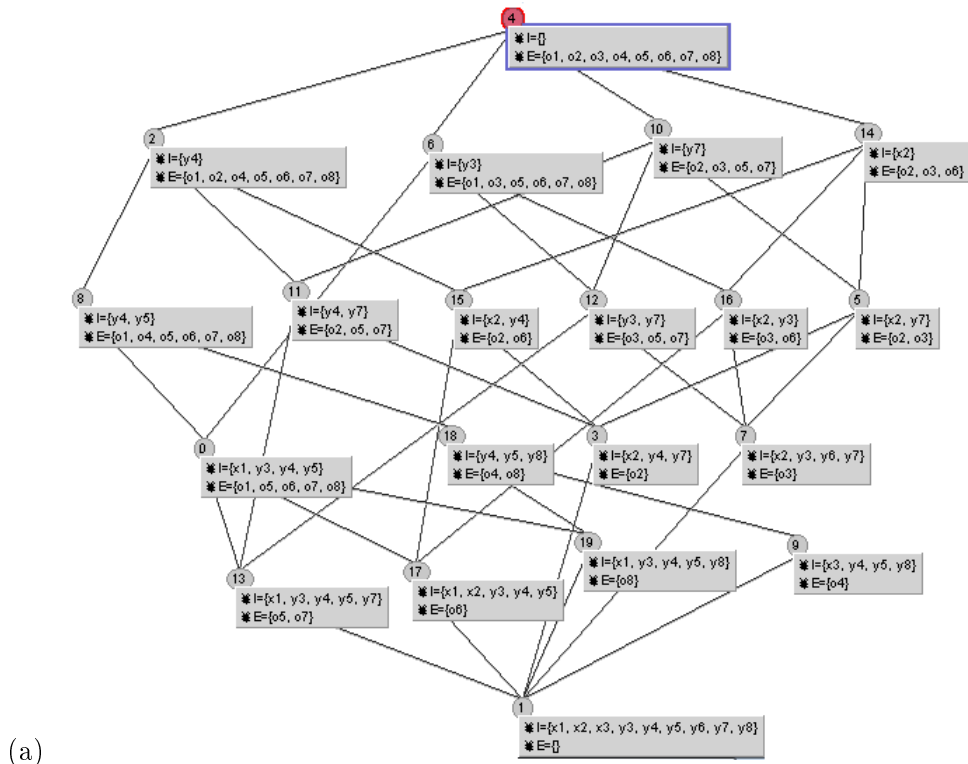
De même, si une propriété est incluse dans *l'intension* d'un concept, alors elle est incluse dans *l'intension* de tout ses descendants. C'est pourquoi dans cette représentation du treillis, les objets et les propriétés n'apparaissent qu'une seule fois [Zenou and Samuelides2003]. Nous avons présenté des exemples de cette représentation dans la Figure.2.1. (b). Nous avons utilisé ce type de présentation dans certaines expérimentations surtout dans le cas où le nombre d'objets et d'attributs est trop élevé.

2.3.2 Un exemple illustratif d'un contexte Formel

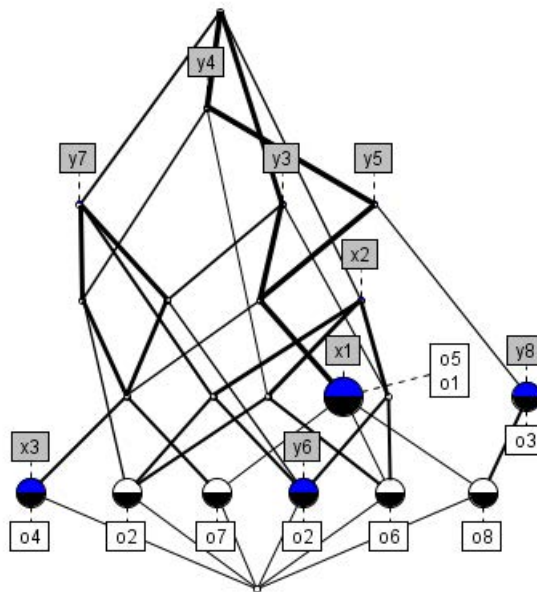
Nous présentons cet exemple, qui reste un exemple classique pour introduire la notion de l'Analyse Formelle de Concepts. Malgré sa simplicité, il est très efficace pour bien exposer le formalisme du treillis de Galois. Soit un contexte $\mathcal{K}=(\mathcal{O}, \mathcal{A}, \mathfrak{S})$ où \mathcal{O} et \mathcal{A} sont respectivement les ensembles d'objets et d'attributs et \mathfrak{S} est la relation binaire entre \mathcal{O} et \mathcal{A} : $(o, a, 1)$ signifie que l'objet o est caractérisé par l'attribut a avec $o \in \mathcal{O}$ et $a \in \mathcal{A}$. Dans l'exemple des "planètes",

- l'ensemble des objets $\mathcal{O}=\{\text{Mercure, Vénus, Terre, Mars, Jupiter, Saturne, Uranus, Neptune, Pluton}\}$,
- l'ensemble des attributs $\mathcal{A}=\{\text{Petite, Moyenne, Grande, Proche, Loin, Présence(lune), Absence(lune)}\}$,
- la relation binaire \mathcal{R} est la relation *Présence* d'un attribut : $\mathcal{R}(o, a)=1$ si $o \in \mathcal{O}$ possède l'attribut $a \in \mathcal{A}$.

Cet exemple sert d'illustration pour la comparaison entre les algorithmes de classification qui sont décrits ci-après.



(a)



(b)

FIGURE 2.1 – Représentation du (a) diagramme de Hasse construit en se basant sur l’algorithme de Godin [Godin and Missaoui1994] et du (b) treillis d’héritage de l’exemple présenté dans la Table.2.1. Le diagramme de Hasse est un graphe composé des nœuds appelés aussi concepts contenant deux parties : l’*extension* représente l’ensemble des objets partageant les mêmes caractéristiques de l’*intension*. Le premier concept se situant en haut du graphe contient : $\{\{o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8\}, \emptyset\}$. Dans le treillis d’héritage les concepts ne contiennent pas toute l’information car les parties *extension* et *intension* sont héritées d’un niveau à un autre. 41

TABLE 2.2 – Exemple des "planètes" dans le contexte de l'Analyse Formelle de Concepts.

	Taille	Dist soleil	Présence d'un satellite
	km	10 ⁶ km	Nombre
Mercure	4 878	58	0
Vénus	12 400	108	0
Terre	12 756	150	1
Mars	6 800	228	2
Jupiter	142 800	778	16
Saturne	120 800	1 427	19
Uranus	47 600	2 870	5
Neptune	44 600	4 500	8
Pluton	2 320	9 950	1

TABLE 2.3 – Table binarisée de l'exemple des "planètes" de la table.2.2.

	Taille			Dist soleil		Présence d'un satellite	
	Petite	Moyenne	Grande	Proche	Loin	Présence	Absence
Mercure	✓			✓			✓
Vénus	✓			✓			✓
Terre	✓			✓		✓	
Mars	✓			✓		✓	
Jupiter			✓		✓	✓	
Saturne			✓		✓	✓	
Uranus		✓			✓	✓	
Neptune		✓			✓	✓	
Pluton	✓				✓	✓	

2.4 Algorithmes de Construction du Treillis de Galois

Nous abordons les aspects algorithmiques du formalisme du treillis de Galois. Nous exposons quelques algorithmes de construction du treillis qui nous ont parus les plus appropriés à notre approche, suivie d'une description détaillée des systèmes de la littérature qui sont basés sur le treillis de concepts. Comme nous nous intéressons au problème de classification, nous décrivons les stratégies adoptées pour apprendre le classifieur pour qu'il puisse prédire par la suite et au cours de la phase de classification la classe de nouveaux exemples.

Elle apporte une connaissance suffisante sur les bases de la structure du treillis de Galois utilisé dans notre travail. La taille du treillis de concepts rend cette structure complexe et difficile à utiliser, pour cela la performance de cet outil dépend généralement des algorithmes utilisés soit pour la construction, l'apprentissage ou encore pour la classification.

2.4.1 Phase d'apprentissage

Dans le cas général, la phase d'apprentissage consiste à organiser un ensemble de données dans la structure d'un treillis de concepts à partir d'un ensemble d'objets. Le problème d'apprentissage supervisé est exprimé par un ensemble de données d'apprentissage des exemples oi (aussi appelés tuples, objets, instances ou observations) décrits par un ensemble fini de propriétés aj

(aussi appelées attributs ou descripteurs), et appartenant à une ou plusieurs classes y_i .

L'ensemble d'apprentissage ayant une taille $n \in \mathcal{N}$ est un ensemble de couples $\{(o_{1,Y_1}), \dots, (o_{n,Y_n})\}$, où $y_i = f(o_i)$ pour $1 \leq i \leq n$. Chaque exemple o_i est représenté par un vecteur $(o_{i_1}, \dots, o_{i_m})$, où o_{i_j} est la valeur de la propriété j appartenant à l'ensemble des attributs $a_i \in \mathcal{A}$. Cette valeur peut être numérique (valeur discrétisée par exemple) ou symbolique (booléen, ou nominale). y_i est la classe de l'ensemble des exemples o_i .

Nous allons passer en revue quelques algorithmes de construction que nous avons utilisés durant les travaux de cette thèse. Dans la littérature, un nombre important de méthodes diverses ont été proposées. La communauté qui s'intéresse à la fouille de grandes bases de texte prouve que le treillis de concepts est un outil assez efficace et répond à leurs besoins quand il est associé à des traitements d'optimisation et des heuristiques de réduction de taille. Pour faire face à ce problème, nous allons explorer les algorithmes les plus efficaces offrant un bon compromis entre la performance et la complexité de la structure du treillis de concepts. Plusieurs algorithmes de construction du treillis de concepts ont été proposés et ont fait l'objet d'études comparatives détaillées [Fu *et al.*2004b], [Kourie and Oosthuizen1998] et [Kuznetsov and Obiedkov2002] ou synthétiques [Guénoche1990]. Une extension de l'étude de Guénoche [Guénoche1990], Njiwoua [Fu *et al.*2004b] présentent six algorithmes de construction de treillis. Ces algorithmes diffèrent selon la méthode de construction retenue [Fu *et al.*2004b] (méthode ascendante ou descendante) et peuvent être divisés en deux catégories : les algorithmes incrémentaux et non-incrémentaux. La différence entre ces deux approches réside dans la façon dont l'algorithme traite de nouveaux exemples qui s'ajoutent au treillis.

Les algorithmes incrémentaux (Norris [Norris1978], Godin [Godin and Missaoui1994], Carpineto and Romano [Carpineto and Romano1996]) construisent le treillis complet et à chaque nouvel ajout, un parcours de la totalité du graphe est alors nécessaire (Chein [Chein1969], Ganter [Ganter2010], Bordat [Bordat1986]).

Le parcours du treillis dans ce cas se fait en général de deux manières différentes :

- Le parcours *Top-Down* : permet de parcourir partiellement le graphe de la racine vers les feuilles.
- Le parcours *Bottom-Up* : permet de construire le treillis en partant des feuilles vers la racine.

Tous ces algorithmes permettent de construire soit un treillis *complet* soit un *sup-demi treillis*¹. Pour certains algorithmes la taille du treillis est linéaire en fonction du nombre de concepts [Norris1978]. De nombreuses techniques ont été développées dans le but de réduire la complexité de cette structure dans le pire des cas. Le lecteur peut se référer à une évaluation assez complète proposée par [Kuznetsov and Obiedkov2001] où les auteurs exposent les avantages et les inconvénients de chacun des algorithmes énumérés ci-dessus. Nous présentons dans ce qui suit, quelques algorithmes dont nous avons eu recours tout au long de notre travail.

Algorithmes Incrémentaux

Dans ce qui suit, nous allons décrire quelques algorithmes incrémentaux que nous avons testés durant nos travaux de thèse :

Algorithme de Godin [Godin and Missaoui1994] Il s'agit d'un algorithme incrémental qui se base sur le principe du concept générateur. Un nœud (X_i, Y_i) est dit générateur d'un nœud (X_j, Y_j) si et seulement si (X_i, Y_i) est une borne supérieure de $\{(X_k, Y_k) \in \mathcal{G} \text{ tel que } Y_j = \{Y_k$

1. Un *sup* (respectivement *inf*) *demi-treillis* est un ensemble ordonné dans lequel deux éléments quelconques x et y admettent toujours une borne supérieure (respectivement inférieure)

$\cup \text{intension}(o)\}$ avec o est le nouvel exemple inséré et $\text{intension}(o)$ est la liste de ses attributs. En vue de collecter quelques statistiques sur cet algorithme, nous l'avons appliqué sur l'exemple des "planètes". Le treillis de Galois résultant est présenté dans la Figure.2.2. Les concepts qui sont générés sont présentés dans la Table.2.4.

Algorithm 1 Algorithme de Godin

```

 $\mathcal{G} := \emptyset$ 
Pour tout objet  $o \in \mathcal{O}$  Faire
  Insérer( $o, \mathcal{G}$ )

Insérer( $o, \mathcal{G}$ )
 $\mathcal{G}' = \emptyset$ 
Pour tout  $(\mathcal{X}_i, \mathcal{Y}_i) \in \mathcal{G}$ 
  si  $(\mathcal{X}_i, \mathcal{Y}_i)$  est un nœud générateur alors
     $\mathcal{G}' := \mathcal{G}' \cup (\mathcal{X}_i, \mathcal{Y}_i) \cup (\mathcal{X}_i \cup o, \mathcal{Y}_i \cap o')$ 
  sinon si  $\mathcal{Y}_i \cap o' = \emptyset$  alors
     $\mathcal{G}' := \mathcal{G}' \cup (\mathcal{X}_i, \mathcal{Y}_i)$ 
  sinon  $\mathcal{G}' := \mathcal{G}' \cup (\mathcal{X}_i \cup o, \mathcal{Y}_i)$ 
 $\mathcal{G} := \mathcal{G}'$ 
    
```

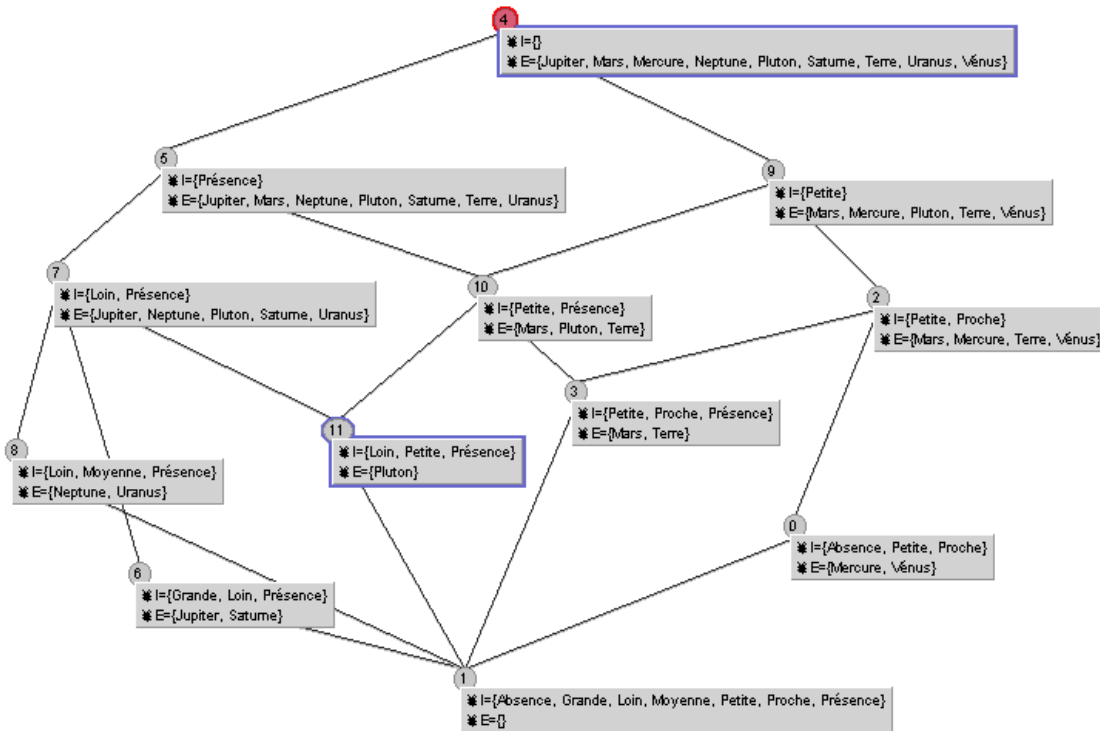


FIGURE 2.2 – Le Treillis de concepts du contexte de l'exemple des "planètes" généré en utilisant l'algorithme de Godin.

Niveau	Intension	Extension
1	\emptyset	{Jupiter, Mars, Mercure, Neptune, Pluton, Saturne, Terre, Uranus, Vénus}
2	{Présence}	{Jupiter, Mars, Neptune, Pluton, Saturne, Terre, Uranus}
3	{Petite}	{Mars, Pluton, Mercure, Terre, Vénus}
4	{Loin, Présence}	{Jupiter, Neptune, Pluton, Saturne, Uranus}
5	{Petite, Présence}	{Mars, Pluton, Terre}
6	{Loin, Moyenne, Présence}	{Neptune, Uranus}
7	{Loin, Petite, Présence}	{Pluton}
8	{Petite, Proche, Présence}	{Mars, Terre}
9	{Grande, Loin, Présence}	{Jupiter, Saturne}
10	{Petite, Proche}	{Mars, Mercure, Terre, Vénus}
11	{Absence, Petite, Proche}	{Mercure, Vénus}
12	{Petite, Proche, Absence, Grande, Loin, Moyenne, Présence}	\emptyset

TABLE 2.4 – Les concepts du treillis de Galois construit avec l’algorithme de Godin.

Algorithme de Nourine et Raynaud L’algorithme [Nourine and Raynaud2002] a montré de résultats satisfaisants en comparaison aux autres algorithmes les plus récents. Son principe est basé sur deux parties :

- La phase de génération des concepts organisés dans un arbre lexicographique.
- La phase de détermination de la relation d’ordre entre les concepts appelée aussi couverture dans le but de construire le treillis de Galois.

Algorithm 2 Algorithme de [Nourine and Raynaud2002]

Entrées : L’ensemble des concepts \mathcal{C}
Sorties : Treillis de Concepts \mathcal{G}
 $\mathcal{C}_a[1..m] = \text{extensionConceptsAttributs}(\mathcal{C}_a, \mathcal{C})$;
 Pour chaque concept $c_i \in \mathcal{C}$ faire
 compteur $_{c_i} \leftarrow 0$
 Pour chaque concept $c_i \in \mathcal{C}$ faire
 candidats $\leftarrow \emptyset$;
 Pour chaque attribut $a \in \text{Intension}(c_i)$ faire
 $\mathcal{E} \leftarrow \mathcal{C}_a[a] \cap \text{Extension}(c_i)$;
 candidat $\leftarrow \text{trouverConcept}(\mathcal{C}, \mathcal{E})$;
 compteur $_{\text{candidat}} = \text{compteur}_{\text{candidat}} + 1$;
 si compteur $_{\text{candidat}} + |\text{Intension}(c_i)| = |\text{Intension}(\text{candidat})|$ alors
 RajouterLien(c_i , candidat)
 Initialiser les compteurs;

Algorithme de Norris L’algorithme de Norris modifié que nous avons adopté dans notre travail sera exposé dans le prochain chapitre. L’algorithme original [Norris1978] est présenté dans l’algorithme.3 et nous l’avons appliqué aussi sur l’exemple des planètes comme c’est montré dans la Figure.2.4. Dans la représentation graphique que nous avons choisi (voir Figure.2.4), seuls les nouveaux objets et attributs apparaissent d’un niveau à un autre comme nous l’avons déjà expliqué. Dans nos travaux, nous avons eu recours à cet algorithme car il s’est avéré le plus adapté à notre contexte du point de vue de sa complexité. Il est le plus performant en comparaison aux

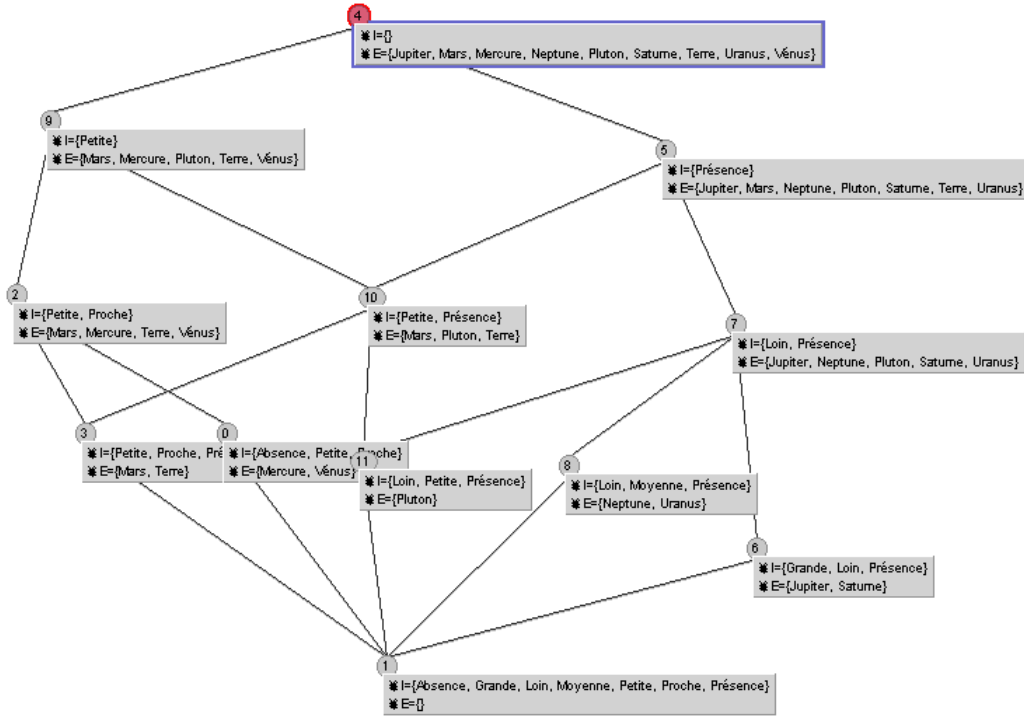


FIGURE 2.3 – Le Treillis de concepts du contexte de l'exemple des "planètes" généré en utilisant l'algorithme de Nourine et Raynaud.

autres algorithmes de la littérature car il se base sur la méthode incrémentale qui permet de ne pas générer la totalité du treillis ce qui permet de réduire la complexité et la taille de l'espace de recherche et offre ainsi un outil de classification concis et précis.

Algorithm 3 Construction du treillis de Galois [Norris1978]

```

 $\mathcal{G} := (x_1, f(x_1))$ 
Pour k := 2 à n Faire
  Pour tout  $(X_i, Y_i) \in \mathcal{G}$  Faire
    si  $Y_i \subset f(x_k)$  Faire
       $(X_i, Y_i) \leftarrow (X_i \cup \{x_k\}) \times Y_i$ 
      marquer  $x_k$ 
    sinon Faire
      si  $(X_i \cup \{x_k\}) \times (Y_i \cap f(x_k))$  est maximum alors  $\mathcal{G} := \mathcal{G} + (X_i \cup \{x_k\}) \times (Y_i \cap f(x_k))$ 
    FinPour
  si  $x_k$  n'est pas marqué Alors  $\mathcal{G} := \mathcal{G} + (x_k, f(x_k))$ 
  FinPour

```

Le treillis de Galois illustré dans la Figure.2.4 est un treillis d'héritage dont chaque concept représente une classe. Les classes sont hiérarchisées dans le treillis. Par exemple, le concept $(\{\text{Pluton, Mars, Terre}\}, \{\text{Petite}\})$ est moins particulier que le concept $(\{\text{Neptune, Uranus}\}, \{\text{Moyenne, Loin, Présence}\})$ car il possède moins de propriétés communes que ce dernier. Le parcours de l'arborescence en partant vers le haut (*top*) correspond à une généralisation et le parcours vers le bas (*bottom*) correspond à une spécialisation. Nous avons opté pour ce type de treillis vu le

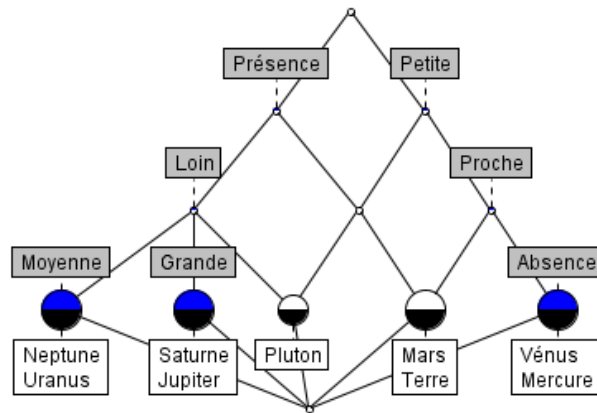


FIGURE 2.4 – Le treillis de concepts du contexte de l'exemple des "planètes" généré en utilisant l'algorithme de Norris.

Niveau	<i>Intension</i>	<i>Extension</i>
1	\emptyset	\emptyset
2	{Présence}	\emptyset
3	{Petite}	\emptyset
4	{Loin}	\emptyset
5	{Proche}	\emptyset
6	{Moyenne}	{Neptune, Uranus}
7	{Grande}	{Saturne, Jupiter}
8	\emptyset	{Pluton}
9	Absence	{Vénus, Mercure}

TABLE 2.5 – Les concepts du treillis de Galois construit avec l'algorithme de Norris.

nombre réduit de concepts ainsi que l'absence de redondance d'information dans le treillis.

Algorithmes Non Incrémentaux

De nombreux algorithmes non incrémentaux ont été proposés dans la littérature pour générer le treillis de Galois d'une relation binaire. Des algorithmes connus actuellement comme par exemple l'algorithme de Bordat [Bordat1986] et [Guénoche1990] sont utilisés.

L'idée de base de la méthode de Bordat est donnée dans l'algorithme 4. Le treillis \mathcal{G} est construit en commençant par la borne inférieure du treillis ($E \times \emptyset$) et en générant tous les prédécesseurs qui sont ensuite ajoutés à \mathcal{G} et liés à leurs parents. Ce processus de génération des nœuds fils est itératif et répété pour chaque nouvel objet ($X_i \times Y_i$).

Algorithm 4 Algorithme de Bordat

```

 $\mathcal{G} := (E \times \emptyset)$ 
Pour tout  $X \times Y \in \mathcal{G}$  Faire
  Soit  $y_1, \dots, y_p$  les  $(X_i \times Y_i) \in \mathcal{G}$  Faire
    si  $Y_i \subset f(x_k)$  Faire
       $X_i \times Y_i \leftarrow (X_i \cup \{x_k\}) \times Y_i$ 
      marquer  $x_k$ 
    sinon Faire
      si  $(X_i \cup \{x_k\}) \times (Y_i \cap f(x_k))$  est maximum alors  $\mathcal{G} := \mathcal{G} + (X_i \cup \{x_k\}) \times (Y_i \cap f(x_k))$ 
  FinPour
  si  $x_k$  n'est pas marqué Alors  $\mathcal{G} := \mathcal{G} + x_k \times f(x_k)$ 
  FinPour
  
```

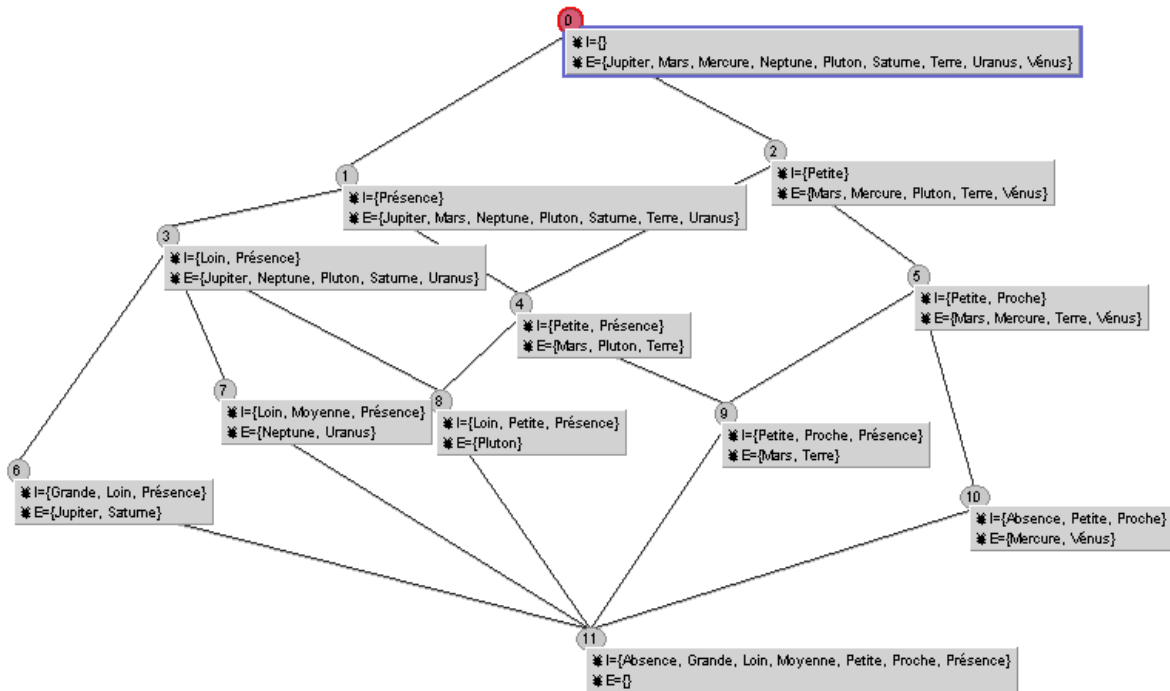


FIGURE 2.5 – Le diagramme de Hasse du contexte de l'exemple des "planètes" généré en utilisant l'algorithme de Bordat.

2.4.2 Phase de Classement

Après avoir détaillé la phase d'apprentissage du treillis de concepts consistant principalement à l'étape de sa construction, passons maintenant à la phase de classement. Durant cette étape, chaque système de classification sélectionne des stratégies appropriées pour prédire une classe pour un nouvel exemple. Ces stratégies comprennent celles se basant sur le treillis en lui même et en utilisant par exemple le vote majoritaire, le calcul de la similarité, la déduction ou celles ayant recours à des classifieurs usuels comme par exemple le k -ppv ou le classifieur de Bayes.

2.4.3 Complexité des Algorithmes

Un constat général que nous pouvons retirer après cette analyse de la littérature est qu'il existe de nombreux travaux qui se sont intéressés à comparer les algorithmes de construction du treillis de Galois en terme de complexité algorithmique ainsi qu'en terme du temps de calcul. Un travail intéressant dans ce contexte est le papier de Kuznetsov et Obiedkov [Kuznetsov and Obiedkov2002]. Dans le tableau.2.6, nous présentons une comparaison des algorithmes de construction de treillis de Galois selon l'étude présentée par [Kuznetsov and Obiedkov2002] et [Njiwoua2000].

La complexité du treillis de Galois dépend essentiellement du nombre d'objets $m=|\mathcal{O}|$, du nombre d'attributs $n=|\mathcal{A}|$, du nombre de concepts dans le treillis $|\mathcal{G}|$, et de la densité du contexte (nombre de connexions de Galois) :

$$O(\mathcal{G}) = \frac{\sum_{o,f}(\zeta(o,f))}{|\mathcal{O}|x|\mathcal{A}|} \quad (2.1)$$

Dans nos travaux de thèse, nous nous sommes basés sur l'algorithme de Norris qui s'est avéré le plus performant et le plus rapide par rapport aux autres algorithmes incrémentaux ce qui est justifié dans la table.2.6. Il fournit le plus petit nombre de concepts en comparaison aux autres méthodes. Ainsi le recours aux treillis d'héritage comme représentation graphique assure une réduction assez significative de l'information contenue dans les concepts en comparaison aux nœuds générés avec les autres méthodes. Ce qui permettra d'obtenir de bonnes performances de reconnaissance qui seront détaillées dans le chapitre suivant.

Algorithmes	Incrémental	Complexité [Kuznetsov and Obiedkov2002]	Complexité [Njiwoua2000]
Godin	✓	$O(l^4)$	$O(2^{2m} \cdot n)$
Carpineto et Romano	✓		$O(3^m \cdot 2^n \cdot n)$
Oosthuizen	✓		$O(2^n \cdot n^4)$
Norris	✓	$O(m^2 \cdot n \cdot l)$	-
Chein		$O(m^2 \cdot n \cdot l)$	$O(2^m \cdot \min(m, n)^3 \cdot (m+n))$
Ganter		$O(m^2 \cdot n \cdot l)$	$O(2^m \cdot n \cdot m^2)$
Bordat		$O(m \cdot n^2 \cdot l)$	$O(2^{\min(m,n)} \cdot n^3 \cdot m^2 \cdot \min(m, n))$

TABLE 2.6 – Évaluation de la complexité Algorithmique des différents algorithmes de construction du treillis de concepts. (-) complexité non précisée.

2.5 Les règles d'association

Le volume et la nature de données excédant les capacités humaines en terme d'analyse a fait émerger le besoin de l'utilisation des méthodes simples et faciles à interpréter comme les

règles d'association. L'objectif de l'utilisation des règles d'association est la recherche de relations d'association ou de corrélation intéressantes parmi un grand ensemble de données.

D'une manière générale, une règle d'association est une implication de la forme $A \Rightarrow B$ avec $(A \cap B) = \emptyset$.

Les règles d'association, dans le domaine de la fouille de données, sont des implications dont la valeur de vérité est observée sur un ensemble d'instances I . Chaque règle est associée à une valeur de support (la fréquence de sa partie prémisse dans I), et à une valeur de confiance. Lorsque la confiance est fixée à 1, les règles sont dites règles d'implication. Lorsque l'on considère les treillis de concepts, l'ordre partiel induit sur les termes du langage par la correspondance de Galois, correspond à un ensemble de règles d'implication. Plus précisément, $T2 \preceq_I T1$ signifie que $\text{extI}(T1) \subseteq \text{extI}(T2)$. Deux mesures sont utilisées pour les règles d'association :

- *Support* : c'est le pourcentage de transactions qui contiennent $A \cup B$ (à la fois A et B), en d'autres termes $\text{Support}(A \Rightarrow B) = P(A \cup B)$.
- *Confiance* : c'est le pourcentage de transactions contenant A qui contiennent aussi B en d'autres terme $\text{Confiance}(A \Rightarrow B) = P(B/A)$.

Dans la littérature, plusieurs algorithmes d'extraction des règles d'association ont été proposés [Latiri and Yahia2001], [Agrawal and Srikant1994a] et [Pasquier *et al.*1999]. On peut citer l'algorithme Apriori [Agrawal and Srikant1994a] dans lequel, deux paramètres, support minimum (*minsup*) et confiance minimale (*minconf*) sont introduits et deux phases principales résument l'algorithme :

- La première phase consiste à rechercher tous les sous ensembles de motifs fréquents (frequent Itemset) tel que $\text{support}(X) > \text{minsup}$.
- La deuxième phase consiste à rechercher toutes les règles d'association intéressantes et pertinentes à partir des sous ensembles fréquents X déterminés lors de la première phase tel que la $\text{confiance}(X) > \text{minconf}$.

Bien que ces règles présentent des sources de données assez riches, elles présentent deux problèmes majeurs liés à l'utilisation des règles d'association spatiales à savoir, le problème des temps d'extraction des règles d'association spatiales à partir du jeu de données et le problème de la pertinence et de l'utilité des règles d'association spatiales extraites.

Dans l'exemple ci-dessous, nous avons choisi de présenter les implications de Duquenne-Guigues [Guigues and Duquenne1986] de l'exemple des planètes, pour mettre l'accent sur l'utilité du treillis de Galois et sa capacité à inférer de grands volumes d'informations et de connaissances. Ces implications montrent les relations entre les propriétés et peut représenter une source d'information très puissante dans le contexte de grands volumes de données. Les implications relatives au contexte de l'exemple des "planètes" sont les suivantes :

1. $\langle 0 \rangle$ Petite Moyenne Loin Présence $= [100\%] \Rightarrow \langle 0 \rangle$ Grande Proche Absence ;
2. $\langle 0 \rangle$ Petite Grande Loin Présence $= [100\%] \Rightarrow \langle 0 \rangle$ Moyenne Proche Absence ;
3. $\langle 0 \rangle$ Petite Proche Loin Présence $= [100\%] \Rightarrow \langle 0 \rangle$ Moyenne Grande Absence ;
4. $\langle 0 \rangle$ Petite Proche Présence Absence $= [100\%] \Rightarrow \langle 0 \rangle$ Moyenne Grande Loin ;
5. $\langle 0 \rangle$ Moyenne Grande Loin Présence $= [100\%] \Rightarrow \langle 0 \rangle$ Petite Proche Absence ;
6. $\langle 2 \rangle$ Moyenne $= [100\%] \Rightarrow \langle 2 \rangle$ Loin Présence ;
7. $\langle 2 \rangle$ Grande $= [100\%] \Rightarrow \langle 2 \rangle$ Loin Présence ;
8. $\langle 4 \rangle$ Proche $= [100\%] \Rightarrow \langle 4 \rangle$ Petite ;
9. $\langle 5 \rangle$ Loin $= [100\%] \Rightarrow \langle 5 \rangle$ Présence ;
10. $\langle 2 \rangle$ Absence $= [100\%] \Rightarrow \langle 2 \rangle$ Petite Proche ;

11. $\langle 5 \rangle$ Petite= $[80\%]$ = $\langle 4 \rangle$ Proche ;

Ces règles représentent une source d'information très robuste qui pourra être utilisée pour extraire de la connaissance à partir des bases d'images et permet d'ouvrir un axe de recherche très prometteur.

2.6 Systèmes à Base du Treillis de Galois

Dans la section qui suit, nous allons passer en revue quelques systèmes de classification supervisée basés sur la structure du Treillis de Galois. Nous allons présenter les algorithmes d'apprentissage et de classification utilisés pour chaque système ainsi que la complexité algorithmique de chaque méthode. Nous allons évoquer aussi les avantages et les inconvénients de chaque méthode [Nguifo and Njiwoua2005]. Ces systèmes utilisent le treillis de Galois pour la phase d'apprentissage afin de choisir les éléments (objets, attributs) les plus pertinents parmi l'ensemble des données d'apprentissage et en ce qui concerne la phase de classification, ils se basent sur des classifieurs usuels comme l'algorithme de k -plus proche voisin, le réseau bayésien ou encore le vote majoritaire. Cette possibilité de combiner le treillis avec d'autres classifieurs usuels pour répondre aux besoins d'un même système justifie notre intérêt à l'étude d'une telle structure et son application dans le cadre des systèmes de reconnaissance ou encore les systèmes de localisation de symboles dans les documents. Pour des raisons de clarté, dans la suite nous allons utiliser des paramètres lors de l'évaluation de la complexité des systèmes présentés. Ces paramètres désignent les notions suivantes :

- v : le nombre maximal de valeurs qu'un attribut peut avoir.
- $|L|$: le nombre de classes dans le treillis à mettre à jour
- m : le nombre d'attributs dans chaque concept (*l'intension*)
- h : la hauteur maximale du *sup-demi treillis*
- n : le nombre de concepts d'un certain seuil α
- $k = \min(n, m)$

2.6.1 GRAND

Principe et Construction du pseudo-treillis

GRAND (GRAPh iNDuction) est le système développé par Oosthuizen [Venter *et al.*1997], qui est adapté à la classification et à l'apprentissage non supervisé. Il construit le treillis de concepts progressivement. Quand un nouvel objet est inséré, ses attributs sont comparés avec les *intensions* des nœuds déjà existants. Un nouveau nœud est alors créé dont la partie *intension* contenant les attributs communs entre l'*intension* du nouvel objet et les *intensions* des autres concepts. La particularité de la méthode GRAND est qu'elle permet de construire un pseudo-treillis avant la mise en œuvre de la phase de classification. Dans le treillis généré, seules les *intensions* sont présentées. Trois types de nœuds existent :

- les nœuds-attributs
- les nœuds-objets
- les nœuds intermédiaires

Apprentissage et Classement

GRAND applique les règles les plus spécifiques pour classer de nouveaux exemples. Un vote majoritaire est adopté. Les auteurs suggèrent de choisir les trois meilleures règles. Les règles

les plus spécifiques sont générées à partir des concepts qui contiennent le plus grand nombre d'exemples et le plus petit nombre d'attributs [Fu *et al.*2004b]. Cet algorithme offre des résultats intéressants même si d'autres algorithmes que nous allons décrire dans la suite tendent à être plus performants.

2.6.2 Complexité

L'ordre de complexité en temps de ce système est de l'ordre de $O(2^k.k^4)$, où $k=\min(n, m)$ [Oosthuizen1988]. Le temps nécessaire pour la mise à jour du treillis de Galois dépend du nombre de concepts déjà générés et de l'objet de la borne supérieure. L'ordre de complexité dans le meilleur des cas est $O(m^3)$ [Oosthuizen1988]. Dans les cas pratiques étudiés, la complexité en mémoire du treillis est de l'ordre de $O(nm^3)$.

2.6.3 LEGAL et LEGAL-E

2.6.4 Principe et Construction du pseudo-treillis

LEGAL a été développé par [Mephu Nguifo *et al.*1998]. Il est basé sur ses paramètres d'apprentissage pour construire un *sup-demi-treillis*, ce qui réduit les hypothèses dans l'espace de recherche. Cette restriction est justifiée par le fait que les nœuds se situant en bas du treillis de Galois sont trop spécifiques. Ils sont donc moins adaptés à la généralisation et le système peut conduire au phénomène de sur-apprentissage. Les objets sont séparés en deux ensembles : les exemples et les contre-exemples. L'algorithme commence à partir du nœud le plus général (\mathcal{O} , \emptyset). Pour chaque nouveau nœud, l'algorithme construit tous ses descendants, puis récupère les nœuds valides (un nœud est dit *valide* s'il contient au moins α exemples en d'autres termes s'il est vérifié par un grand nombre d'exemples positifs), ensuite, les nœuds valides sont insérés dans le treillis. Si aucun nœud valide n'existe, l'algorithme s'arrête.

Apprentissage et Classement

LEGAL est limité à une étape d'apprentissage d'une classe qui contient des instances positives et négatives. Les connaissances acquises correspondent à toutes les hypothèses pertinentes. Ce système utilise le vote majoritaire pour classer de nouveaux exemples. Il fournit deux paramètres qui sont respectivement la *justification* et la *réfutation*. Ces critères sont utilisés pour l'assignement d'une classe au nouvel exemple : si l'objet à reconnaître vérifie un nombre suffisant de concepts pertinents (seuil de justification est élevé), alors il aura la classe majoritaire parmi ces concepts. À noter que :

- Un concept est dit *pertinent* s'il est *valide*, *quasi-cohérent*. Soit \mathcal{V} l'ensemble des hypothèses valides : $\mathcal{V} = Y \subseteq \mathcal{A}$ tq $\exists \{\mathcal{A}1, \mathcal{O}1\} \in \mathcal{L}, Y = \mathcal{A}1$ et

$$\frac{\mathcal{O} \cap \mathcal{O}^+}{\mathcal{O}^+} \geq \alpha \tag{2.2}$$

Ce critère correspond à la notion de support utilisé pour générer les motifs fréquents pour les règles d'association [Pasquier *et al.*1999].

- Un concept est dit quasi-cohérent s'il est valide et contient β contre-exemples et maximal.
- Un concept *valide* et *quasi-cohérent* est *maximal* s'il n'est couvert par aucun autre concept *valide* et *quasi-cohérent* [Guillas *et al.*2005].

2.6.5 Complexité

Une analyse de la complexité du système LEGAL a été proposée par [Njiwoua and Mephu Nguifo1997].

1. Taille du *sup-demi treillis* est de l'ordre de \mathcal{G} :

$$|\mathcal{G}| \leq 2^k - 2^{\inf(k, \alpha \cdot n) - 1}, k = \min(n, m) \quad (2.3)$$

2. Complexité en Temps : l'étape de construction du treillis de Galois consiste à générer tous les sous-nœuds d'un certain nœud et insérer les nœuds *valides* dans \mathcal{G} , ce qui donne :

$$O(|\mathcal{G}| \cdot n \cdot (1 - \alpha)) \cdot (|I| + \min(n \cdot (1 - \alpha), m)) + n \cdot m \quad (2.4)$$

2.6.6 GALOIS

Principe et Construction du pseudo-treillis

Le système GALOIS utilise un algorithme incrémental pour générer un ensemble de concepts au cours de la phase d'apprentissage. Il a été développé par [Carpineto and Romano1996]. Il est plus adapté à la classification non supervisée (par exemple il est adapté dans le cadre de l'interrogation des bases de données de documents [Carpineto and Romano1996]). Dans ce cas, l'étape d'apprentissage est non supervisée et le treillis de Galois est ensuite utilisé pour classer les nouveaux exemples.

Carpineto et Romano [Carpineto and Romano1993] ne caractérisent les nœuds du treillis que par leurs intensions. Pour l'étape de mise à jour, la même technique proposée par [Godin and Missaoui1994] est appliquée : les concepts déjà présents dans le treillis peuvent être modifiés, mais ne sont jamais supprimés. Carpineto et Romano [Carpineto and Romano1996] proposent ensuite, en plus de maintenir le diagramme de Hasse du treillis, un principe très simple à suivre à chaque étape afin d'assurer le résultat souhaité. Ce principe est une conséquence du théorème fondamental de l'Analyse Formelle de Concepts [Wille2009] :

si $(\mathcal{A}_i, \mathcal{O}_i)$ est la borne supérieure de deux concepts $(\mathcal{A}_j, \mathcal{O}_j)$ et $(\mathcal{A}_l, \mathcal{O}_l)$ alors $\mathcal{A}_i = \{\mathcal{A}_j \cap \mathcal{A}_l\}$.

La phase d'apprentissage du système GALOIS possède deux modes : découverte de classe et prédiction de classe. Pour le premier mode, les objets sont insérés dans le treillis en tenant compte des valeurs des attributs et de la classe : c'est l'apprentissage supervisé. En ce qui concerne, l'apprentissage par prédiction de classe, les objets sont insérés sans tenir compte de leur classe. GALOIS utilise deux méthodes différentes pour attribuer une classe à un objet. Le système effectue une mesure de similarité entre le nouvel objet et chaque concept *cohérent* et *maximal*.

Apprentissage et Classement

La similarité entre un objet et un concept est le nombre de propriétés du concept vérifiées par l'objet. Le système assigne alors la classe d'objet au concept le plus similaire. GALOIS offre également un processus empirique qui attribue à chaque objet la classe majoritaire parmi les concepts *pertinents* déjà vérifiés.

Complexité et Comparaison expérimentale

Dans le pire des cas, la complexité en temps de GALOIS est égale à :

$$O((v + 1)^m \cdot v^m \cdot n) < O((v + 1)^{2m} \cdot n) \quad (2.5)$$

où v est le nombre maximal de valeurs possibles de l'attribut. La complexité en terme du temps de la construction progressive du treillis est comprise entre $O(1)$ et $O(l^2)$ où l est le nombre de concepts mise à jour dans le treillis de concepts.

2.6.7 RULEARNER

Principe et Construction du pseudo-treillis

Le système RULEARNER a été développé par [Sahami1995] basé sur les règles de classification extraites à partir du treillis. Il utilise la méthode du système GRAND décrit ci-dessus pour construire le treillis. Un paramètre N défini par l'utilisateur indiquant le pourcentage d'erreurs tolérées pour chaque règle. Au départ, tous les concepts sont considérés comme actifs. RULEARNER associe un label pour chaque concept en commençant par les nœuds instances qui reçoivent le label de la classe relatif à l'objet du concept. Ensuite, les nœuds internes reçoivent soit un label MIXTE si aucune classe n'est majoritaire, soit un label de classe si la classe est bien représentée.

2.6.8 Apprentissage et Classement

RULEARNER sélectionne les règles qui respectent l'ordre des antécédents (du général au particulier) pour classer les nouveaux exemples. La classe de la première règle vérifiée est affectée au nouvel exemple.

Complexité

La complexité du système a un ordre polynômiale de $O(nm^3)$.

2.6.9 CIBLe

Principe et Construction du pseudo-treillis

CIBLe (Concept Induction Based Learning) a été proposé par [Njiwoua and Mephu Nguifo1999]. Le système de classification supervisée se compose de trois étapes successives : (a) la construction d'un *sup-demi treillis*, (b) la sélection des concepts *pertinents* et la redescription du contexte en fonction des attributs des concepts *pertinents* (c) l'utilisation d'une mesure de similarité pour la classification de nouveaux exemples.

CIBLe adapte l'algorithme de **Bordat** [Bordat1986] pour construire le *sup-demi treillis* mais uniquement à partir des exemples positifs. La sélection des concepts *pertinents* est effectuée à l'aide d'une heuristique, et en se basant sur un paramètre spécifiant la hauteur du treillis. CIBLe est capable de traiter des problèmes avec plusieurs classes et de prendre en compte les données décrites par des attributs symboliques et numériques.

Apprentissage et Classement

CIBLe [Njiwoua and Mephu Nguifo1999] utilise l'algorithme de k -plus proche voisin pour prédire une classe d'un nouvel exemple. Les distances utilisées pour la mesure des similarités sont les suivantes : Mahalanobis, Manhattan et la distance euclidienne. La classe attribuée à l'objet est la classe majoritaire parmi les k plus proches voisins.

2.6.10 Complexité

La complexité du système CIBLe est influencée par deux phases importantes : la construction du *sup-demi treillis* et la recherche du seuil optimal. L'algorithme utilisé est basé sur l'algorithme de **Bordat** avec un paramètre supplémentaire qui limite la hauteur du treillis de Galois et donc

le nombre de nœuds générés. L'ordre supérieur du nombre de nœuds situés à la hauteur de $h=i+1$ est :

$$|L_{i+1}| \leq |L_i| \cdot \min(n-i, m-i) \quad (2.6)$$

Nous avons obtenu une borne supérieure pour la taille du *sup-demi treillis* égale à :

$$|L_i| \leq \sum_{i=0}^h (|L_i|) \leq \sum_{i=0}^h (\min(n-1, m-1))^i \leq (\min(n-1, m-1))^{h+1}$$

Par conséquent, la génération de chaque nœud est de l'ordre de $O(m^3)$, la complexité de l'algorithme dans le pire des cas est : $O(\min(n, m)^{h+1} \cdot m^3)$.

2.6.11 CLNB & CLNN

Principe et Construction du pseudo-treillis

CLNB [Xie *et al.*2002] (Concept Lattice Naive Bayes) et CLNN [Xie *et al.*2002] (Concept Lattice Nearest Neighbor) permettent de construire le treillis de Galois d'une manière partielle en partant du haut en bas, et qui se basent respectivement pour la phase de classification sur le réseau Bayésien (Naive Bayes) et l'algorithme de plus proche voisin (Nearest Neighbor). En ce qui concerne l'étape d'apprentissage, ce système se base sur la stratégie décrite ci-dessous.

Apprentissage et Classement

CLNB et CLNN appliquent des *règles contextuelles* pour prédire le type de classificateur à utiliser pour assigner une classe à un nouvel exemple. Un vote majoritaire est appliqué en cas d'une multitude de résultats retournés.

2.6.12 Complexité et Comparaison expérimentale

Les auteurs ne donnent aucune indication de la complexité théorique de leur système. Ils se réfèrent à des travaux antérieurs montrant que la complexité dépend du contexte initial. Bien qu'en théorie la complexité est exponentielle dans le pire des cas en fonction de la taille des données, mais le pire des cas est rarement rencontré dans la pratique.

2.6.13 Comparaison des systèmes à base du Treillis de Galois

Les six systèmes de classification basés sur le treillis de concepts présentés dans la Table.2.4, appliquent différentes méthodes soit pour la construction et l'apprentissage du treillis, soit pour la stratégie de classification.

Les démarches adaptées dans ces systèmes sont décomposées en deux types. La première méthode construit le treillis en sa totalité, et ensuite l'explore pour engendrer le classifieur. Les systèmes adaptant cette démarche sont : GRAND, GALOIS et RULEARNER. Ces méthodes ont en général la particularité d'utiliser un algorithme incrémental pour construire le treillis.

La seconde démarche adoptée par les systèmes LEGAL, CIBLE, CLNN et CLNB ne construit pas la totalité du treillis. Des heuristiques sont utilisées pour limiter la partie générée du treillis et permettre de réduire considérablement la taille de l'espace de recherche des hypothèses. Ces méthodes utilisent une démarche non incrémentale, descendante, et par niveaux, permettant d'élaguer le treillis à partir du biais d'apprentissage utilisé.

Apprentissage et Classement Les données organisées dans la structure du treillis de Galois peuvent être sous la forme binaire, attribut/valeur, symbolique et numérique. C'est le cas des systèmes CIBLe, CLNB et CLNN. Il faut noter que le traitement des attributs numériques pose un vrai problème. Dans CIBLe, les attributs ne sont pris en compte que lors du calcul de la similarité entre objets (phase de classement), alors que dans les deux autres systèmes une étape de discrétisation est effectuée au préalable.

CIBLe utilise des méthodes statistiques pour mesurer la quantité d'information présente dans chaque concept, les autres systèmes utilisent des méthodes empiriques basées sur d'autres mesures de sélection de l'information comme par exemple les notions de *support*, *complétude*, *cohérence*, *précision*, *rejet* et *maximalité*.

Les différents systèmes utilisent différentes formes de la connaissance apprise par le treillis. Quelques systèmes par exemple se basent sur les concepts sélectionnés (concepts pertinents) dans le treillis. Alors que GRAND, RULEARNER, CLNN et CLNB extraient des règles. Ces dernières ont l'avantage d'être plus compréhensibles et plus simples à interpréter par l'utilisateur. Le seul hic de cette méthode est que la relation d'ordre présentée par les liens de dépendance entre les concepts est perdue lors de la génération des règles. Il y a donc un choix à faire entre la compréhension et la richesse de la connaissance acquise à partir des données.

Pour le classement des nouveaux objets, la méthode *k-ppv* est la plus utilisée. Un vote majoritaire est appliqué sur la connaissance apprise (sous-ensemble de règles ou concepts) qui caractérise le mieux le nouvel objet.

Complexité et Expérimentations En terme de complexité, les systèmes CLNN et CLNB ne fournissent aucun élément d'information. Pour les autres systèmes, la complexité exponentielle apparaît clairement dans la formule, à l'exception de CIBLe pour lequel on a un polynôme de degré $h+4$ (h : la hauteur maximale du *sup-demi treillis*).

Systèmes	Treillis de Galois	Algorithme de Construction	Stratégie de classification	Complexité
GRAND	Complet	Oosthuizen	Vote Majoritaire	$\mathcal{O}(2^m \cdot m^4)$
LEGAL	Sup-demi treillis	Bordat	Vote Majoritaire	$\mathcal{O}(L .n.(1-\alpha))$
GALOIS	Complet	Carpineto-Romano	Mesure de similarité ou Vote Majoritaire	$\mathcal{O}((v+1)^{2m} \cdot n)$
RULEARNER	Complet	Oosthuizen	Règles de classification ou Vote Majoritaire	$\mathcal{O}(2^m \cdot m^4)$
CiBLe	Sup-demi treillis	Bordat	<i>k-nn</i>	$\mathcal{O}(m^{h+4})$
CLNN & CLNB	Sup-demi treillis	-	Règles de classification ou Vote Majoritaire	-

TABLE 2.7 – Tableau récapitulatif des propriétés des systèmes à base du treillis de concepts : v : le nombre maximal de valeurs qu'un attribut peut avoir, $|L|$: le nombre de classes dans le treillis à mettre à jour, m : le nombre de primitives dans chaque concept (*l'intension*), h : la hauteur maximale du *sup-demi treillis*, n : le nombre de concepts d'un certain seuil α , (-) non précisé.

2.6.14 Discussion

Nous avons passé en revue quelques systèmes se basant sur la structure du treillis de Galois. Le tableau.2.7 présente une synthèse des méthodes décrites en termes d'algorithmes d'apprentissage et de construction du treillis de Galois et en terme de leur complexité algorithmique. Pour la construction du treillis, deux approches sont utilisées dans ces systèmes : l'approche *incrémentale* et l'approche *non-incrémentale*.

À notre avis, l'approche incrémentale est plus adaptée dans le cas du traitement de grands volumes d'informations car il n'est pas nécessaire de reconstruire tout le treillis à chaque inser-

tion d'un nouvel objet. Donc, durant nos travaux nous avons adapté cette stratégie lors de la construction du treillis. Nous constatons que ces systèmes malgré qu'ils tournent autour d'une seule structure qui est le treillis de concepts, ils utilisent des techniques qui ne sont pas directement liées au treillis comme par exemple l'algorithme de plus proche voisin, les réseaux Bayésiens ou encore les règles *pertinentes* pour assigner une classe à un nouvel exemple.

En terme de complexité, le temps nécessaire pour la construction du treillis est fortement lié au nombre d'attributs utilisés et pour faire face au problème de la complexité, des heuristiques peuvent être appliquées pour réduire le nombre de nœuds du treillis ou encore de ne générer qu'un *sup-demi treillis* comme c'est le cas de plusieurs systèmes décrits ci-dessus. Les résultats expérimentaux présentés par les auteurs de ces systèmes montrent que le treillis de concepts est un outil très adapté aux problèmes de classification mais aussi au contexte de la fouille de données.

2.7 Conclusion

Dans ce chapitre, nous avons commencé par l'introduction des fondements théoriques et mathématiques du treillis de concepts. Nous avons présenté les algorithmes de construction et d'apprentissage du treillis afin de trancher entre eux et retenir uniquement ceux qui sont les plus adaptés à nos objectifs. Quelques systèmes se basant sur cette structure ont été énoncés et comparés dans cet état de l'art.

Comme nous l'avons détaillé dans l'introduction générale de ce manuscrit, notre intérêt durant cette thèse porte spécialement sur la structure du treillis de concepts qui a connu un fort essor depuis ces dernières décennies grâce à la prolifération de l'Analyse Formelle de Concepts. Nous nous sommes focalisés sur l'étude du treillis de concepts qui a prouvé sa robustesse d'une part, dans le contexte de la fouille de données et d'autre part, dans l'organisation très intuitive de données. Il s'agit d'une structure de données facile à explorer par une simple navigation entre ses concepts menant à un ensemble de chemins pertinents. En outre, la généricité de cette structure représente un point fort car elle représente une super structure de données pouvant contenir divers types de données (numériques, booléennes, symboliques...). Alors la question qui se pose dans ce cadre est la suivante : *Comment faire émigrer la structure du treillis de Galois du contexte de l'Intelligence Artificielle au contexte de la Vision par ordinateur ? Quelles sont les stratégies à adopter pour concevoir un système de reconnaissance de symboles à base du treillis ?* Ainsi dans le chapitre suivant, nous allons présenter notre chaîne de traitement qui s'inscrit dans le cadre de la reconnaissance de symboles graphiques en se basant sur le formalisme du treillis de Galois. Dans notre approche et dans le but de réduire la complexité de la structure du treillis de Galois, nous avons eu recours à un algorithme de construction à savoir l'algorithme de Norris qui s'est avéré le plus robuste par rapport aux autres algorithmes présentés dans ce chapitre. Dans nos travaux de thèse, nous avons adapté la structure du treillis de Galois dans divers contextes afin de bénéficier de ses capacités de hiérarchisation des données et sa facilité de navigation. Nous avons eu recours à la structure du treillis dans le cadre de la reconnaissance des objets graphiques (partie 2 du manuscrit) et dans le cadre de la localisation des symboles dans les documents graphiques (partie 3 du manuscrit).

Deuxième partie

Reconnaissance de Symboles
Graphiques

Reconnaissance de Symboles Graphiques

"On peut dire, sous forme de boutade, qu'il y a presque autant d'approches différentes des méthodes [...] qu'il y a d'applications différentes et de chercheurs participant à leur mise en œuvre, pour exprimer l'intérêt accru porté à ce domaine scientifique."

Alain Faure

Sommaire

1.1	Introduction	62
1.2	Approche Proposée	63
1.2.1	Représentation de symboles	64
1.2.2	L'approche sac de mots	67
1.3	Le Treillis de Galois et la Classification	68
1.3.1	Phase d'apprentissage	68
1.3.2	Construction du treillis de Galois	72
1.3.3	Comment attribuer une classe à un objet ?	75
1.3.4	Discussion	76
1.4	Résultats Expérimentaux	76
1.4.1	Protocole d'expérimentation	76
1.4.2	Mesures de performances	77
1.4.3	Résultats et Discussions	79
1.4.4	Évaluation de la taille du treillis de concepts en fonction du nombre d'attributs	81
1.4.5	Comparaison avec quelques classifieurs usuels	85
1.4.6	Évaluation de l'étape de représentation de symboles	85
1.4.7	Comportement de la taille du treillis de concepts en fonction de la taille du vocabulaire	94
1.5	Conclusion	94

1.1 Introduction

Notre objectif dans le cadre de cette thèse est la reconnaissance de formes et plus particulièrement la reconnaissance de symboles graphiques. La représentation des symboles est une tâche difficile en raison des distorsions que peut subir un objet, comme les occlusions, les déformations ou le bruit. En général, les propriétés souhaitables d'une approche de reconnaissance de symboles peuvent être divisées en deux étapes : la définition des descripteurs de formes expressifs et compacts, et la formulation de méthodes de classification robustes.

Commençons tout d'abord par l'étape de description des primitives. Dans ce contexte, de nombreuses méthodes de représentation ont été proposées [Frejlichowski and Forczmanski2010]. Le choix d'une méthode de représentation est généralement lié au type d'application considérée. Ce choix a une influence directe sur les performances du système de reconnaissance. Plus précisément, les méthodes locales basées sur l'extraction des points d'intérêt [Mikolajczyk and Schmid2005] ont montré une robustesse aux transformations d'échelle et aux occultations. Une fois qu'une bonne représentation des symboles graphiques est assurée, l'étape qui suit dans la chaîne de traitement d'un système de reconnaissance est l'étape de classification.

La classification supervisée est une tâche de fouille de données qui consiste à construire un classifieur à partir d'exemples étiquetés par leur classe (phase d'apprentissage), et ensuite à prédire la classe de nouveaux exemples avec le classifieur (phase de classement) [Boullé2009]. Les approches classiques de reconnaissance utilisent des classifieurs usuels comme le k -ppv, les arbres de décision et les réseaux Bayésiens qui ont déjà été appliqués avec succès à un grand nombre de problèmes en apprentissage et en reconnaissance de formes.

Dans le chapitre précédent, nous avons passé en revue plusieurs classifieurs utilisés dans le cadre de la reconnaissance de symboles graphiques. Ces classifieurs ont prouvé de bonnes performances dans diverses contributions. En particulier, nous nous intéressons au treillis de Galois qui a connu un fort essor depuis ces dernières décennies. Nous nous sommes inspirés du domaine de la fouille des documents textuels pour l'appliquer dans le cadre de la reconnaissance de symboles et des documents graphiques. Le cas des documents graphiques sera étudié dans la troisième partie de cette thèse.

L'intérêt croissant porté à l'Analyse Formelle des Concepts (AFC) depuis les années 2000 soit dans le domaine de la fouille de données, soit en représentation des connaissances, a fait émerger la structure du treillis des concepts. Comme nous l'avons explicité au chapitre précédent, le treillis de concepts [Ferré and Rudolph2009] est un graphe possédant la propriété du treillis qui se définit à partir de données organisées sous forme d'une table binaire $\{Objets * Attributs\}$. Les sommets du graphe sont des concepts - un concept est un regroupement maximal d'objets possédant des attributs en commun. Le treillis ainsi composé de concepts reliés par inclusion, fournit une représentation des données très intuitive. Une étude assez récente [Nguifo and Njiwoua2005] propose une comparaison de différentes méthodes de classification supervisée basées sur un treillis de Galois, où les expérimentations présentées montrent clairement que le treillis de Galois offre un cadre intéressant en classification.

L'étude de la littérature montre que les travaux basés sur le treillis de Galois sont rares et il existe peu de contributions dans le contexte de la reconnaissance d'où afin de positionner notre travail parmi les approches de l'état de l'art, nous avons choisi de la comparer avec celle proposée par [Guillas *et al.*2005] qui nous a parue l'approche la plus liée à notre travail. Dans cette perspective, la méthode Navigala [Guillas *et al.*2005] a été conçue pour reconnaître des symboles issus de documents techniques. À partir des images de symboles, des signatures ont été extraites. Les auteurs ont utilisé trois signatures statistiques (Radon, Fourier-Mellin et Zernike).

Dans ce contexte, nous distinguons deux étapes : l'apprentissage et la construction du treillis de

Galois utilisé comme classifieur dans notre travail et la stratégie de classification qui consiste à assigner une classe à un objet requête. Le treillis est constitué de tous les attributs constituant les objets graphiques. Ces attributs sont les mots visuels du dictionnaire visuel construit à partir de la base de symboles. L'utilisation des mots visuels comme des primitives permet d'évaluer le classifieur avec une approche symbolique qui n'a plus besoin de l'étape de discrétisation primordiale pour la construction du treillis. Notre méthode est comparée aux approches classiques, sans sac de mots et à plusieurs classifieurs usuels, évalués sur différentes bases de symboles. Nous montrons la pertinence et la robustesse de notre proposition pour la classification de symboles graphiques.

Dans ce chapitre, nous commençons dans la première section par une description détaillée des différentes étapes de la chaîne de traitement de reconnaissance que nous avons proposée dans le cadre de la résolution du problème de la classification supervisée. Les résultats du système de reconnaissance proposé sont étalés dans la dernière section ainsi qu'une comparaison avec les approches de l'état de l'art permettant d'évaluer l'approche proposée.

1.2 Approche Proposée

L'originalité de notre approche consiste à proposer une alliance entre l'Analyse Formelle de Concepts (AFC) et la reconnaissance de formes en combinant la représentation par sac de mots avec la structure du treillis de concepts. Notre approche présentée dans la Figure.1.1, consiste à adapter la structure du treillis de Galois dans le cadre de la reconnaissance de symboles graphiques et sa combinaison avec la représentation par sac de mots. Concernant les phases de capture de l'information et de représentation de données, nous exploitons l'approche par sac de mots où les symboles sont représentés par des mots visuels. Des primitives locales sont extraites à partir des symboles puis regroupées en clusters à l'aide de l'algorithme *k-means* où chaque cluster représente un mot visuel. Un vecteur de fréquences des mots visuels est construit et à chaque symbole, nous lui associons le groupe de mots visuels qui le décrit comme présenté dans la Figure.1.2. Cette méthode s'est avérée une alternative efficace de l'approche classique basée sur la signature numérique. Le formalisme du treillis de Galois que nous avons utilisé s'est avéré très performant dans le cadre de la reconnaissance de symboles car il permet la hiérarchisation de concepts constitués d'une partie *extension* représentant les objets graphiques et une partie *intension* représentant les attributs caractérisant ces derniers. Dans la section suivante, nous

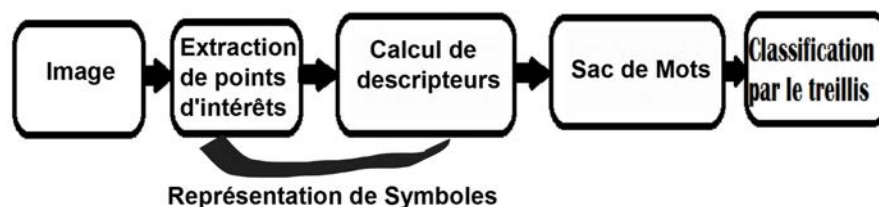


FIGURE 1.1 – Les différentes étapes de la chaîne de traitement proposée dans le cadre de la reconnaissance des symboles graphiques.

allons détailler les différentes étapes de la chaîne de traitement que nous avons proposée en commençant par la description de la méthode de représentation de primitives retenue, suivie par une section dédiée à la présentation de la stratégie de classification basée sur le treillis de Galois.

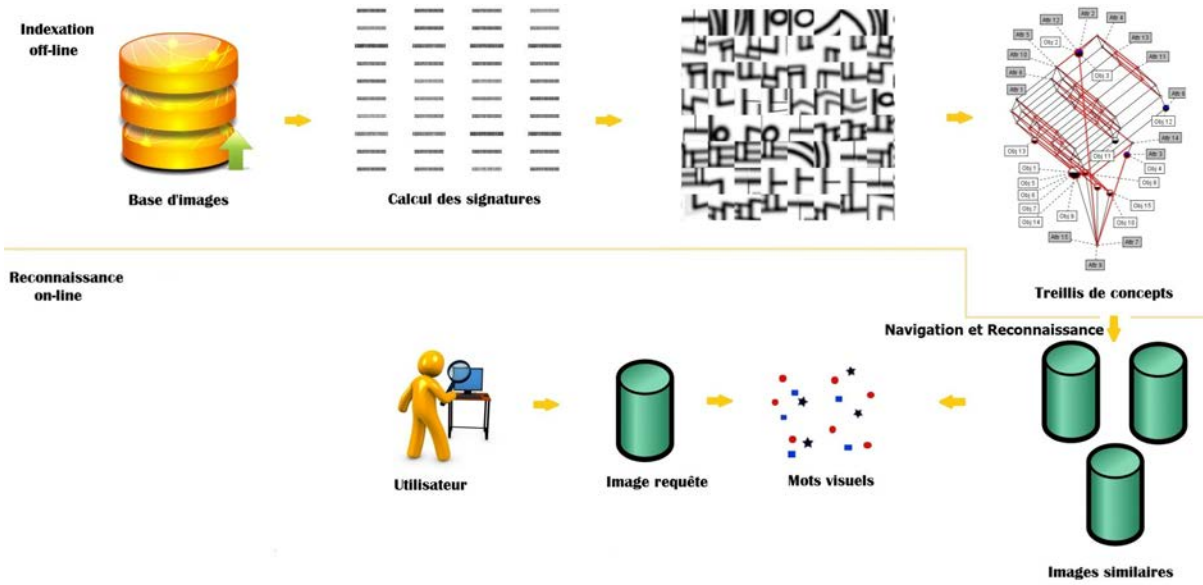


FIGURE 1.2 – Illustration de l’architecture du système proposé pour la reconnaissance de symboles graphiques.

1.2.1 Représentation de symboles

Le système de reconnaissance de symboles requiert l’utilisation de descripteurs riches et de classificateurs robustes. Dans cette section, nous détaillons les traitements appliqués aux symboles graphiques pour extraire les caractéristiques visuelles qui les composent. C’est le premier jeu de traitement nécessaire pour notre chaîne de reconnaissance. Dans notre approche, nous avons eu recours aux moments géométriques et aux descripteurs SIFT et SURF qui possèdent un pouvoir discriminant élevé permettant d’obtenir d’excellentes performances pour un problème de reconnaissance. Leur usage permet de réduire la complexité algorithmique tant à l’étape d’extraction qu’à celle de description.

Tout d’abord, les moments de Zernike et la R -signature sont des signatures statistiques que nous avons utilisées pour évaluer notre système de reconnaissance. Le choix de ces signatures est justifié par notre intérêt à évaluer notre approche par rapport aux contributions de l’état de l’art et plus précisément par rapport à l’approche de [Guillas *et al.*2005] qui est proche de notre système. Nous nous sommes orientés vers les approches statistiques car elles décrivent plus fidèlement la forme et elles sont invariantes aux transformations géométriques. Plus précisément, nous avons recours aux moments qui sont des descripteurs caractérisant l’agencement des pixels d’une forme. Étant donné une forme relative à une image discrète et une fonction de densité $f(\cdot)$, les moments d’ordre $p+q$ issus de cette image sont exprimés par :

$$\forall p, q \in N, \mathbf{M} = \sum_x \sum_y \Phi_{p,q}(x, y) f(x, y) \quad (1.1)$$

où x et y désignent les coordonnées cartésiennes des pixels et $\Phi_{p,q}(\cdot)$ est une fonction caractéristique.

Il existe de nombreuses fonctions caractéristiques qui ont été proposées afin d’obtenir des moments invariants aux transformations affines d’une image. Dans notre approche, nous avons utilisé deux moments géométriques dans le cadre de notre système de reconnaissance de symboles.

- Les moments de Zernike : permettent de maintenir les propriétés de la forme, et sont invariantes à la rotation, à l'échelle et aux déformations. Les moments de Zernike ζ reposent sur l'utilisation de polynômes complexes $\mathcal{Z}_{p,q}(\rho, \theta)$ considérés comme fonctions caractéristiques au niveau du disque unitaire. Nous avons retenu l'ordre 7 des moments de Zernike ce qui permet d'obtenir une signature de 24 valeurs numériques.
- Signature de Fourier-Mellin : dans notre cas et étant donné que nous allons utiliser le même protocole d'expérimentation adopté dans [Guillas *et al.*2005], nous avons fixé $p \in [0, 2]$ si $q=0$ et $p \in [-2, 2]$ si $q \neq 0$ et $q \in [0, 3]$, ce qui permet d'obtenir une signature à 33 valeurs numériques.
- Signature de Radon : nous avons choisi de faire varier θ pour obtenir 50 valeurs de la R -signature.
- SURF [Bay *et al.*2008] : le descripteur SURF proposé dans notre système est basé sur des propriétés similaires à SIFT, avec une complexité moins élevée et plus robuste aux différentes transformations d'images.

Extraction de points d'intérêt

La représentation de la forme est une tâche difficile parce que des distorsions peuvent affecter l'objet, comme par exemple les occlusions, les déformations élastiques et le bruit. Dans la littérature, plusieurs travaux se sont intéressés à la problématique de la reconnaissance de symboles, et plus particulièrement à l'étape de description des primitives à savoir [Tombre and Chhabra1998], [Cordella and Vento2000], [Blostein and Kwon2002], [Ogier and Trupin2003] et [Lladós and Kwon2004].

Dans la littérature, l'étape d'extraction des caractéristiques se base le plus souvent sur des détecteurs de points d'intérêt classiques où un descripteur local est calculé au voisinage de ces points. Des études récentes montrent la robustesse de ces méthodes dans les applications de traitement des images [Mikolajczyk and Schmid2004]. Récemment, il y a une tendance à l'utilisation des points clés ou les points d'intérêt locaux pour la recherche de symboles et pour la classification. Les points d'intérêt sont les primitives saillantes qui contiennent l'information locale riche d'un symbole quelconque, et peuvent être détectés automatiquement en utilisant différents détecteurs. Pour cela, nous avons expérimenté plusieurs algorithmes de détection de points [Mikolajczyk and Schmid2004] afin de se prononcer pour telle ou telle technique et pouvoir l'adapter dans notre contexte. Dans toutes ces approches, un point d'intérêt est défini comme étant un coin, un maximum ou un minimum local de l'intensité, ou aussi un point d'une courbe où la courbure est localement maximale. La qualité d'un détecteur de point est souvent jugée en terme de sa capacité à détecter une même région dans des symboles différents.

Notre choix pour les points d'intérêt comme primitives se justifie pas les nombreux avantages que présentent ces dernières : (a) les points d'intérêt présentent des sources d'informations plus fiables que les contours car il n'y a plus de contraintes sur la fonction d'intensité, (b) ils sont robustes aux occultations, (c) il n'y a pas d'opérations de chaînage comme est le cas des contours donc ils sont plus simples à détecter, (d) ils sont présents dans une grande majorité d'images qui ne peut pas être toujours le cas pour les contours. Afin de renforcer le pouvoir modélisant et discriminant du descripteur qui sera intégré dans notre chaîne de traitement, nous avons proposé une technique d'extraction de ces points à partir des objets graphiques. Nous avons évalué plusieurs détecteurs [Mikolajczyk and Schmid2004] utilisés dans la littérature.

Calcul de descripteurs

Dans notre approche, nous avons choisi de décrire les points d'intérêt déjà extraits par un descripteur local qui utilise la méthode des histogrammes de gradients orientés qui se base sur un calcul de gradient de façon simple et efficace. Le principe général est de découper l'objet à caractériser en plusieurs cellules pour lesquelles on comptabilise les occurrences de l'orientation du gradient dans un histogramme. Le descripteur SURF [Bay *et al.*2008] décrit l'intensité de pixels dans un voisinage autour de chaque point d'intérêt en se basant sur une distribution de réponses d'ondelettes de Haar dans le voisinage de ce dernier. Les descripteurs SURFs sont composés de 64 sommes de gradients au voisinage du point d'intérêt. Cet algorithme [Bay *et al.*2008] a eu un succès très important au sein de la communauté de vision, mais aussi en dehors de la communauté, et de nombreuses adaptations existent [Zhang2010].

La première étape de cet algorithme consiste à transformer l'image d'origine en image intégrale permettant d'accélérer le calcul de convolution et d'aires rectangulaires plus que l'image de départ. L'image intégrale \mathcal{II} est une image de même taille que l'image initiale \mathcal{I} avec $\mathcal{I}(x, y)$ est la valeur d'un pixel de l'image au niveau des coordonnées x et y . Chaque pixel de l'image intégrale correspond à la somme des pixels au dessus et à gauche de ce pixel dans l'image initiale. En d'autres termes, le pixel dans l'image \mathcal{II} est obtenu à partir des pixels de l'image de départ \mathcal{I} comme suit :

$$\mathcal{II}(x, y) = \sum_{x' \leq x, y' \leq y} \mathcal{I}(x', y') \quad (1.2)$$

L'Aire d'un rectangle de l'image de départ est normalement obtenue en calculant toutes les valeurs de pixels de ce dernier, alors qu'avec l'image intégrale il suffit de calculer les valeurs de ses quatre sommets comme suit :

$$Aire = \sum_{x_C \leq x' \leq x_D, y_C \leq y' \leq y_D} \mathcal{I}(x', y') = \mathcal{II}(C) + \mathcal{II}(D) - \mathcal{II}(B) - \mathcal{II}(A) \quad (1.3)$$

Ensuite, la matrice Hessienne est calculée afin de localiser les zones de fort changement d'intensité des pixels dans l'image. La matrice Hessienne est basée sur les dérivées partielles qui sont calculées par un produit de convolution avec des gaussiennes comme suit :

$$\mathcal{H}(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (1.4)$$

Un extremum est détecté si les valeurs de la matrice sont toutes les deux positives ou toutes les deux négatives. Les points d'intérêt sont alors localisés là où le déterminant de la matrice Hessienne est maximal. La réponse en x et en y des ondelettes de Haar est calculée dans un voisinage de $6s$ où s est l'échelle où le point d'intérêt a été trouvée. À partir de ces valeurs, l'orientation dominante de chaque point d'intérêt est calculée en faisant glisser une fenêtre d'orientation. Pour calculer le descripteur, un carré de taille $20s$ orienté selon l'orientation dominante est extrait. Cette zone est subdivisée en 4×4 carrés. Pour chacune de ces sous-régions, les ondelettes de Haar sont calculées sur 5×5 points. Soit dx et dy la réponse à l'ondelette de Haar, 4 valeurs sont calculées pour chacune des sous-régions : $\sum dx$, $\sum dy$, $\sum |dx|$ et $\sum |dy|$. Au final, chacun des points extraits à l'étape précédente est décrit par un vecteur composé de $4 \times 4 \times 4$ valeurs soit 64 dimensions (voir Figure.1.3).

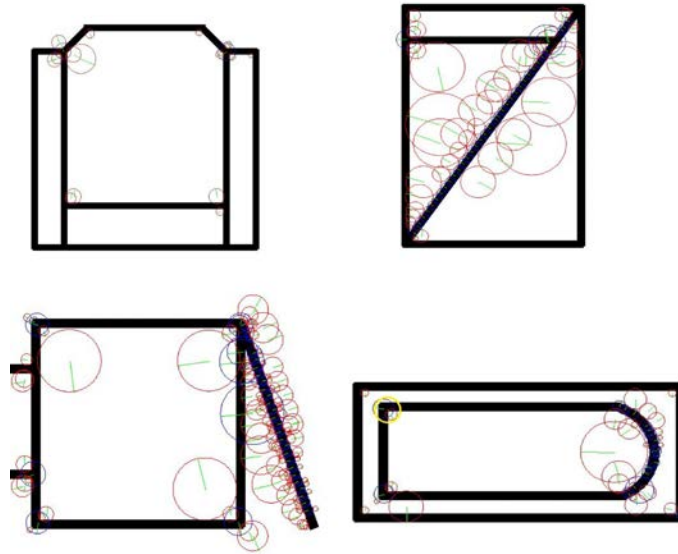


FIGURE 1.3 – Descripteur SURF appliqué sur quelques échantillons de symboles isolés.

1.2.2 L'approche sac de mots

L'utilisation de signatures numériques conduit à les discrétiser pour les céder ensuite à la table binarisée du treillis. Pour éliminer cette étape de discrétisation et en vue de renforcer l'approche symbolique, il nous est paru intéressant de pouvoir combiner le treillis de Galois avec l'approche par sac de mots pour atteindre des performances de reconnaissance très élevées. Alors, nous nous sommes basés sur le dictionnaire visuel pour transformer le problème de son aspect numérique à un aspect symbolique. Les mots visuels seront donc utilisés directement dans la table binarisée du treillis sans avoir recours à l'étape de discrétisation. Il est plus facile d'introduire les attributs directement dans la table binaire et la relation entre objets et attributs n'est plus une relation d'appartenance à un intervalle comme proposé dans [Guillas *et al.*2005], mais il s'agit d'une relation de $\{Présence/Absence\}$ d'un mot dans l'objet. L'affirmation de la présence ou de l'absence d'un mot dans un objet est mesurée par un critère de sélection que nous avons choisi et qui sera détaillé par la suite. Tout en tentant de conserver une approche générique, le dictionnaire visuel est utilisé dans notre travail en vue de combiner une approche symbolique de description de symboles avec le treillis de Galois. Ce contexte de classification permet d'évaluer notre méthode dont le principal objectif est d'éliminer l'étape de discrétisation de signatures numériques avant la construction du treillis.

Le dictionnaire visuel consiste en une opération de quantification vectorielle des descripteurs numériques pour représenter chaque caractéristique par le centroïde du groupe (ou cluster) qui le contient. Nous avons utilisé l'algorithme *k-means* présenté dans le chapitre.1. Ainsi, chaque image de la base sera représentée par un histogramme de fréquences de ces mots visuels. Après avoir construit le dictionnaire visuel de la base de symboles, nous pouvons procéder à l'extraction des vecteurs de mots visuels représentatifs de chaque symbole de la base. Le sac de mots (voir Figure 1.4) consiste à trouver les occurrences d'apparition des mots dans le dictionnaire visuel pour chaque symbole. Ainsi, à chaque symbole on lui associe un vecteur des mots visuels ainsi que les poids représentant les fréquences d'apparition de ces mots.



FIGURE 1.4 – Vocabulaire visuel ayant une taille $|\Omega|=270$ mots visuels construit sur les échantillons de symboles graphiques de la base GREC2003.

1.3 Le Treillis de Galois et la Classification

Dans cette partie, nous allons présenter les stratégies de construction, d'apprentissage et de classification basées sur le treillis de Galois que nous avons retenues dans le cadre de cette thèse. Ensuite, nous allons évaluer notre approche de reconnaissance d'objets graphiques isolés de leurs contextes en la positionnant parmi les travaux de l'état de l'art. Malgré que le treillis de concepts soit un sujet d'actualité et a connu un bond ces dernières décennies dans le domaine de la fouille du texte, le nombre de travaux qui s'y intéressent surtout dans la communauté de la reconnaissance de formes reste très restreint. Nous avons choisi le treillis de concepts comme classifieur car il nous a paru une solution élégante et efficace de représentation des données. Il offre un aspect sémantique vu que les objets et les propriétés qu'il représente sont reliés par des relations ce qui permet de renforcer le pouvoir discriminant du classifieur. Dans le cadre de notre système, l'exécution de la chaîne de traitement de reconnaissance de symboles conduit à assigner une classe à un nouvel objet passé en requête et lui associer les instances les plus ressemblantes parmi l'ensemble des objets présentés dans le treillis de concepts. Pour atteindre des performances de reconnaissance très satisfaisantes, il est nécessaire d'apprendre le classifieur en regroupant les informations et les configurations des objets de la base.

1.3.1 Phase d'apprentissage

Dans le deuxième chapitre de ce manuscrit, nous avons dressé un panorama des algorithmes de manipulation du treillis de Galois. Malgré que de nombreux travaux s'intéressent à cette structure, la plupart des méthodes n'offrent pas de réponse au problème de la reconnaissance puisqu'elles se focalisent sur le domaine de l'analyse formelle et non pas sur le domaine de la reconnaissance de formes. Les premiers travaux formels sur les treillis de concepts se situent en algèbre universelle, à travers les travaux de Birkhoff [Birkhoff1967], et plus tard les travaux de Barbut [Barbut and Monjardet1970]. Les travaux de Duquenne [Duquenne1999], de Ganter [Ganter1996] ont permis de montrer l'efficacité de cette structure pour l'analyse de données.

En classification supervisée, l'apprentissage se ramène dans le cas général à un problème de recherche dans l'espace des hypothèses. Malgré la complexité exponentielle sous-jacente de la construction de la structure du treillis de concepts, de nombreux techniques de classification ont été développées et ont produit des résultats comparables à ceux des méthodes

standards en terme de précision qu'en terme d'efficacité en temps. En outre cette structure présente des propriétés particulières qui favorisent son usage pour la fouille de données, notamment pour la recherche de règles d'association [Pasquier *et al.*1999] et pour l'apprentissage de concepts [Kourie and Oosthuizen1998]. Dans le cas général, le problème de l'apprentissage supervisé est exprimé par un ensemble de données d'apprentissage contenant des exemples (encore appelés tuples, objets, instances ou observations) décrits par un ensemble de propriétés (encore appelés attributs ou descripteurs), et appartenant à une ou plusieurs classes.

Nous présentons une méthode supervisée de classification d'objets graphiques basée sur l'exploitation des treillis de Galois. L'Analyse Formelle de Concepts se fait à partir de relations entre des objets graphiques et leurs propriétés ou caractéristiques extraites de l'ensemble d'apprentissage. Le treillis de Galois résultant fournit des concepts qui sont utilisés pour assigner la classe la plus vraisemblable d'un nouvel objet. En général, l'apprentissage supervisé améliore le processus de classification des symboles isolés. Il semble nécessaire donc de s'appuyer sur un travail dans ce contexte pour évaluer nos contributions. Dans cette section, nous allons analyser le treillis de Galois utilisé en rapport avec le traitement de symboles et plus précisément le travail de [Guillas *et al.*2005].

Dans [Guillas *et al.*2005], les deux étapes principales de la phase d'apprentissage sont la *discrétisation* des signatures numériques et la *construction* du treillis de Galois. Les données utilisées dans [Guillas *et al.*2005] sont numériques ce qui rend difficile de distinguer les objets, néanmoins pour notre approche nous allons utiliser des données symboliques pouvant être cédées directement au treillis sans étape préalable de discrétisation. Un gain en complexité est alors attendu par le recours aux données symboliques au lieu des données numériques.

Passons maintenant à l'étude de la méthode décrite dans [Guillas *et al.*2005] afin de la comparer à la notre. En effet dans les travaux de [Guillas *et al.*2005], les symboles graphiques sont représentés par un descripteur numérique. Chaque signature est discrétisée en un certain nombre d'intervalles suivant un critère de découpage [Coustaty *et al.*2010] comme par exemple la distance d'Entropie. L'algorithme d'apprentissage proposé par [Guillas *et al.*2005] est décrit ci-dessous. L'approche de reconnaissance de symboles présentée dans [Guillas *et al.*2005]

Algorithm 5 Apprentissage du treillis de concepts dans [Guillas *et al.*2005]

Entrées un ensemble d'objets \mathcal{O} où chaque objet $o \in \mathcal{O}$ est décrit par une signature numérique $o=(v_1, \dots, v_n)$, les valeurs de ce vecteur sont normalisées par la valeur de la classe $c(o)$.

Sorties un treillis de Galois (\mathcal{G}, \leq) décrit une hiérarchie de concepts reliés par une relation d'inclusion \leq .

Paramètres critère de coupe de la discrétisation (l'entropie, la distance maximale, le coefficient de Hotelling...).

se repose sur une première étape de normalisation de données en appliquant la loi normale à chaque attribut de la signature. Cette étape s'avère indispensable pour leur approche car les auteurs utilisent des signatures numériques comme la signature de Radon, les moments de Zernike... ce qui rend difficile de les considérer dans un même ordre de grandeur sans les normaliser. Une fois les données normalisées, une étape de discrétisation est alors appliquée pour découper les signatures des objets en un ensemble d'intervalles \mathcal{I} disjoints et couvrant l'ensemble des valeurs prises par tous les objets \mathcal{O} . Un constat global après

l'étude de cette approche est que la discrétisation des signatures et leurs transformations en intervalles figés peut conduire à une perte d'information sur les objets graphiques ce qui affecte sûrement le pouvoir discriminant des descripteurs et par la suite dégrade les performances du classifieur.

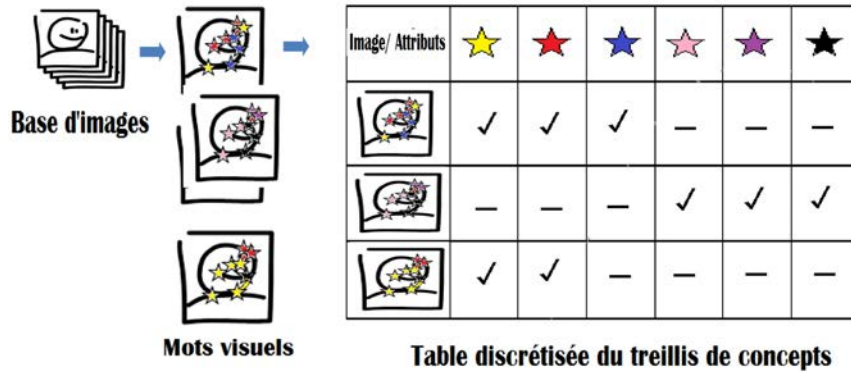
Pour faire face à cette problématique, il nous est paru intéressant de modéliser les attributs par l'approche sac de mots. Dans un premier temps, pendant la phase d'apprentissage, les relations entre les ensembles de symboles et les caractéristiques sont structurées et hiérarchisées à travers le formalisme du treillis de Galois. La table binarisée du contexte formel de notre problème est construite en se basant sur la relation $\{Présence/Absence\}$ d'un attribut (un mot visuel) dans un objet. En outre, ce choix de modéliser des mots dans la table nous facilite par la suite la mesure de similarité entre les attributs du symbole requête et les attributs des symboles appris dans le treillis. Grâce à cette approche, il est possible de mesurer la similarité entre les attributs d'une façon très simple en la comparant aux distances utilisées généralement dans les autres travaux. La table discrétisée du treillis est alors composée de l'ensemble des objets graphiques en relation d'appartenance avec l'ensemble des mots visuels caractérisant une classe de symboles. Notons que cette méthode nécessite un temps de calcul excessif et une complexité algorithmique assez élevée. Dans ce qui suit, nous justifions nos choix pour l'algorithme d'apprentissage et nous présentons une illustration des méthodes adaptées pour la construction et la stratégie de classification adoptée basée sur le treillis de concepts.

Dans ce cadre et afin d'atteindre des performances acceptables, il semble nécessaire d'éliminer ces étapes de prétraitement des données avant leur introduction dans le treillis de Galois. L'originalité de notre approche consiste à éliminer l'étape de discrétisation des attributs par l'utilisation de la représentation symbolique des objets. Un symbole sera représenté par un vecteur de mots visuels et le treillis de concepts est construit à partir de la relation binaire entre les mots du dictionnaire visuel et l'ensemble des symboles graphiques et non pas la relation entre les symboles et les intervalles des signatures.

Le principe comme nous l'avons décrit ci-avant, consiste à décrire les symboles à l'aide d'un ensemble de descripteurs, qui correspondent à des caractéristiques de bas niveau. Une fois le vocabulaire visuel est généré, les symboles de la base peuvent être représentés par des ensembles, des histogrammes ou des vecteurs de termes visuels. Par ailleurs, les mots visuels peuvent être pondérés afin de représenter leur importance. Nous utilisons les poids afin d'évaluer l'importance d'un mot visuel dans un symbole. L'importance augmente avec l'augmentation du nombre d'apparition de ce mot dans le symbole graphique.

Une simple approche utilisée pour représenter la relation $\{Objets * Attributs\}$ est présentée dans la table (voir Figure.1.5). La table discrétisée est représentée alors par un ensemble d'objets en relation avec l'ensemble des mots visuels qui les caractérisent. Si l'objet o_1 (ligne 1 sur la Table.1.5) est décrit par le vecteur $(M_1 (\star)^3, M_2 (\star)^4, M_3 (\star)^5)$ alors dans la table discrétisée du treillis de concepts décrit les relations suivantes : $\mathcal{R}(o_1, M_1)=1$, $\mathcal{R}(o_1, M_2)=1$, $\mathcal{R}(o_1, M_3)=1$ (sur la Figure.1.5, cette relation est modélisée par " \checkmark "), $\mathcal{R}(o_1, M_4)=0$, $\mathcal{R}(o_1, M_5)=0$ et $\mathcal{R}(o_1, M_6)=0$ (sur la Figure.1.5, ces relations sont modélisées par "-"). Le treillis de Galois de la table est construit, hiérarchisant ainsi les ensembles de caractéristiques les uns par rapport aux autres. Un concept ne comporte dans son *extension* que les nouveaux objets relatifs à ses descendants, et dans son *intension* que des nouvelles caractéristiques relatives à ses ascendants.

Dans le deuxième chapitre de ce manuscrit, nous avons passé en revue quelques algorithmes de construction utilisés dans les systèmes basés sur le treillis de Galois. Au regard de remédier au problème de la complexité algorithmique d'un treillis de Galois, nous avons recours

FIGURE 1.5 – Table binarisée du contexte $\mathcal{K}=(\mathcal{O}, \mathcal{A}, \mathcal{R})$.

à un algorithme assez efficace et qui sera détaillé par la suite.

Discussion

Les deux principales phases dans une approche classique de classification basée sur le treillis de Galois sont la discrétisation des signatures numériques et la classification. Il est à noter que l'étape de discrétisation n'est plus nécessaire dans notre approche qui est basée sur la représentation par sac de mots ce qui nous permet de diminuer le temps de traitement ainsi que l'espace mémoire requis souvent avec l'étape de discrétisation. En effet, pour les données symboliques, les symboles peuvent être directement distingués les uns des autres parce qu'ils ont des caractéristiques bien définies. Alors que pour les approches classiques [Guillas *et al.*2005], la discrétisation est nécessaire et implique la création d'intervalles disjoints. Les objets seront donc distingués, dans ce cas par le fait qu'ils ont des valeurs appartenant à tel ou tel intervalle [Coustaty *et al.*2010]. Pour cela, nous nous sommes appuyés sur les relations de $\{Présence/Absence\}$ entre les objets et l'ensemble des mots visuels qui les caractérisent.

Le constat global est que la discrétisation des signatures en intervalles disjoints peut entraîner une perte d'information ou du pouvoir discriminant de la signature. De plus, le critère de coupe utilisé pour pouvoir définir le point de coupe d'un intervalle est déterminant pour les performances du classifieur. Par exemple le *coefficient de Hotelling* s'est avéré plus performant et fournit de meilleurs résultats que l'entropie et la distance maximale. Les performances du classifieur sont alors fortement liées au choix du critère de coupe. Afin d'amorcer la reconnaissance, notre réflexion nous a emmenés à utiliser le vocabulaire visuel comme une méthode de représentation des objets graphiques. La taille du treillis de concepts construit avec la méthode utilisant les descripteurs numériques comme des attributs, est très élevée tandis que le treillis de concepts construit avec la représentation par sac de mots est plus petite vu le nombre réduit des concepts générés ce qui diminue le problème de la complexité due à sa taille. Aucun critère n'est utilisé lors de la phase de construction du graphe car le treillis présente toutes les combinaisons possibles entre les objets et les mots visuels qui les caractérisent. Plusieurs études ont montré que la taille du treillis est bornée par $2^{|\mathcal{O}+\mathcal{A}|}$ dans le pire des cas, et par $|\mathcal{O} + \mathcal{A}|$. Mais la taille du treillis reste toujours raisonnable en pratique d'après [Nguifo and Njiwoua2005]. Après les deux phases décrites ci-dessus respectivement à savoir le prétraitement des données et la construction de la table binarisée du contexte formel, dans ce qui suit, nous allons décrire

l'étape de construction du treillis et l'étape de classification qui consiste à attribuer une classe au symbole requête.

1.3.2 Construction du treillis de Galois

Comme nous l'avons largement détaillé dans le deuxième chapitre de ce manuscrit, il existe de nombreux algorithmes de construction du treillis de concepts. Ces algorithmes sont classés en deux catégories : les approches incrémentales et les approches non incrémentales. La taille du treillis dépend de la taille de données, mais aussi du contenu du contexte. Plusieurs études comparatives de la complexité algorithmique du treillis ont été proposées, le lecteur peut se rendre aux travaux de [Guénoche1990], [Godin and Missaoui1994] et [Njiwoua and Mephu Nguifo1997]. Pour remédier à la problématique de la complexité du

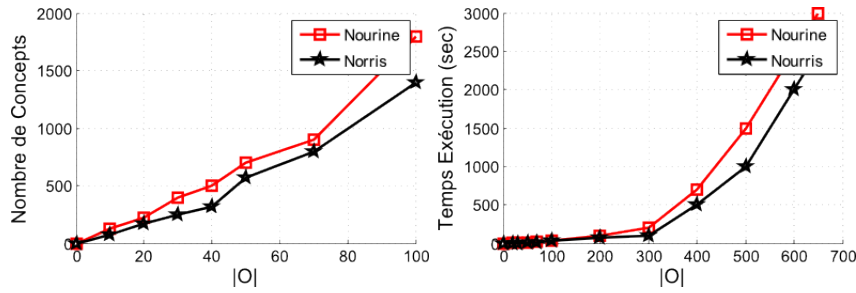


FIGURE 1.6 – Comparaison entre les deux algorithmes de construction du treillis de concepts : Nourine et Norris en terme du nombre de concepts et le temps nécessaire pour leur génération.

treillis, nous avons recours à l'algorithme de Norris qui s'est avéré le plus efficace et le plus rapide par rapport aux autres algorithmes comme celui de Nourine (voir Figure.1.6). Nous présentons l'algorithme de Norris [Norris1978] que nous avons modifié pour qu'il prenne en compte l'insertion des objets en plus de leurs intensions. Cet algorithme a montré son efficacité et son adaptabilité aux grands volumes de données. L'algorithme de Norris parcourt la table binaire du treillis de concepts ligne par ligne et construit l'ensemble des rectangles maximaux \mathcal{L}^k obtenus après examen de k lignes, avec $k \in [1, n]$. \mathcal{L}^1 contient uniquement le rectangle $x_1 \times f(x_1)$. Le principe de cet algorithme consiste à construire \mathcal{L}^k connaissant x_k et \mathcal{L}^{k-1} . \mathcal{L}^k est construit à partir de x^k en considérant tous les rectangles de \mathcal{L}^{k-1} .

Considérons le rectangle $(X_i \times Y_i)$, si Y_i est inclus dans $f(x_k)$, alors le rectangle $(X_i \times Y_i)$ est étendu en $(X_i \cup \{x_k\}) \times Y_i$. Dans le cas contraire (si Y_i n'est pas inclus dans $f(x_k)$), le rectangle $(X_i \cup \{x_k\}) \times (Y_i \cap f(x_k))$. Enfin, si le rectangle $x_k \times f(x_k)$ est maximum dans \mathcal{L}^k , il est alors ajouté.

Pour illustrer le déroulement du processus d'insertion d'un nouvel objet dans le treillis de concepts, nous proposons l'exemple suivant qui explique les différentes étapes de la construction du treillis à chaque nouvelle entrée.

Soit req le nouvel objet à insérer dans le treillis de concepts appartenant à la classe $\mathcal{C}(req)$. La comparaison entre les attributs de req et les attributs du nœud courant \mathcal{N}_i appartenant à la classe $\mathcal{C}(\mathcal{N}_i)$ aboutit à un ajout d'un nœud new ayant comme attribut l'intersection entre les attributs de req et les attributs du nœud courant \mathcal{N}_i .

$$\mathcal{A} = \frac{Attr(req) \cap Attr(\mathcal{N}_i)}{Attr(req) \cup Attr(\mathcal{N}_i)} \quad (1.5)$$

Algorithm 6 Construction du treillis de Galois

$\mathcal{G} := \emptyset$
 Pour chaque symbole $s \in \mathcal{S}$ Faire
 Insertion(s, \mathcal{G})
 \mathcal{G} est le treillis de Galois : $\bigcup(X_i, Y_j)$
 X_i est la partie *extension* d'un concept \mathcal{C}_i
 Y_i est la partie *intension* d'un concept \mathcal{C}_i
Insertion(s_k, \mathcal{G})
 $\forall k \in K$ (l'ensemble des modèles de l'échantillon d'apprentissage) Faire
 $\forall (X_j, Y_j) \in \mathcal{G}$ Faire
 si $Y_j \subset Attr(s)$ Faire
 $(X_j, Y_j) \leftarrow (X_j \cup s) \times Y_j$
 marquer s
 sinon Faire
 si $(X_j \cup Attr(s)) \times (Y_j \cap Attr(s))$ est maximum alors
 $\mathcal{G} := \mathcal{G} + (X_j \cup Attr(s)) \times (Y_j \cap Attr(s))$
 FinPour
 si s n'est pas marqué Alors $\mathcal{G} := \mathcal{G} + (s, Attr(s))$
 FinPour

$$new \in \mathcal{C}(Attr(new)) \quad (1.6)$$

La Figure.1.7 montre un exemple illustratif du processus d'insertion d'un nouvel objet

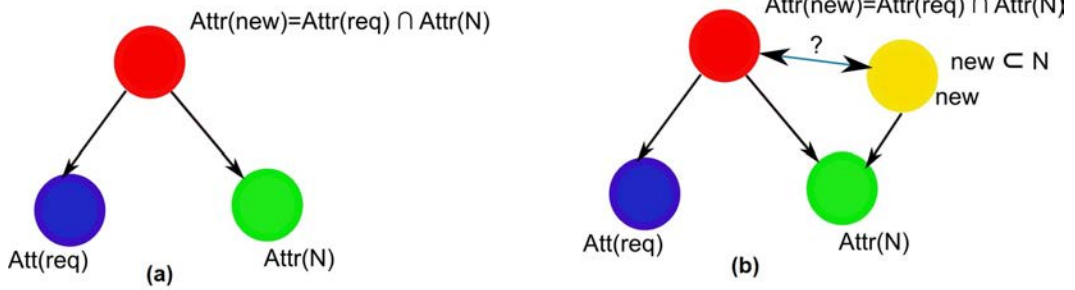


FIGURE 1.7 – Un exemple illustratif du processus incrémental d'insertion d'un nouvel objet lors de la phase d'apprentissage et de construction du treillis de Galois.

dans le treillis de concepts et toutes les étapes qui précèdent son ajout à savoir l'apparition d'un nouveau nœud dont les attributs sont le résultat de l'intersection entre les attributs du nouvel objet et le nœud courant (cet exemple a été inspiré de la thèse d'Emmanuel Zenou). Comme nous pouvons le voir dans la Figure.1.7, il y a trois cas possibles qui résultent suite à l'insertion d'un nouvel objet req .

1. *si* $\mathcal{C}(req) = \mathcal{C}(N_i)$, *alors* $\mathcal{C}(new) = \mathcal{C}(req)$ (Figure.1.7(a)).
2. *si* $N_i = N_0$ (concept initial), *alors* $\mathcal{C}(new) = \mathcal{C}(req)$ (Figure.1.7(b)).
3. *sinon* $\mathcal{C}(new) = 0$.

Une deuxième étape illustrée dans la Figure 1.8, est de mettre à jour les classes des nœuds du treillis en se basant sur le majorant \mathcal{N}_j du nœud courant \mathcal{N}_i comme suit :

1. *si* $Attr(\mathcal{N}_j \subset Attr(new) \subset Attr(\mathcal{N}_i))$ alors $\mathcal{C}(\mathcal{N}_j) = 0$ (Figure.1.8(a)).

2. si $\mathcal{C}(\text{new})=0$ alors $q(\mathcal{N}_j)=0$;
3. si $\text{Attr}(\text{new}) \subset \text{Attr}(\mathcal{N}_j) \subset \text{Attr}(\mathcal{N}_i)$ alors pas de modification à apporter (Figure.1.8(b)).
4. si $\text{Attr}(\text{new})=\text{Attr}(\mathcal{N}_j) \subset \text{Attr}(\mathcal{N}_i)$ alors
si $\text{Attr}(\text{new})=0$ alors $\mathcal{C}(\mathcal{N}_j)=0$ (Figure.1.8(c)).
5. si $\text{Attr}(\text{new})$ et \mathcal{N}_j sont incomparables, alors aucune modification à apporter (Figure.1.8(d)).
6. si $\text{Attr}(\text{new})=\text{Attr}(\mathcal{N}_j)$ alors
si $\mathcal{C}(\text{new}) \neq \mathcal{C}(\mathcal{N}_i)$ alors changer la classe de \mathcal{N}_i et $c(\mathcal{N}_i)=0$ (Figure.1.8(e)).
7. si $\text{Attr}(\text{new})=\text{Attr}(\text{req})$ alors la classe de $\mathcal{C}(\mathcal{N}_i)$ n'est pas changée (Figure.1.8(f)).

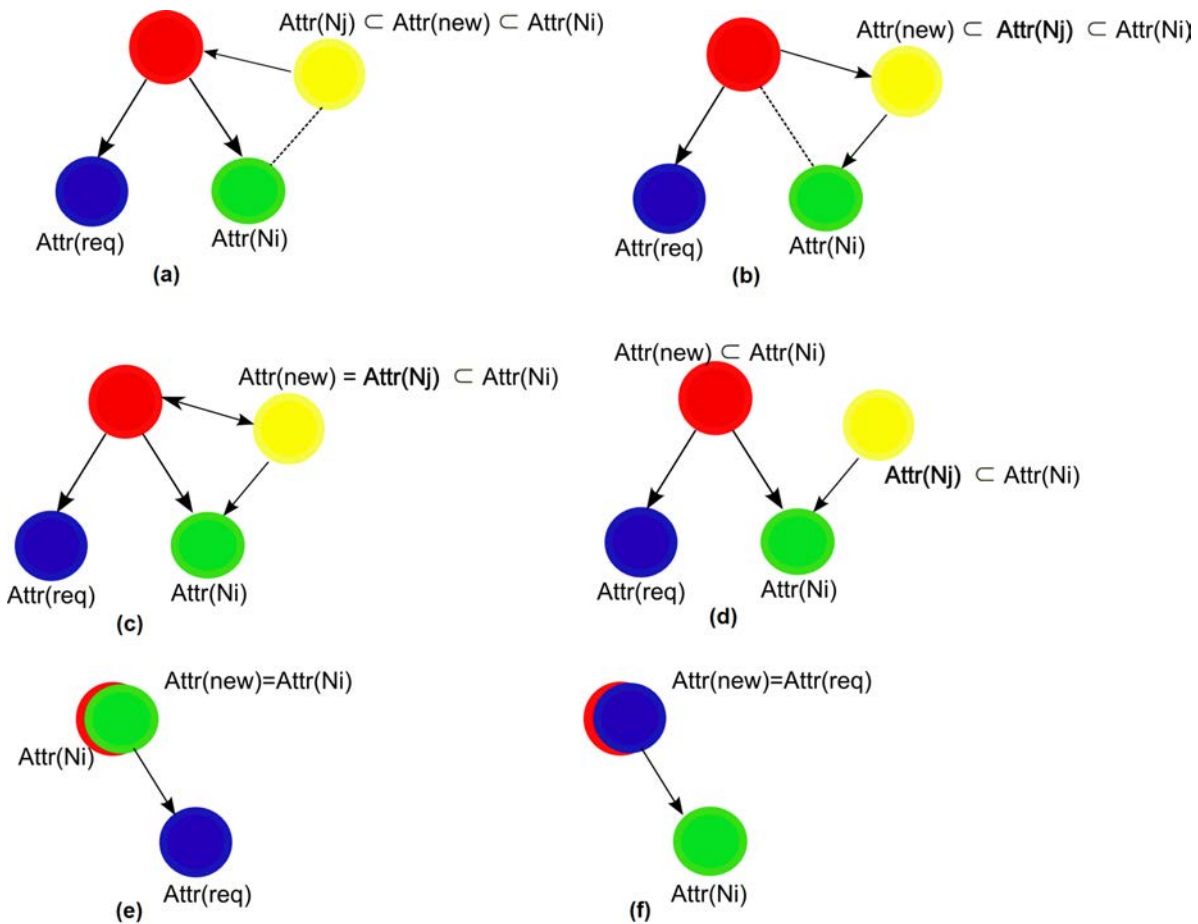


FIGURE 1.8 – Un exemple illustratif de l'étape de la mise à jour des classes des nœuds du treillis suite à l'insertion d'un nouvel objet lors de la phase d'apprentissage.

Une caractéristique du treillis qui justifie notre choix pour un tel classifieur est qu'il n'y a pas de configuration de paramètres à considérer pour sa construction. Notre treillis modélise toutes les combinaisons possibles entre les objets et les mots visuels qui sont liés par la relation binaire (*Présence/Absence*). Dans notre cas, on ne s'intéresse plus aux valeurs numériques mais aux valeurs booléennes cela veut dire qu'on s'intéresse seulement à l'absence ou à la présence des mots visuels dans les symboles. Une fois le classifieur est construit,

passons maintenant à la phase de classification des objets graphiques qui constitue le point central de cette première contribution. L'objectif de la reconnaissance consiste à déterminer la classe à laquelle appartient un nouvel exemple ainsi que les instances qui lui sont semblables.

1.3.3 Comment attribuer une classe à un objet ?

L'objectif de cette contribution étant la classification supervisée de symboles graphiques. Un symbole s dans notre approche ne peut appartenir qu'à une seule classe $\mathcal{C}=\mathcal{C}(s)$. Notre problème est un problème multi-classes (nous avons n classes ou modèles de symboles).

Généralement dans le cadre des problèmes multi-classes, nous distinguons deux types de classification :

- La classification "*chacun pour soi*" où toutes les classes sont prises en compte.
- La classification "*seul contre tous*", où deux valeurs sont attribuées à savoir la valeur "positive" pour la classe considérée et la valeur "négative" pour le reste des classes.

D'autres types de classification peuvent exister en fonction de l'application et le contexte du travail.

Soit \mathcal{K} un contexte et \mathcal{Q} l'ensemble des concepts associés. Un concept $c \in \mathcal{Q}$ appartient à une classe \mathcal{C} si et seulement si tous les objets de son *extension* appartiennent à la classe \mathcal{C} . Le concept c est alors dit classé. Le concept ne possédant pas de classe a une classe nulle : $\mathcal{C}(c)=0$.

Notre objectif étant d'associer une classe à un symbole requête pour cela nous avons besoin d'une mesure de similarité entre les attributs de la requête et les attributs des concepts du treillis (il s'agit de la partie *intension* du concept).

Comme nous l'avons détaillé, dans notre approche chaque nœud du treillis résultant de la phase d'apprentissage est composé d'un sous-ensemble d'objets graphiques (*extension*) et un sous-ensemble de mots visuels (*intension*). La partie *intension* ne contient que les mots visuels partagés par l'ensemble des symboles, et, de même, la partie *extension* contient uniquement les symboles qui partagent le sous-ensemble de mots ou des attributs. L'ensemble des concepts est ordonné par la relation d'inclusion qui caractérise la structure du treillis de Galois. Une des primitives de la structure du treillis qui justifie notre intérêt pour un tel classifieur est que dans le treillis et au cours de la phase de classification, plusieurs chemins peuvent mener à une classe d'objet (plusieurs scénarii existent) ce qui permet de renforcer le pouvoir discriminant souhaité pour n'importe quel classifieur.

Dans notre chaîne de traitement de reconnaissance, la méthode utilisée pour assigner une classe à un nouvel objet consiste à calculer la similarité entre le vecteur de mots visuels représentant l'objet requête et chaque partie *intension* des concepts cohérents et maximaux du treillis. Plusieurs travaux ont proposé des mesures de similarité entre des objets comme par exemple l'utilisation des distances [Kodratoff and Tecuci1988] ou autres mesures comme dans [Foggia et al.1999]. L'originalité de notre approche est l'utilisation des valeurs symboliques dans la table binaire du contexte formel ce qui permet d'utiliser d'autres mesures de similarité entre les ensembles d'attributs et éviter les calculs exhaustifs dus aux valeurs numériques.

Similarité entre un symbole requête et les concepts

L'usage du treillis de Galois comme une structure de hiérarchisation de concepts a pour but de regrouper les objets partageant les mêmes attributs dans les mêmes concepts. Donc

la classification d'un nouvel objet revient à mesurer la similarité entre les attributs de cet exemple avec les attributs de l'ensemble d'objets appris dans le treillis de concepts. La similarité entre les attributs de la requête req et l'ensemble des attributs ou les *intensions* du nœud courant \mathcal{A}_i appris dans le treillis de concepts est mesurée de la façon suivante :

$$Sim(req, \mathcal{A}_i) = \frac{m_c}{m_{req} + m_{\mathcal{A}_i}} \quad (1.7)$$

avec :

- m_c est nombre de mots en commun
- $m_{\mathcal{A}_i}$ les attributs de la partie *intension* du concept \mathcal{A}_i .
- m_{req} les attributs de la partie *intension* de l'objet req .

Cette équation peut aussi se calculer comme suit :

$$Sim(req, \mathcal{A}_i) = \frac{card(intension(req) \cap intension(\mathcal{A}_i))}{card(intension(req) \cup intension(\mathcal{A}_i))} \quad (1.8)$$

Le système attribue ensuite à l'objet req la classe des objets les plus similaires inclus dans l'*extension* du concept retenu. Dans le cas où plusieurs classes sont trouvées, le système opte pour un processus de vote majoritaire qui attribue à chaque objet la classe majoritaire parmi les concepts pertinents retenus. Le concept dont le nombre d'attributs communs de sa partie *intension* est le plus grand est celui retenu. Un concept partageant m attributs communs avec la requête reçoit 2^m-1 votes.

$$\mathcal{C}(req) = max(Sim(\mathcal{C}_k, req)) \quad (1.9)$$

k étant le nombre de concepts pertinents retenus après la phase de la classification.

1.3.4 Discussion

Notre chaîne de reconnaissance de symboles repose sur l'utilisation de la relation $\{Présence/Absence\}$ entre objets et attributs, nous avons opté pour l'approche sac de mots conduisant à une représentation symbolique des caractéristiques extraites à partir des symboles afin d'éliminer l'étape de discrétisation des signatures utilisée dans d'autres travaux. Bien qu'il est difficile d'évaluer chaque étape du système proposé à part, nous avons choisi d'évaluer notre approche dans le cadre d'une chaîne de reconnaissance de symboles segmentés.

1.4 Résultats Expérimentaux

1.4.1 Protocole d'expérimentation

Cette section est dédiée à l'évaluation quantitative et qualitative de notre chaîne de traitement pour la reconnaissance de symboles graphiques. Afin de tester l'aptitude de notre classifieur à reconnaître des symboles requêtes, nous avons choisi de le comparer aux classifieurs usuels de l'état de l'art. La Figure.1.9 montre quelques échantillons de symboles utilisés en guise de requêtes dans notre système de reconnaissance. Dans le but d'obtenir une évaluation réaliste des performances de notre système de classification et afin de le positionner par rapport aux approches de la littérature, nous avons utilisé plusieurs classes de symboles.

Nous avons mené des expérimentations sur une base de symboles segmentés de la base

GREC'03² et de la base *shape*³. Nous distinguons six catégories de bruits résultant de cette base que nous appelons **GREC** :

- Modèles de symboles obtenus par des transformations linéaires : changement d'échelle (voir Figure.1.9(a)). Nous appelons cette base **GREC_{scale}**⁴.
- Modèles de symboles obtenus par des transformations linéaires : changement de rotation (voir Figure.1.9(b)). Nous appelons cette base **GREC_{rot}**⁵.
- Modèles obtenus par des transformations linéaires : changement d'échelle et de rotation (voir Figure.1.9(c)). Nous appelons cette base **GREC_{scale+rot}**⁶.
- Modèles dégradés (voir Figure.1.9(d)). Nous appelons cette base **GREC_{degr1}**⁷.
- Modèles ayant subi une forte dégradation (niveau 1) (voir Figure.1.9(e)). Nous appelons cette base **GREC_{degr1}**⁸.
- Modèles ayant subi une forte dégradation (niveau 2) (voir Figure.1.9(f)). Nous appelons cette base **GREC_{degr2}**⁹.
- Modèles ayant subi une distorsion (niveau 1) (voir Figure.1.9(g)). Nous appelons cette base **GREC_{dist1}**¹⁰.
- Base OC2 (shape 99)¹¹.

1.4.2 Mesures de performances

La littérature est très riche en travaux de reconnaissance de symboles visant à retrouver les objets similaires à une requête. Toutes les approches gravitent autour du problème de l'évaluation de l'étape de représentation des primitives des graphiques, ayant un impact direct sur les performances de la reconnaissance. La première raison de la prolifération de ce type d'évaluations est que cette étape est décisive pour le reste du processus de reconnaissance. La deuxième raison est liée au nombre important des détecteurs de points d'intérêt ainsi que leurs diversités et l'impossibilité de trouver de méthode universelle qui donne de bons résultats indépendamment de la distribution des données. Il sera donc difficile de se prononcer pour tel ou tel détecteur sans évaluer ses performances par rapport à l'état de l'art.

Considérés comme des systèmes de recherche d'information, l'évaluation des performances des systèmes de reconnaissance se base souvent sur des mesures classiques tels que la *précision*, le *rappel* et le *F-score*. L'analyse des réponses du système suite à une interrogation

2. <http://www.cvc.uab.es/grec2003/SymRecContest/index.htm>.

Cette base est composée de 50 modèles de symboles segmentés ou isolés.

3. <http://www.lems.brown.edu/vision/researchAreas/SIID/>

4. Cette base contient 250 symboles de la base **GREC** ayant subi des transformations d'échelle. Le facteur d'échelle varie entre 0.5 et 1.5. Le nombre d'occurrences de chaque modèle varie entre 0 et 13.

5. Cette base contient 250 symboles de la base **GREC** ayant subi des rotations avec un angle $\alpha \in [0, \Pi]$. Le nombre d'instances de chaque modèle de symbole varie entre 0 et 11.

6. Cette base contient 250 symboles de la base **GREC** ayant subi à la fois des transformations d'échelle (le facteur d'échelle varie entre 0.5 et 1.5). et des rotation ($\alpha \in [0, \Pi]$). Le nombre d'instances de chaque modèle varie entre 0 et 10.

7. Cette base contient 250 symboles de la base **GREC**. Le nombre d'occurrences de chaque modèle est 5 occurrences pour chaque modèle.

8. Cette base contient 75 symboles de la base **GREC** avec une forte dégradation et 20 modèles de **GREC**. Le nombre d'occurrences de chaque modèle varie entre 1 et 11.

9. Cette base contient 75 symboles de la base **GREC** avec une forte dégradation et 15 modèles de **GREC**. Le nombre d'occurrences de chaque modèle varie entre 1 et 11.

10. Cette base contient 75 symboles de la base **GREC** avec une distorsion et 15 modèles de **GREC**. Le nombre d'occurrences de chaque modèle varie entre 1 et 10.

11. <http://www.lems.brown.edu/vision/researchAreas/SIID/>

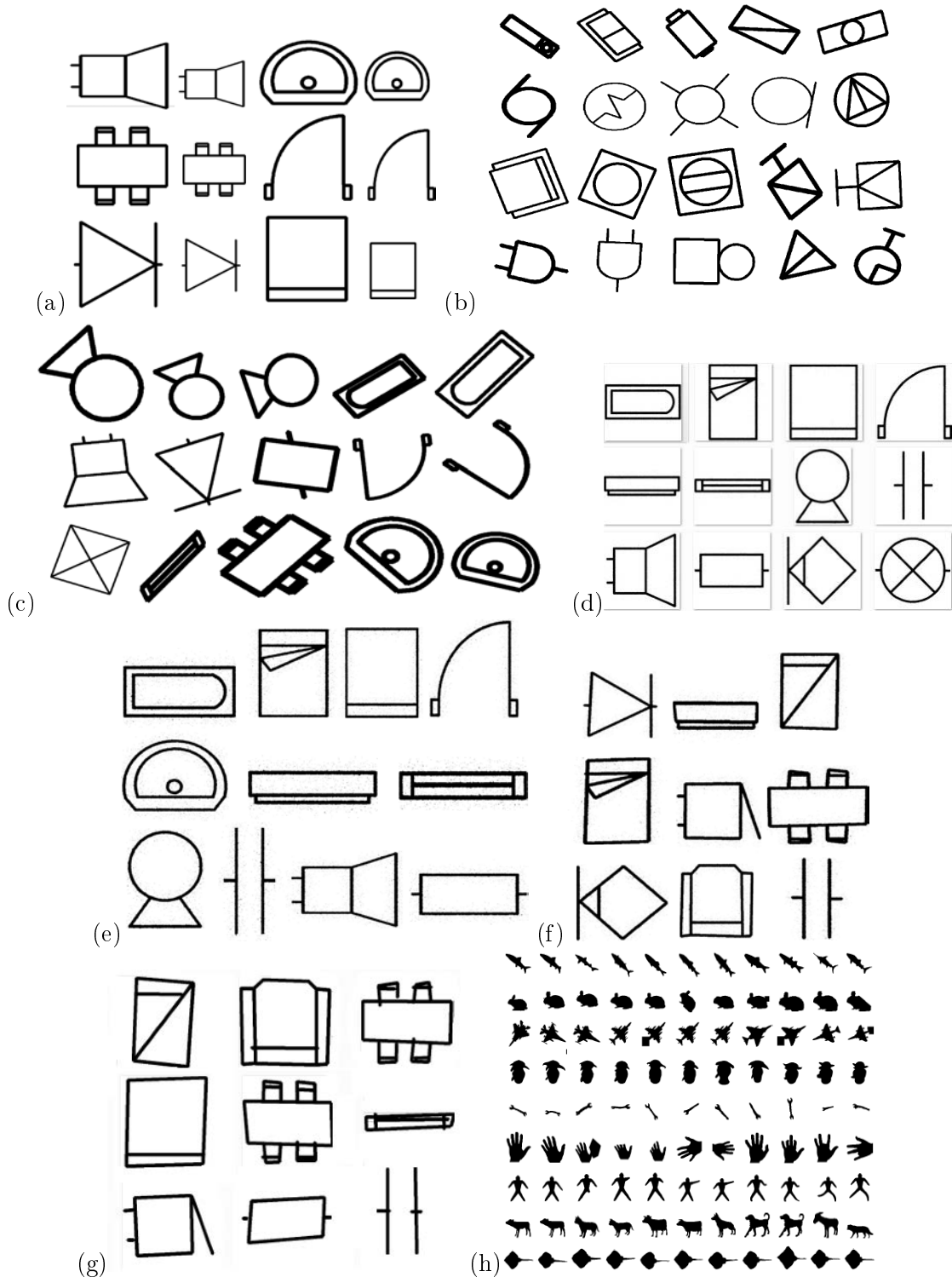


FIGURE 1.9 – Quelques exemples de symboles dans les bases de test (a) Symboles ayant subi des transformations d'échelle $GREC_{scale}$, (b) Symboles ayant subi des rotations $GREC_{rot}$, (c) Symboles ayant subi des transformations d'échelle et de rotation $GREC_{scale+rot}$, (d) Symboles dégradés niveau 1 de la base $GREC_{degr1}$, (e) Symboles dégradés niveau 2 de la base $GREC_{degr2}$, (f) Symboles détériorés niveau 1 de la base $GREC_{dist1}$, (g) Symboles détériorés niveau 2 de la base $GREC_{dist2}$, (h) Base OC2 (shape 99)¹³.

de la base de symboles permet de calculer ces valeurs. Les réponses dans un système de détection de symboles dans leur contexte sont de deux types : des réponses *pertinentes* et des réponses *non-pertinentes*.

Soit une base de données \mathcal{B} contenant un ensemble de symboles graphiques et une requête i , l'objectif d'un système de reconnaissance d'objets graphiques consiste à chercher les objets appartenant à \mathcal{B} et qui soient les plus vraisemblables au symbole requête, nous notons \mathcal{B}_p l'ensemble des éléments *pertinents* et \mathcal{B}_n l'ensemble des éléments *non-pertinents* par rapport à i .

$$\mathcal{B}_p, \mathcal{B}_n \rightarrow \mathcal{B} \quad (1.10)$$

Les critères d'évaluation du système de reconnaissance de symboles graphiques sont les suivantes :

- *Taux de reconnaissance* : il s'agit d'un critère d'évaluation commun entre les systèmes de reconnaissance. Il s'agit de la mesure de l'aptitude du système à reconnaître les occurrences les plus vraisemblables à une requête.
- *Précision* : elle mesure la capacité du système à fournir des réponses *pertinentes*.
- *Rappel* : il est défini par le rapport entre le nombre d'éléments *pertinents* trouvés et le nombre total d'éléments *pertinents* de la base. Il mesure la capacité du système à trouver exhaustivement les éléments *pertinents* existants.

1.4.3 Résultats et Discussions

L'expérience illustrée dans la Table.1.1 vise à évaluer les performances de notre système de reconnaissance en fonction de l'échantillon d'apprentissage. Cela consiste pour chaque type de bruit, à choisir au hasard n symboles pour l'ensemble d'apprentissage et le reste des symboles représentent l'ensemble de test (validation croisée).

Les taux de reconnaissance présentés dans la Table.1.1 correspondent à la moyenne des taux obtenus en validation croisée. Afin de se positionner par rapport aux travaux de l'état de l'art qui sont les plus proches à notre approche et vu l'absence des contributions dans ce contexte, nous avons choisi d'utiliser le même protocole d'expérimentation présenté dans [Bertet *et al.*2008]. Les auteurs ont choisi d'inverser l'ensemble de test et d'apprentissage en utilisant respectivement 5 blocs, 10 blocs et 26 blocs. Parmi les 5 blocs, 1 bloc sert pour l'apprentissage voire 182 symboles et les 4 autres pour le test voire 728 symboles. Les résultats obtenus pour 5 et 10 blocs montrent que notre approche basée sur l'approche par sac de mots fournit de meilleurs résultats même quand le nombre de symboles appris n'est pas trop important en comparaison avec [Bertet *et al.*2008]. Ceci peut montrer le pouvoir discriminant du vocabulaire utilisé à l'encontre des signatures discrétisées. À l'encontre de ces résultats, pour 26 blocs, le treillis combiné avec la représentation par sac de mots s'avère moins performant que l'approche basée sur la discrétisation des signatures, ce qui pourra s'expliquer par un phénomène de sur-apprentissage. La probabilité d'apparition de fausses détections est élevée. Cela peut être dû à l'insuffisance des mots visuels dans le cas de grands volumes de bases de symboles. La taille de l'ensemble d'apprentissage est décisive pour l'obtention de bonnes performances de classification. Le taux de classification augmente considérablement en présence d'un nombre satisfaisant de symboles de l'échantillon d'apprentissage ainsi qu'une description fine et discriminante des caractéristiques de ces derniers grâce au vocabulaire visuel utilisé. Un constat global d'après cette série d'expériences est que le taux de reconnaissance s'améliore en fonction de l'augmentation du nombre de symboles appris. Au dessus d'un certain nombre de symboles appris, le taux de reconnaissance diminue. En conclusion, nous pouvons dire que pour un nombre fixé

de classes, l'augmentation de la taille de l'ensemble d'apprentissage améliore la reconnaissance jusqu'à un certain niveau. Lorsque le nombre de classes de symboles atteint un seuil précis, nous remarquons que le taux de reconnaissance baisse. Une solution pour pallier à

Validation croisée	5 blocs	10 blocs	26 blocs
Notre approche	98.1	94.3	71.6
[Bertet <i>et al.</i> 2008]	97.4	92.2	82.6

TABLE 1.1 – Comparaison des taux de reconnaissance obtenus par le treillis de Galois de notre approche et ceux de [Bertet *et al.*2008]. Nous avons mené les expérimentations avec le même protocole d'expérimentation utilisé dans [Bertet *et al.*2008]. Pour cela, nous avons utilisé respectivement 5 blocs, 10 blocs et 26 blocs. Parmi les 5 blocs, 1 bloc sert pour l'apprentissage voire 182 symboles et les 4 autres pour le test voire 728 symboles. La taille du vocabulaire $|\Omega|$ est fixée à 80 mots visuels.

un tel problème serait d'utiliser une heuristique qui consiste à ne générer que les concepts nécessaires à la classification comme nous l'avons expliqué dans les sections précédentes et d'intégrer une étape d'élagage du treillis et de se focaliser sur des chemins prédominants issus de la phase d'apprentissage. Néanmoins, nous avons constaté que certains mots visuels peuvent nuire aux performances de reconnaissance, nous avons choisi alors de filtrer le vocabulaire visuel en termes des mots les plus discriminants et les plus informatifs ce qui s'est avéré une bonne solution. Cette approche sera détaillée dans le chapitre suivant. Toute la difficulté de ce travail réside dans la phase de préparation de la table binaire du treillis en essayant de migrer d'un espace d'histogrammes de fréquences d'apparition des mots à un ensemble de motifs symboliques qui seront cédés directement au treillis. Nous avons utilisé pour cela une structure de fichier xml où nous avons stocké les mots visuels et leurs fréquences d'apparition dans chaque modèle, ensuite nous avons extrait les relations $\{Absence/Présence\}$ entre chaque modèle et les mots qui le caractérisent. De cette façon, nous ne traitons plus de numérique, ni de fréquences mais plutôt du symbolique ce qui rendra le traitement et la manipulation du treillis très facile. Une amélioration considérable des résultats de reconnaissance est alors attendue par le recours aux données symboliques au lieu des données numériques ce qui est déjà approuvé et vérifié d'après les résultats obtenus.

Dans ce contexte et dans le but de présenter une évaluation réaliste et en vue de se rendre compte de l'originalité de l'idée proposée, nous avons choisi de nous comparer aux différents algorithmes de discrétisation utilisés dans les autres travaux (voir Table.1.2). Dans les approches classiques de reconnaissance utilisant le treillis de Galois, l'étape de discrétisation des signatures numériques et leur découpage en intervalles s'avère nécessaire. Il existe différentes techniques de discrétisation des valeurs continues et qui ont été utilisées dans [Guillas *et al.*2005] comme par exemple la technique de découpage récursif minimisant l'entropie ou encore l'algorithme de plus petit intervalle.

Nous avons choisi de comparer l'approche par sac de mots aux différentes techniques utilisées dans [Guillas *et al.*2005] afin de pouvoir trancher entre les deux approches. La comparaison des méthodes est effectuée en appliquant différents classifieurs (BFTree¹⁴, J48, réseau Bayésien et NBTrees¹⁵). Les résultats sont présentés dans la Table.1.2. Le treillis de Galois donne de meilleurs résultats (97,08%), si on l'associe à une approche symbolique qui

14. Best First DecisionTree

15. L'algorithme NBTree utilise un arbre de décision avec un classifieur Bayésien naïf dans les feuilles

n'a pas besoin d'un algorithme de discrétisation, tandis que la discrétisation semble être la forte raison de dégradation des résultats du classifieur vu que le taux de reconnaissance retourné par la méthode de Plus Petit Intervalle (PPI) atteint 91,02%, et la technique de la distance d'Entropie combiné au treillis de Galois fournit un taux de 90%.

Nous constatons aussi qu'avec les signatures numériques discrétisées en utilisant la technique de la distance d'Entropie, le réseau Bayésien s'avère plus performant que le treillis de Galois, mais notre choix d'un tel classifieur tient à sa capacité et sa robustesse vu qu'il présente un espace de recherche exhaustif et concis. Cette expérimentation permet de trancher entre la discrétisation des signatures et l'utilisation de l'approche par sac de mots combinée avec le treillis de Galois. En guise de conclusion, il nous a paru très intéressant de favoriser l'utilisation d'un vocabulaire visuel avec le treillis de Galois au lieu des signatures numériques classiques qui peuvent induire une complexité algorithmique exponentielle (en termes de temps et d'espace) dans le pire des cas ainsi qu'une dégradation considérable des performances du classifieur.

Classifieur/Algorithme	Sac de mots	PPI	Entropie
Réseau Bayésien	97.14%	84.29%	98.57%
BFTree	95.83%	90%	88.35%
J48	92.86%	84.29%	95.71%
NBTrees	92.8%	84.2%	89.57%
Treillis de Galois	97.08%	91.02%	90%

TABLE 1.2 – Comparaison entre différents algorithmes de discrétisation des signatures numériques et l'approche par sac de mots. Les signatures utilisées dans ces expérimentations sont les moments de Zernike. Dans cette expérimentation, nous avons utilisé 9 modèles de symboles, chaque modèle contient 50 symboles. L'ensemble d'apprentissage contient 325 symboles et l'ensemble de test contient 75 symboles. La taille du vocabulaire est $|\Omega|=80$.

1.4.4 Évaluation de la taille du treillis de concepts en fonction du nombre d'attributs

Dans ce qui suit, nous avons comparé les taux de reconnaissance et les tailles des structures générées par notre approche et l'approche proposée dans [Guillas *et al.*2005]. Comme nous l'avons précisé, la taille du treillis est contraignante pour le système de reconnaissance de symboles. Cette problématique peut être appréhendée par la mise en place d'heuristiques d'élagage du treillis en éliminant les concepts vides et non pertinents.

Dans notre approche et pour une taille moindre, nous avons proposé d'utiliser une variante de l'algorithme de Norris permettant de générer uniquement les concepts nécessaires à la classification. Cette piste nous a permis de faire face au problème de la complexité exponentielle du treillis de Galois dans le pire des cas. Quelques résultats sont représentés dans la Table.1.3, permettant d'évaluer la taille du treillis en ayant recours à l'algorithme de Norris permettant de générer les concepts nécessaires à la classification.

	Nb _{concepts}	Temps _{appr.} (sec)	Temps _{classif.} (sec)
Trellis entier [Guillas et al.2005]	3185	430.2	2
Génération à la demande [Guillas et al.2005]	282	0.5	9.8
Notre approche : Treillis entier	2270	253.58	26.08
Notre approche : Génération des concepts nécessaires	202	1.7	4.7

TABLE 1.3 – Comparaison des différentes approches à base du treillis de concepts en termes de Nb_{concepts} : nombre de concepts dans le treillis généré, Temps_{appr} : temps nécessaire pour la construction du treillis de Galois et Temps_{classif} : le temps nécessaire pour la classification d'un nouvel objet avec le treillis de concepts. Dans cette expérimentation, nous avons utilisé 9 modèles de symboles, chaque modèle contient 50 symboles. L'ensemble d'apprentissage contient 325 symboles et l'ensemble de test contient 75 symboles. La taille du vocabulaire est $|\Omega|=80$.

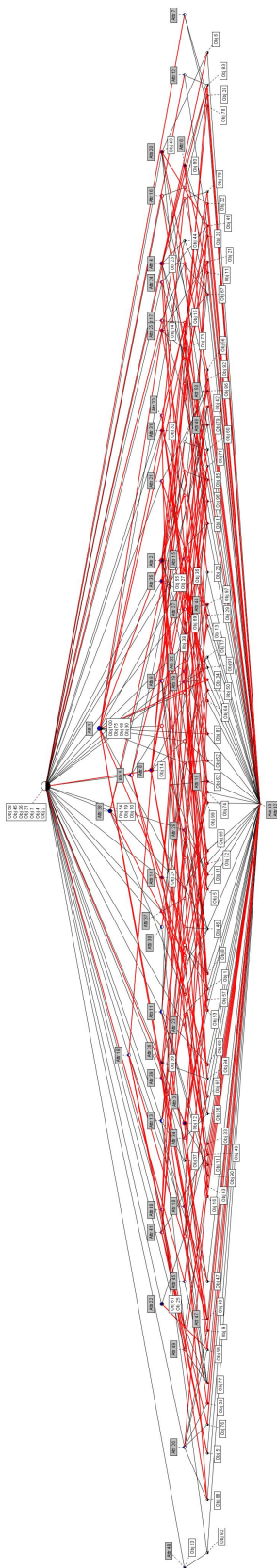


FIGURE 1.10 – La représentation graphique d'un treillis de Galois pour un sous-ensemble de 100 symboles et 60 attributs.

Les résultats obtenus dans la Table.1.3 montrent que la taille de notre treillis reste raisonnable dans la plupart des cas et dépend de la taille de l'ensemble d'apprentissage et de l'ensemble de test. En employant notre approche, nous remarquons que la taille du treillis est considérablement réduite en comparaison au treillis généré dans [Guillas *et al.*2005] ce qui se justifie par la robustesse de l'algorithme utilisé pour la construction du treillis ainsi que l'heuristique adoptée permettant la génération des concepts nécessaires à la classification. Le temps d'apprentissage est très rapide quand les concepts nécessaires sont les seuls à générer. Le temps de classification est parfois élevé, cela pourra s'expliquer par les modifications que peut subir le treillis lors d'une interrogation d'un symbole requête.

Cette évaluation montre que l'algorithme de construction utilisé a un impact direct sur la taille du treillis et le nombre de ses concepts qui s'est avéré beaucoup plus réduit par rapport aux autres approches. En conclusion, nous pouvons dire que si les symboles traités ne sont pas trop diversifiés il y a une forte probabilité d'obtenir de très hautes performances de classification en un temps minimal. Le treillis présente un espace de recherche concis et robuste offrant la possibilité de garder l'aspect sémantique à travers les relations $\{Absence/Présence\}$ entre les symboles et leurs attributs.

Dans la Figure 1.11, nous présentons un scénario possible de classification d'un nouvel objet dans le treillis de concepts. Remarquons que chaque objet et chaque attribut apparaît une seule fois dans le graphe grâce au treillis d'héritage qui, comme nous l'avons largement développé, permet de réduire la taille du treillis ainsi que l'information contenue dans ses concepts. Cette approche permet de répondre aux objectifs fixés ci-avant et plus précisément offre un bon compromis entre les performances de reconnaissance et la complexité algorithmique du classifieur.

L'idée de notre approche consiste à reconnaître la classe d'un symbole requête dont les propriétés sont les mots visuels du vocabulaire dans leur forme symbolique. La première étape consiste à insérer le nouvel objet comme source de donnée fictive. Une modification du treillis est alors prévue suite à cet ajout, soit par un ajout d'un nouveau nœud (si aucun nœud du treillis ne partage l'*intension* de la requête), soit par modification des nœuds existants dans le cas contraire. Suite à cet ajout, une recherche à partir du nœud inséré est établie en partant vers le haut de l'arborescence et en explorant les *intensions* des concepts et retenant les concepts partageant les attributs de la requête. Notons qu'il existe un seul concept qui contient exactement les attributs de la requête. Il s'agit d'une propriété du treillis de Galois. Nous allons détailler toutes ces étapes avec l'exemple suivant :

Soit un symbole requête défini par l'ensemble de ses attributs $\{Attr1, Attr4, Attr8\}$. La partie *extension* de ce nœud n'est plus dans le treillis présenté dans la Figure.1.11(a), il y aura alors un ajout d'un nouveau nœud (appelé *new*) dans le treillis. L'ajout de ce nœud aboutit au treillis présenté dans la Figure.1.11(b). L'étape de la reconnaissance consiste à chercher les concepts dont la partie *intension* contient des attributs communs avec la requête en partant vers le *top* à partir du nœud inséré. Dans notre cas, la Figure.1.11. (c) montre les concepts retenus comme candidats en partant de bas vers le haut selon l'ordre décroissant de leur pertinence : l'algorithme de recherche aboutit premièrement à l'objet *obj6* qui possède un attribut commun avec le symbole requête à savoir l'attribut *Attr1*, ensuite passons au deuxième niveau contenant les deux objets *obj9* qui partagent les attributs $\{Attr1, Attr4\}$ avec la requête et *obj5* qui partagent exactement le même nombre d'attributs $\{Attr1, Attr4, Attr8\}$ avec la requête. Vu que le treillis est complet alors ce nœud est unique donc la classe du symbole requête est la même classe de l'objet *obj5*. Ce processus est rapide et s'est avéré très efficace dans le contexte de la classification de symboles. Cette stratégie est prometteuse car elle est plus simple et efficace en comparaison à

celle proposée dans [Guillas *et al.*2005] dont le principe est de parcourir le treillis afin de sélectionner un ensemble d'intervalles à partir du concept courant. Une mesure de distance entre la valeur v_i de la requête et un intervalle x_j est nécessaire. Cette méthode requiert des calculs excessifs et engendre des incertitudes de reconnaissance, à l'encontre de notre stratégie de classification qui permet de comparer des attributs sans aucune opération de calcul.

1.4.5 Comparaison avec quelques classifieurs usuels

Nous avons souhaité comparer notre approche avec les classifieurs de l'état de l'art combinés à de nombreux descripteurs. Les symboles sont décrits par des mots visuels résultant de l'étape de quantification des descripteurs en appliquant l'algorithme de *k-means*. Les descripteurs utilisés sont SURF, SIFT et ceux utilisés dans [Guillas *et al.*2005] (signature de Radon, les moments de Zernike et le descripteur de Fourier) (voir Table.1.4 et Figure.1.12). Nous présentons les taux de classification obtenus par le treillis de Galois ainsi que par quelques classifieurs de l'état de l'art.

Le descripteur SURF fournit de meilleurs résultats avec le treillis de Galois. De plus, l'approche symbolique fournit de meilleurs résultats par rapport à l'approche classique qui utilise une signature numérique (dans ce cas c'est la signature de Radon). Dans ce cadre, nous avons choisi d'utiliser une taille du vocabulaire qui permet de représenter les objets et afin de ne pas rendre fastidieux l'usage du treillis. Pour que le treillis de Galois aboutisse à une bonne reconnaissance des symboles, il faut le combiner avec des attributs ayant un pouvoir discriminant assez important.

Descripteur/TR	Treillis de Galois	6-ppv	Arbre de Décision
SIFT	79.7%	85.8%	70.2%
BoF+SURF	94.4%	82.9%	87.1%
Signature de Radon	88.1%	87.6%	75.9%
Descripteur de Fourier	87%	75.8%	62%
Moments de Zernike	59.3%	60.4%	46%

TABLE 1.4 – Comparaison des Taux de Reconnaissance (TR) de quelques classifieurs combinés à différentes signatures. Dans cette expérimentation, nous avons utilisé 9 modèles de symboles, chaque modèle contient 50 symboles. Le nombre de symboles d'apprentissage utilisés dans cette expérimentation est 302 symboles bruités et 98 symboles pour l'échantillon du test. La taille du vocabulaire est $|\Omega|=80$.

1.4.6 Évaluation de l'étape de représentation de symboles

La Figure.1.13 évalue la robustesse du treillis de Galois combiné d'une part, à des signatures numériques comme les moments de Zernike, le descripteur de Fourier, SIFT, GFD et la signature de Radon et d'autre part, avec la représentation par sac de mots. Les résultats obtenus montrent que l'approche sac de mots fournit les meilleurs résultats, et le taux de rappel est de l'ordre de 80% dans la plupart des cas. L'approche par sac de mots que nous avons utilisée afin de faire face à la complexité algorithmique, permet de grouper les symboles en classes en maximisant la similarité intra-classes et minimisant celle inter-classes.

L'association entre l'approche par sac de mots pour l'extraction des primitives visuelles des

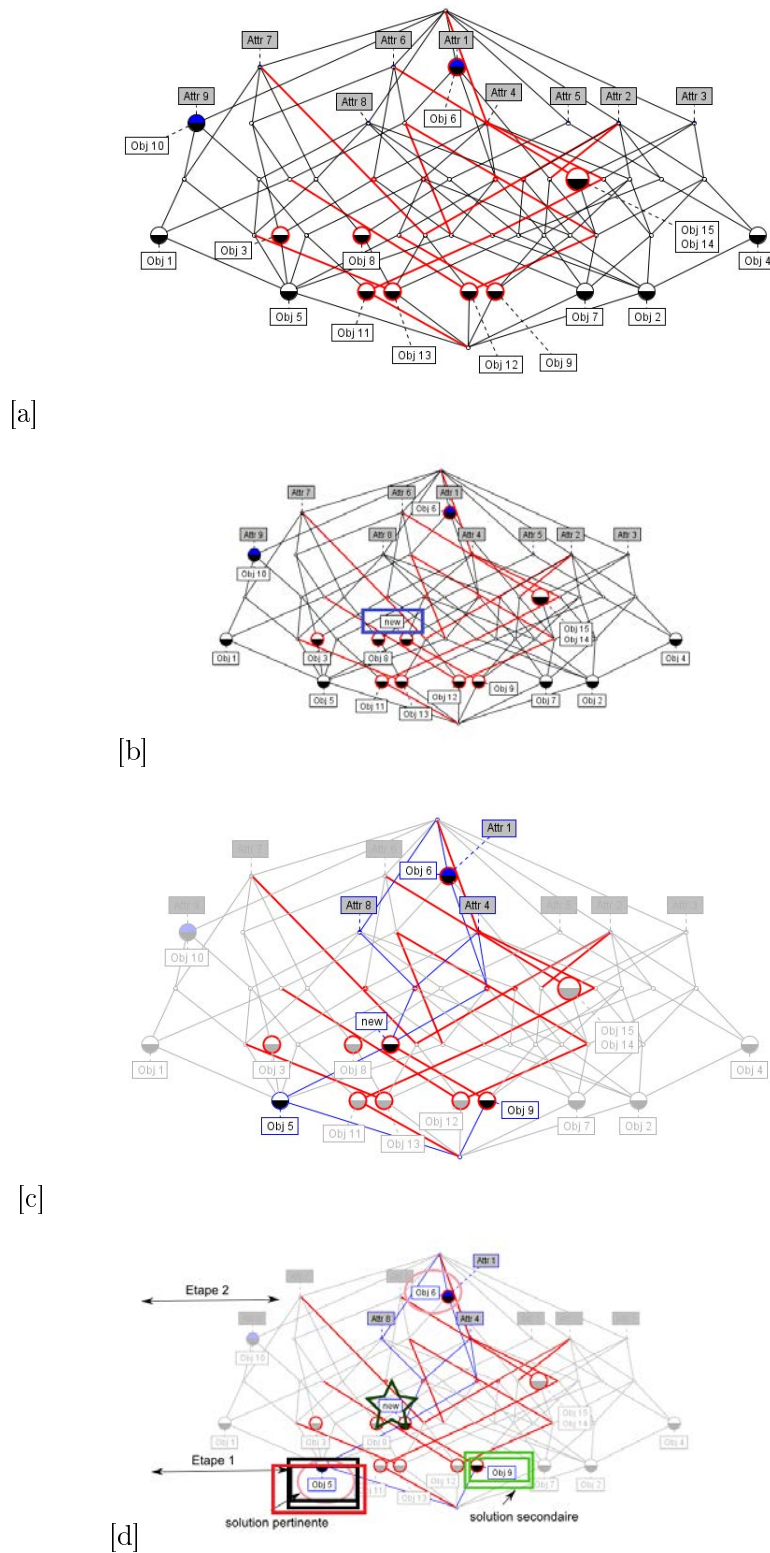


FIGURE 1.11 – Exemple illustratif d’un processus de classification d’un symbole requête. (a) Le treillis de Galois représentant les relations entre 15 symboles et 9 attributs. (a) et (b) Mise à jour du treillis de concepts suite à l’introduction de l’objet requête, (c) Navigation dans l’arborescence de bas vers le haut en vue de la détermination des concepts dont la partie *intension* correspond à la partie *intension* de l’objet requête, (d) Les concepts retenus comme candidats résultant de la phase de la classification.

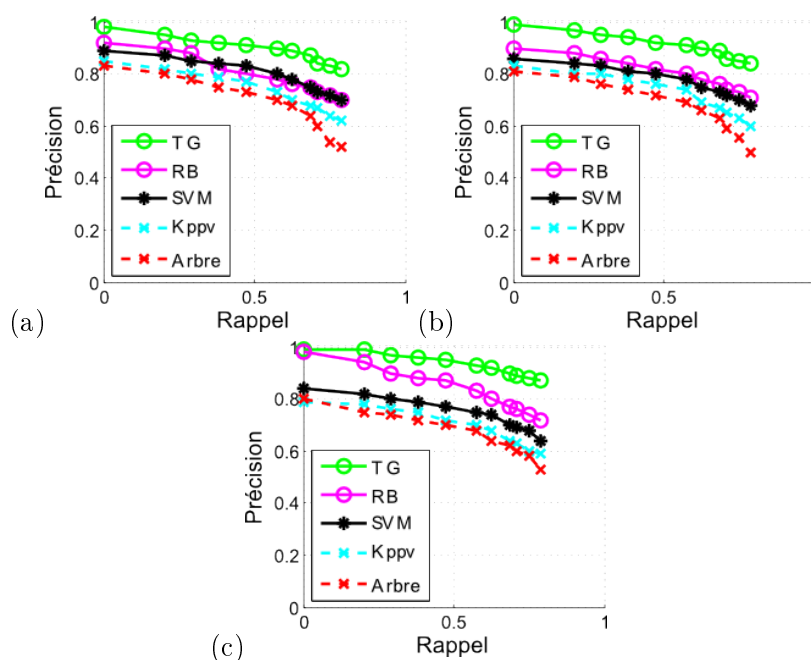


FIGURE 1.12 – Comparaison des taux de *Précision/Rappel* du treillis de Galois et les classifieurs usuels de l'état de l'Art. Dans cette expérimentation, nous avons utilisé 9 modèles de symboles, chaque modèle contient 50 symboles. Le nombre de symboles d'apprentissage utilisés dans cette expérimentation est 302 symboles bruités et 98 symboles pour l'échantillon du test. Dans (a) la taille du vocabulaire est $|\Omega|=30$, (b) $|\Omega|=50$ et (c) $|\Omega|=80$.

symboles graphiques avec le treillis de Galois fournit un taux de reconnaissance assez élevé sachant qu'elle permet aussi d'éliminer l'étape de discrétisation des signatures numériques utilisées dans les approches classiques.

Impact de la combinaison et de la réduction des caractéristiques

Afin de bien mener l'évaluation de notre système de classification de symboles basé sur le treillis de concepts, nous avons choisi dans cette série d'expérimentations d'évaluer les descripteurs utilisés en introduisant une étape de sélection des caractéristiques basée sur des techniques existantes dans la littérature. Nous avons évalué l'impact de cette étape sur les performances du classifieur. Dans une première série d'expérimentations et en vue d'évaluer l'étape de représentation des objets graphiques et son impact sur les taux du classification, nous avons opté pour une combinaison de différents descripteurs afin de décrire un même objet. La Figure.1.14 montre l'impact de l'étape de combinaison de quelques descripteurs à savoir la R-signature (désignée par R), les Moments de Zernike (Z) et le GFD (G). Pour évaluer la qualité des résultats de classification obtenus par notre classifieur, nous avons testé l'impact de la taille de l'échantillon d'apprentissage sur le taux de classification. Nous constatons que la combinaison des descripteurs améliore dans la plupart des cas les taux de classification obtenus par le classifieur et en augmentant à chaque fois le nombre des symboles lors de l'étape d'apprentissage.

Dans une seconde série d'expérimentations, nous avons appliqué la réduction des caractéristiques des descripteurs cités ci-dessus. Le recours à la sélection des caractéristiques permet la réduction de la dimensionnalité des descripteurs qui parfois peut affecter les

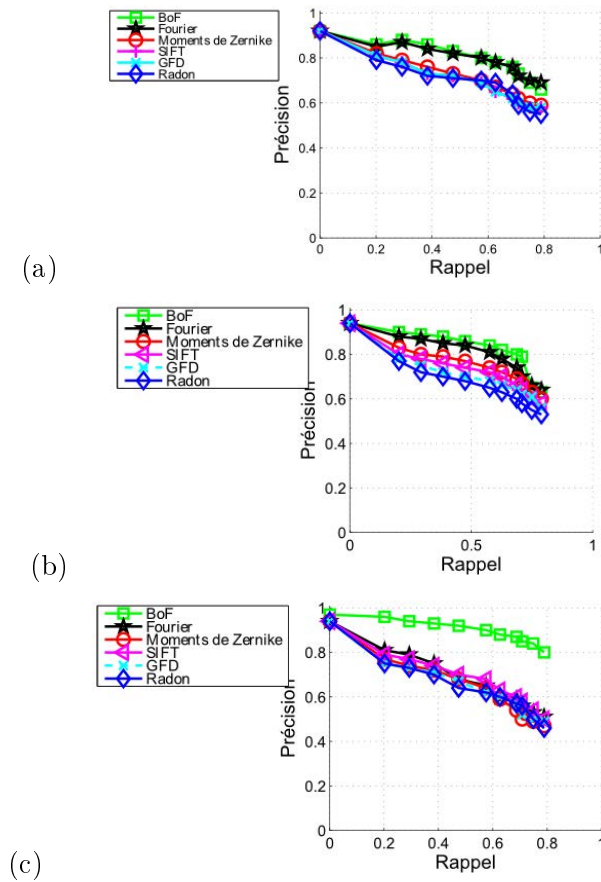


FIGURE 1.13 – Comparaison des taux de *Précision/Rappel* du treillis de Galois+BoF et du treillis de Galois+signatures numériques. Dans cette expérimentation, nous avons utilisé 9 modèles de symboles, chaque modèle contient 50 symboles. Le nombre de symboles d'apprentissage utilisés dans cette expérimentation est 302 symboles bruités et 98 symboles pour l'échantillon du test. (a) $|\Omega| = 30$, (b) $|\Omega| = 50$ et (c) $|\Omega| = 80$.

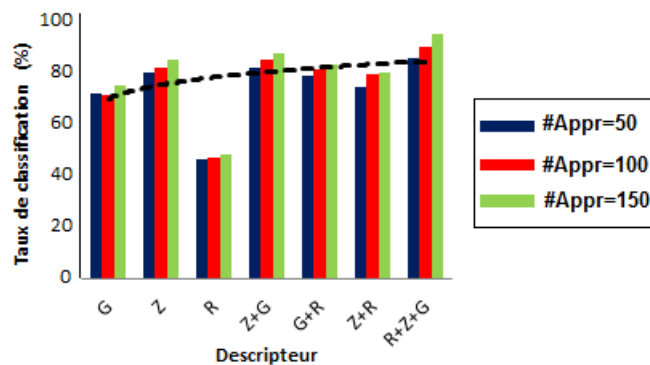


FIGURE 1.14 – Évaluation du taux de classification du treillis de Galois en fonction de la taille de l'échantillon d'apprentissage en utilisant la combinaison des descripteurs suivants : Moments de Zernike (**Z**), GFD (**G**) et la *R*-signature (**R**).

performances du classifieur. Généralement, il existe deux types de méthodes de réduction de dimensionnalité :

- La réduction basée sur la sélection des caractéristiques les plus pertinentes parmi un ensemble de données avec souvent un nombre beaucoup plus faible d'individus. Dans le cadre de cette évaluation, nous nous intéressons particulièrement à la méthode LASSO permettant de réduire la taille des matrices des descripteurs diminuant ainsi considérablement les temps de calculs.
- La réduction basée sur la transformation des caractéristiques initiales en un nouvel ensemble plus réduit (voir Figure.1.15 qui illustre la méthode ACP pour la réduction des caractéristiques).

Dans cette évaluation, nous nous sommes focalisés sur le premier type de méthodes par l'utilisation de la technique LASSO de réduction des caractéristiques et plus précisément la réduction des nombres des composantes des descripteurs extraits à partir de symboles graphiques. Le problème de la sélection des caractéristiques est défini généralement comme suit :

Soit $\mathcal{I} = \{i_1, i_2, \dots, i_n\}$ un ensemble de caractéristiques de taille n où n est le nombre total de caractéristiques. Soit \mathcal{F} une fonction qui permet d'évaluer un sous-ensemble de caractéristiques. La sélection des caractéristiques a donc pour objectif l'obtention de la valeur maximale de \mathcal{F} avec un sous-ensemble \mathcal{I}' ($\mathcal{I}' \subseteq \mathcal{I}$) plus réduit de caractéristiques de taille n' ($n' < n$) tel que :

$$\mathcal{F}(\mathcal{I}') = \max_{\mathcal{K} \subseteq \mathcal{I}} (\mathcal{F}(\mathcal{K})) \quad (1.11)$$

Où $|\mathcal{F}|=n'$ et n' est un nombre prédéfini par l'utilisateur ou soit contrôlé par les méthodes de génération de sous-ensembles. Dans notre cas, nous avons choisi de concaténer des vecteurs caractéristiques afin d'évaluer l'impact de la combinaison des descripteurs sur les résultats de classification ce qui nous place face au problème de dimensionnalité des composantes des vecteurs utilisés. Ce grand nombre de caractéristiques peut causer un apprentissage incorrect d'où nous avons eu recours à la méthode LASSO de réduction de dimensionnalité. Les caractéristiques les plus pertinentes sont alors extraites. LASSO [Tibshirani1996] est une méthode de sélection de variables et de diminution de coefficients pour la régression linéaire. Cette méthode se base sur une minimisation de la somme des erreurs quadratiques en utilisant un seuil correspondant à la somme des valeurs absolues des coefficients :

$$\beta^{lasso} = \operatorname{argmin}_{\beta} \sum_{i=1}^N (y_{i\beta_0} - \sum_{j=1}^p x_{ij}\beta_j)^2 \quad (1.12)$$

La contrainte imposée sur la pénalité $\sum_{j=1}^p \|\beta_j\|$ est la suivante : $\sum_{j=1}^p \|\beta_j\| \leq s$. Certains coefficients β_j seront nuls pour des petites valeurs de s ($s \geq 0$). Ces coefficients nuls correspondent aux variables qui seront éliminées de la base. Pour utiliser la méthode LASSO, nous avons utilisé l'algorithme de Least Angle Regression (LAR) [Efron *et al.*2004]. La Table 1.5 montre les résultats obtenus lors de l'étape de classification des symboles de la base *shape* en utilisant des descripteurs *avec* ou *sans* étape de sélection des caractéristiques. Durant ces expérimentations et dans le cas de l'usage des techniques de sélection de caractéristiques, une fois que les composantes les plus pertinentes des descripteurs sont sélectionnées, nous procédons au calcul du vocabulaire visuel qui sera cédé au treillis de concepts comme nous l'avons décrit tout au long de ce chapitre. Le classifieur est ensuite invité à reconnaître la requête. Notre constat est que les taux de classification *avant* et *après* la sélection sont relativement proches. Donc, la réduction des caractéristiques des descripteurs permet de réduire l'espace de caractéristiques tout en garantissant de bons

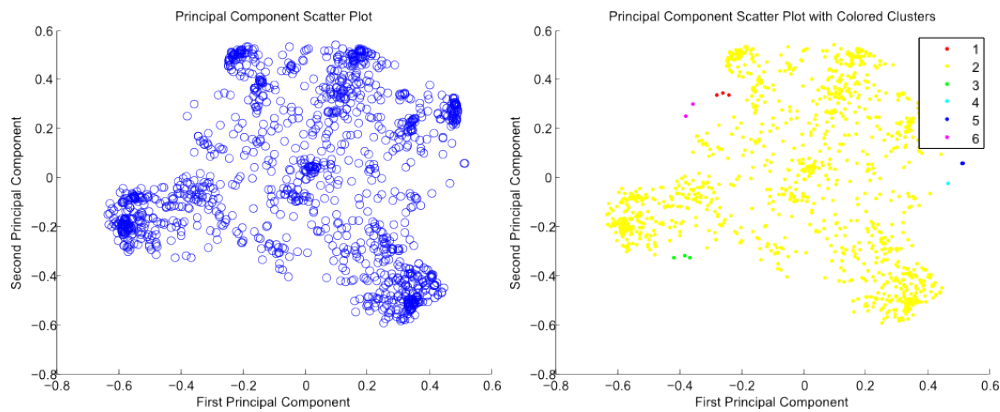


FIGURE 1.15 – Sélection des caractéristiques du descripteur SIFT extrait d’un symbole graphique en se basant sur la méthode d’ACP. À gauche, nous avons l’ensemble des vecteurs SIFTs représentant un symbole graphique. À droite, l’ensemble des variables sélectionnées par la méthode ACP représentant les vecteurs SIFTs les plus discriminants : il s’agit de 5 groupes de variables principales.

taux de classification.

La Table 1.6 montre le taux de classification pour chaque descripteur et pour quelques modèles de symboles *avant* et *après* la sélection. Nous remarquons que la sélection des caractéristiques des descripteurs permet dans la plupart des cas d’améliorer les taux de classification. Les résultats de classification obtenus avec la combinaison des descripteurs sont égaux ou parfois meilleurs que ceux obtenus par les descripteurs utilisés séparément. De plus, la sélection des variables ne dégrade pas les taux de classification mais l’améliore dans la plupart des cas.

La Table 1.7 montre les résultats de classification de quelques classificateurs : treillis de concepts, *k-ppv* et le classifieur de Bayes combinés à des descripteurs *avec* ou *sans* étape de sélection des caractéristiques. Les résultats montrent que le treillis de concepts associé à un ensemble réduit de caractéristiques offre de très bonnes performances de classification. Le classifieur de Bayes s’avère très compétitif à notre système et le dépasse dans certains cas en terme de classification. La méthode *k-ppv* combiné à une étape de sélection des caractéristiques des descripteurs offre de bons taux de classification en comparaison à ceux obtenus par le treillis de concepts ou encore par le classifieur de Bayes. L’inconvénient majeur de la méthode de *k-ppv* est l’obligation du stockage de l’ensemble de l’échantillon d’apprentissage. En outre, le classifieur de Bayes requiert une hypothèse (uniforme, normale,...) sur la distribution des données. Ces deux problèmes sont évités en ayant recours au treillis de concepts ce qui explique les bonnes performances fournies par ce classifieur. En conclusion, la sélection de caractéristiques permet d’améliorer la tâche d’extraction de caractéristiques, augmenter la précision des taux de classification ainsi qu’améliorer la fiabilité de l’estimation de la performance.

Descripteur	Sans sélection		Avec sélection	
	Nbcarac	Taux de classification (%)	Nbcarac	Taux de classification (%)
GFD_8x12	96	91.78%	38	92.05%
GFD_10x15	150	94.07%	58	97.32%
Moments de Zernike	66	93.4%	29	91.89%
R-signature	180	77.86%	42	80.45%
R-signature+GFD_8x12	276	75.6%	80	83.7%
R-signature+GFD_10x15	330	81.42%	132	85.56%
R-signature+Moments de Zernike	246	78.92%	71	79.87%
Moments de Zernike+GFD_8x12	162	95.7%	67	97.03%
Moments de Zernike+GFD_10x15	216	94.83%	94	96.03%

TABLE 1.5 – Résultats de l'étape de classification du treillis de concepts *avec/sans* sélection des caractéristiques à partir des descripteurs (cas de plusieurs descripteurs). Dans cette expérimentation, nous avons utilisé 5 modèles de symboles, chaque modèle contient 30 symboles. Le nombre de symboles d'apprentissage utilisés dans cette expérimentation est 102 symboles bruités et 48 symboles pour l'échantillon du test. La taille du vocabulaire est $|\Omega|=50$.

	Sans sélection			Avec sélection		
	Modèle 1	Modèle 2	Modèle 3	Modèle 1	Modèle 2	Modèle 3
GFD_8x12	70.32%	69.89%	82.32%	71.34%	78.45%	84.98%
Moments de Zernike	73.8%	84.5%	91.7%	75.87%	84.79%	94.7%
R-signature	53.45%	64.71%	71.56%	55.67%	67.08%	76.8%
R-signature+GFD_8x12	63.54%	72.76%	75.32%	66.24%	73.41%	72.97%
R-signature+Moments de Zernike	67.32%	74.12%	78.54%	65.34%	75.67%	79.67%
Moments de Zernike+GFD_8x12	76.89%	78.87%	92.34%	78.13%	82.06%	95.57%
Moments de Zernike+GFD_8x12+R-signature	86.82%	87.23%	95.56%	88.45%	90.56%	95.89%

TABLE 1.6 – Comparaison des taux de classification du treillis de concepts de quelques modèles de symboles de la base *shape* (cas de différentes signatures *avec/sans* sélection de caractéristiques). Dans cette expérimentation, nous avons utilisé 7 modèles de symboles, chaque modèle contient 30 symboles. Le nombre de symboles d'apprentissage utilisés dans cette expérimentation est 170 symboles bruités et 40 symboles pour l'échantillon du test. La taille du vocabulaire est $|\Omega|=50$.

Descripteur	TG		k -ppv		CB	
	Sans sélection	Avec sélection	Sans sélection	Avec sélection	Sans sélection	Avec sélection
Z	93.6%	94.02%	86.4%	89.45%	90.67%	92.67%
GFD_8x12	87.4%	88.56%	74.6%	72.67%	89.02%	93.65%
R	56.89%	65.78%	45.76%	48.34%	47.89%	54.76%
Z+GFD8x12	87.87%	89.98%	67.45%	68.98%	83.98%	85.76%
Z+R-signature	67.45%	70.56%	56.67%	59.56%	62.78%	64.34%
GFD_8x12+R	47.04%	49.18%	42.56%	47.87	47.04%	53.78%
GFD_8x12+R+Z	87.67%	89.78%	68.56%	70.1%	77.78%	79.04%

TABLE 1.7 – Comparaison des Taux de Reconnaissance (TR) de quelques classifieurs : Treillis de Galois (TG), k -ppv et le classifieur Bayésien (CB) combinés à différentes signatures (Z : Moments de Zernike, GFD et R : R-signature) *avec/sans* sélection de caractéristiques. Dans cette expérimentation, nous avons utilisé 8 modèles de symboles, chaque modèle contient 50 symboles. Le nombre de symboles d'apprentissage utilisés dans cette expérimentation est 350 symboles bruités et 50 symboles pour l'échantillon du test. La taille du vocabulaire est $|\Omega|=50$.

1.4.7 Comportement de la taille du treillis de concepts en fonction de la taille du vocabulaire

Le nombre de mots visuels a un impact direct sur le taux de reconnaissance retourné par le classifieur. Nous constatons que la performance du treillis augmente significativement avec l'augmentation du nombre d'attributs. Nous constatons qu'au bout d'un certain nombre d'attributs, le treillis stagne et fournit des performances semblables aux résultats avec un nombre plus réduit de mots visuels (prenons la taille 200 et 150 dans la Figure.1.16) ce qui nous emmène à supposer que le pouvoir discriminant du vocabulaire réside dans une partie de ses attributs : il peut contenir des attributs pertinents et d'autres non pertinents et redondants. Cela peut expliquer pourquoi la taille 200 aboutit aux mêmes résultats que la taille 150. Ce constat nous a inspiré par la suite et nous a incité à mieux explorer le vocabulaire visuel utilisé. Ainsi un treillis de petite taille signifie qu'il y a peu d'objets isolés et l'approche par sac de mots donne des résultats plus satisfaisants avec une complexité réduite (voir Table.1.8). Nous constatons d'après les résultats fournis dans la Table.1.8 que la taille du treillis est raisonnable quand nous utilisons un nombre modéré de mots visuels. Le bon choix de l'algorithme de construction et les heuristiques adoptées et le vocabulaire visuel aboutissent à une réduction considérable de la complexité du classifieur et nous ont permis de surpasser le problème de la complexité exponentielle dans le pire des cas.

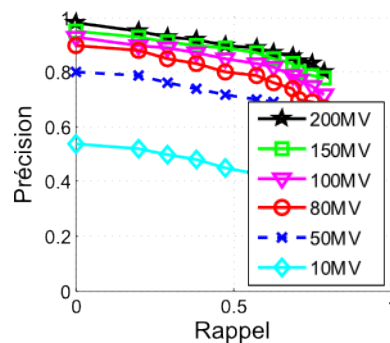


FIGURE 1.16 – Évaluation des performances du treillis de Galois en fonction de la taille du vocabulaire. Dans cette expérimentation, nous avons utilisé 5 modèles de symboles, chaque modèle contient 30 symboles. Le nombre de symboles d'apprentissage utilisés dans cette expérimentation est 102 symboles bruités et 48 symboles pour l'échantillon du test. La taille du vocabulaire est $|\Omega|=50$.

$ \Omega $	10	20	50	70	100	120	130
Nombre de concepts	11	32	43	67	120	265	282

TABLE 1.8 – Évolution de la taille du treillis en fonction du nombre de mots visuels ($|\Omega|$).

1.5 Conclusion

Dans ce chapitre, nous avons exposé notre chaîne de traitement en rapport avec la reconnaissance de symboles. L'originalité de l'approche proposée réside dans la combinaison d'une représentation par sac de mots avec le treillis de Galois. Les résultats présentés dans

ce travail ont permis une comparaison entre l'utilisation de la signature discrétisée pour la construction du treillis de concepts et l'approche symbolique fondée sur les mots visuels. Les contributions présentées dans ce chapitre sont diverses. Notre première contribution réside dans l'exploitation de l'approche sac de mots au lieu des signatures numériques discrétisées combinée au treillis de Galois. Nos choix ont été motivés par l'aspect symbolique offert par le sac de mots démontrant son intérêt pour augmenter la robustesse du treillis de concepts utilisé en tant que classifieur intégré dans notre chaîne de traitement de reconnaissance. Outre cette contribution, la charge de calcul a été considérablement réduite en comparaison avec les anciennes approches qui se base sur le calcul des signatures numériques et leur discrétisation en intervalles figés.

Un constat général que nous pouvons faire est que le treillis de Galois offre des perspectives intéressantes pour la reconnaissance d'objets quand il est combiné à une représentation par sac de mots. La méthode de représentation des caractéristiques des symboles graphiques est décisive pour l'obtention de très bonnes performances du classifieur, puisque la construction du treillis est obtenue à partir de la table binarisée décrivant la relation entre *objets-attributs* et ne dépend d'aucun paramètre. Dans ce cadre et en vue de l'amélioration de cette approche, nous allons présenter une amélioration dans le prochain chapitre qui sera inspirée de celle introduite dans ce chapitre, c'est ce que nous appelons le *Treillis de Galois Fréquent* basé non pas sur tout le dictionnaire visuel mais sur une partie comportant les mots visuels ayant le pouvoir discriminant le plus élevé et qui garantissent des performances de classification meilleures. Une perspective de ce travail est d'améliorer le pouvoir modélisant de la représentation par sac de mots que nous avons adoptée afin d'obtenir d'excellentes performances de reconnaissance. Pour pallier aux phénomènes d'explosion de la taille du treillis de Galois quand il est utilisé avec de grandes bases de données, le *Treillis de Galois Fréquent* s'avère une solution.

Treillis de Galois Fréquent pour la reconnaissance de symboles

"Quand un rêve se répète, c'est qu'il cherche son interprète".

ANNA LIETTI

Sommaire

2.1	Introduction	97
2.2	Approche proposée	98
2.3	Résultats Expérimentaux	105
2.3.1	Comparaison du TGF avec les autres approches	106
2.3.2	Invariance aux transformations : Bases $GREC_{scale}$, $GREC_{rot}$ et $GREC_{scale+rot}$	112
2.3.3	Invariance aux déformations ou occlusions : Bases $GREC_{degr1}$, $GREC_{degr2}$ et $GREC_{dist1}$	112
2.3.4	Évaluation des techniques de sélection de caractéristiques	114
2.3.5	Évaluation du taux de reconnaissance du treillis de Galois en fonction de la taille du dictionnaire	114
2.4	Conclusion	115

2.1 Introduction

Dans le chapitre précédent, nous avons présenté notre première contribution dans le cadre de la conception d'un système de reconnaissance de symboles graphiques. L'approche proposée a mis en exergue la nécessité de combiner une approche symbolique avec le formalisme du treillis de concepts. Cette méthode a donné de bonnes performances dans le contexte de la reconnaissance de symboles. Une question découlait après l'évaluation de la chaîne de traitement proposée qui nous a parue intéressante : *Si le vocabulaire visuel combiné avec la structure du treillis de Galois a montré des résultats très satisfaisants, pourquoi ne pas renforcer le pouvoir discriminant et modélisant du vocabulaire visuel en ne conservant que les mots visuels les plus "informatifs" ?*, *Quel sera l'impact d'une étape de sélection d'attributs du vocabulaire sur les performances du classifieur ?*, *Y-aura-t-il aussi un impact*

sur la taille du treillis ?

Motivé par les bonnes performances du treillis approuvées dans le premier système, nous nous sommes inspirés de l'idée présentée dans le chapitre précédent et avons essayé de proposer une amélioration de la représentation des primitives introduites dans le treillis de concepts. Dans ce chapitre, nous proposons une approche à base d'un treillis de Galois de caractéristiques fréquentes que nous appelons *le Treillis de Galois Fréquent (TGF)* utilisé dans un cadre d'apprentissage supervisé.

L'originalité de ce travail réside dans l'idée de construire un treillis de Galois non pas en termes de toutes les caractéristiques visuelles des symboles comme nous avons fait dans le chapitre précédent, mais en termes des caractéristiques les plus fréquentes ou en d'autres termes les mots visuels ayant le pouvoir discriminant le plus élevé après filtrage du vocabulaire. Notre approche consiste premièrement, à créer un vocabulaire visuel, contenant les mots visuels les plus fréquents, à partir de vecteurs caractéristiques définis sur une base d'apprentissage de symboles. Deuxièmement, nous proposons un système de classification de symboles graphiques basé sur le treillis de Galois. Le treillis de Galois obtenu en fonction des motifs fréquents est entraîné et utilisé comme un classificateur. Ceci, nous permet d'aboutir à un prototype nommé *le Treillis de Galois Fréquent*.

Ce chapitre est structuré comme suit : en première partie, nous présentons le contexte de cette contribution et nous présentons une vision globale de l'approche proposée. La section suivante détaille quelques travaux de la littérature s'inscrivant dans le même axe, suivie par une description détaillée des étapes de la chaîne de traitement proposée dans le cadre de la conception d'un treillis de Galois à base de caractéristiques fréquentes. Nous décrivons ensuite l'algorithme de filtrage de vocabulaire visuel proposé. Les résultats sont discutés dans la section finale et des comparaisons avec les approches de la littérature sont aussi présentées.

2.2 Approche proposée

Les méthodes classiques de représentation des symboles nécessitent un stade intensif d'apprentissage en raison de l'utilisation d'un grand nombre de primitives, soit par l'utilisation de la totalité du dictionnaire visuel comme nous l'avons proposée dans le chapitre précédent, soit par la discrétisation des signatures numériques comme dans l'approche de [Guillas *et al.*2005]. Ces deux méthodes ont montré de bonnes performances dans le contexte de la classification mais aussi présentent des inconvénients. Pour les raisons évoqués ci-dessus, et en vue d'atteindre des performances stables et réduire la complexité inhérente du treillis de Galois, il est intéressant de procéder à un filtrage et une sélection intensive des attributs les plus pertinents découlant du dictionnaire visuel. Nous détaillons dans cette section nos stratégies et nos choix concernant la construction du treillis en se basant uniquement sur les attributs pertinents et caractérisant les différentes classes de symboles.

Notre travail s'inscrit dans le cadre de l'extension du système de reconnaissance présenté dans le chapitre précédent en se basant sur une nouvelle structure que nous avons appelée *Treillis de Galois Fréquent (TGF)*. Pour éviter toute confusion, la structure que nous proposons est différente des notions du treillis élaborées dans les travaux de [Agrawal and Srikant1994a] et [Agrawal and Srikant1994b]. Dans ces approches, les auteurs ont introduit les treillis des fermés fréquents où seuls les attributs des concepts sont représentés ou les treillis fréquents utilisant les seuils de fréquence dans le but de limiter la taille du treillis et d'en extraire un

nombre moins important de règles d'association. L'originalité de notre approche consiste à reconnaître notre alphabet (l'ensemble des symboles graphiques) en utilisant un treillis de Galois basé uniquement sur les caractéristiques qui apparaissent fréquemment dans chaque modèle de symboles d'où l'appellation du *Treillis de Galois Fréquent*. Le nouveau treillis est construit non pas en termes de toutes les caractéristiques décrivant le symbole, mais en termes des motifs graphiques les plus significatifs.

Par ailleurs, notre *Treillis de Galois Fréquent* sera nommé TGF tout au long de ce chapitre, est décrit comme suit : $\langle \text{Objets} * \text{Propriétés Fréquentes} \rangle$. L'idée retenue est qu'au lieu d'utiliser des signatures numériques résultantes de la phase de segmentation comme dans [Guillas *et al.*2005], ou d'utiliser tout un dictionnaire visuel comme dans notre première contribution, nous proposons de construire le classificateur en termes de primitives les plus fréquentes. Une étape d'élagage des primitives non significatives est nécessaire afin de filtrer le vocabulaire visuel et retenir uniquement les motifs fréquents qui apparaissent le plus souvent dans chaque modèle de symboles et apportent ainsi une description fine du contenu visuel des symboles graphiques d'une catégorie j . Dans cette section, nous n'al-

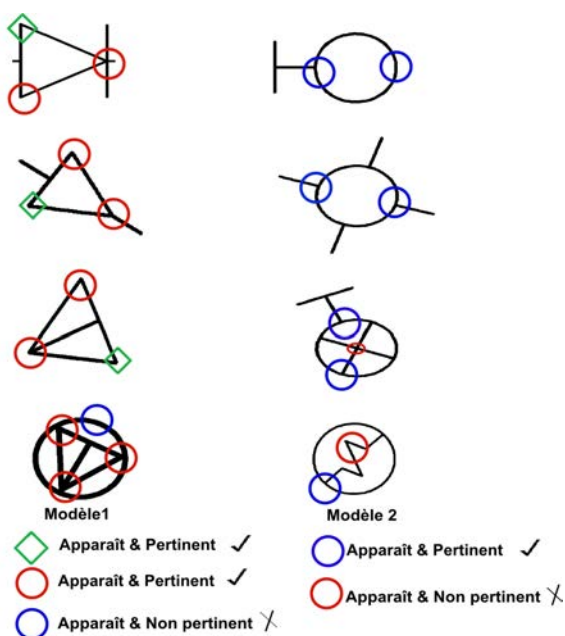


FIGURE 2.1 – Idée de filtrage du dictionnaire visuel en fonction de *l'informativité* et de la *pertinence* des mots visuels apparaissant dans deux modèles de symboles graphiques. \circ , \circ , \diamond et \diamond représentent des mots visuels appartenant au vocabulaire Ω .

lons pas détailler toutes les étapes de la chaîne de traitement proposée dans le cadre de la conception d'un système de classification de symboles basé sur un treillis de Galois car nous avons adopté la même stratégie décrite dans le chapitre précédent. Nous allons nous focaliser sur l'étape du filtrage du vocabulaire visuel représentant la nouveauté apportée par rapport à la première chaîne de traitement. Afin d'évaluer la robustesse du système, nous proposons une évaluation de l'impact de l'étape de filtrage du dictionnaire visuel en termes de ses mots les plus "informatifs", sur le taux de reconnaissance du classifieur.

Nous nous sommes rendus compte que la première méthode (chapitre 1) est susceptible dans certains cas d'inhériter un problème de sur-apprentissage, et pour y remédier nous choisissons de ne garder qu'une partie du dictionnaire qui s'avère la plus informative et

qui représente le plus fidèlement possible un modèle de symbole. *L'informativité* d'un mot dans le dictionnaire visuel est mesurée par sa probabilité d'apparition dans chaque modèle de symboles. En d'autres termes, nous nous sommes inspirés de l'idée exposée dans la Figure.2.1 où nous constatons qu'un même mot visuel peut apparaître dans plusieurs classes de symboles indistinctement, mais à différentes fréquences. Prenons l'exemple du mot visuel \odot , qui apparaît chez les deux classes présentées alors qu'il est caractérisé de "*pertinent*" chez la première classe mais pas pour la deuxième. Ce mot permet de caractériser le premier modèle des symboles donc s'il est fréquent pour cette classe, il est certainement "rare" pour le deuxième modèle malgré qu'il y apparaît vu que les classes ou les modèles des symboles sont différents. Néanmoins, la différence réside dans le fait que sa présence dans le deuxième modèle n'est pas déterminante dans la caractérisation de la classe ou n'apporte pas suffisamment d'information sur cette dernière, à l'encontre du premier modèle où nous pouvons constater que ce mot ou ce motif représente l'*identité* de cette classe de symboles car il apparaît fréquemment dans l'ensemble des symboles faisant partie de cette classe. Similairement, le mot visuel \circ peut être caractérisé comme "*pertinent*" pour la deuxième classe de symboles mais pas pour la première. Nous nous sommes basés sur cette idée afin de renforcer le caractère discriminant du vocabulaire qui aura une influence directe sur les performances de notre système de classification. Le pouvoir discriminant de l'étape de représentation provient de la sélection fine des caractéristiques les plus pertinentes. L'amélioration de la précision des résultats est obtenue par la réduction considérable du coût de calcul et de construction du treillis de concepts ainsi que la robustesse de l'étape de description des objets. Passons à la description de l'approche proposée.

Les étapes de la chaîne de traitement mise en place dans le cadre de la reconnaissance sont décrites dans la Figure.2.2 où nous retrouvons les principales étapes définies dans le chapitre précédent pour la construction du Treillis de Galois à partir du dictionnaire visuel. Notre méthode repose principalement sur le filtrage du dictionnaire visuel résultant de l'étape de pré-traitement en éliminant les caractéristiques les moins informatives et dont la capacité de discrimination n'est pas assez importante et de ne garder ainsi, que les primitives ou les propriétés qui apportent le plus d'informations sur les symboles afin de construire un *Treillis de Galois Fréquent* utilisé comme un classificateur. Le critère utilisé pour filtrer le dictionnaire visuel est la fréquence d'apparition de chaque mot dans les classes de la base des symboles, en fait, si un mot visuel M_{ik} apparaît dans un modèle du symbole S_k avec une probabilité $\Pr(M_{ik}, S_k) \geq \theta$ avec θ est le pourcentage de symboles du modèle k caractérisés par le mot visuel M_{ik} , alors il est caractérisé de *pertinent*. Pour de raison de clarté, nous avons utilisé le terme probabilité pour indiquer la fréquence normalisée comme montré dans la Figure.2.3. M_{ik} est considéré comme une propriété fréquente et elle est retenue pour construire notre TGF. Les mots visuels fréquents sont utilisés pour construire le Treillis de Galois Fréquent qui est utilisé pour reconnaître les symboles.

Dans la suite, nous nous sommes efforcés à détailler spécialement les nouvelles étapes introduites dans le système de reconnaissance par rapport au chapitre précédent. Les principales étapes de notre chaîne de traitement sont les suivantes :

- La première étape consiste à trouver des motifs apparaissant fréquemment dans chaque classe de symboles.
- La deuxième étape utilise ces caractéristiques fréquentes pour créer la table binaire du contexte du Treillis de Galois Fréquent : $\mathcal{I}(o, a)=1$ ssi la propriété a caractérise l'objet o sinon $\mathcal{I}(o, a)=0$.
- Enfin durant l'étape de construction du treillis de Galois, des concepts vides et redondants sont élagués.

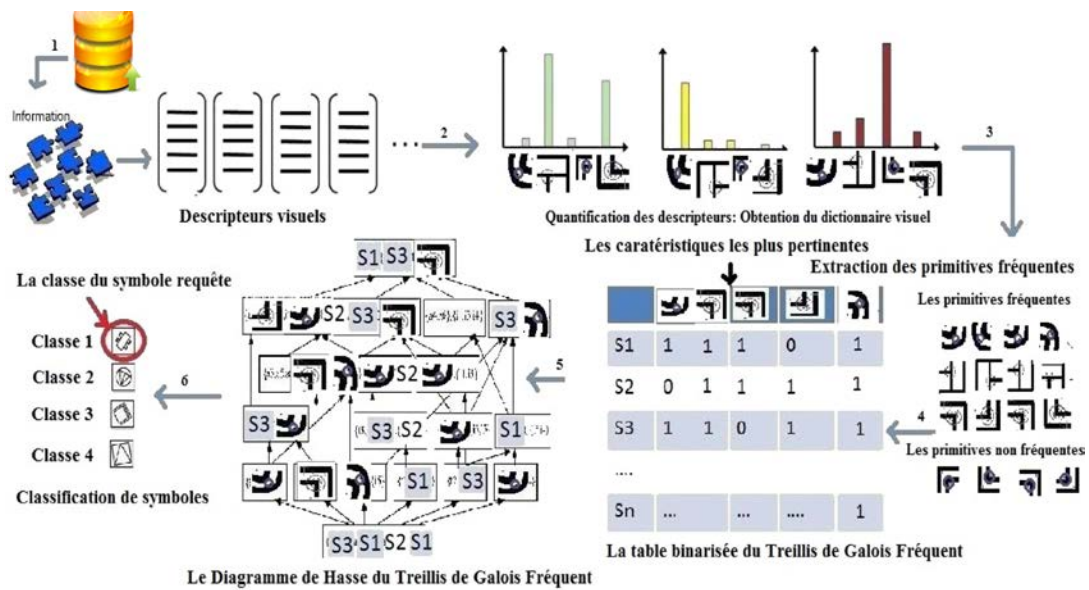


FIGURE 2.2 – L’architecture de notre système proposé pour la reconnaissance de symboles graphiques basé sur la structure TGF. Les étapes sont les suivantes : (1) Extraction et description des primitives à partir des symboles, (2) Quantification des descripteurs visuels et construction du vocabulaire visuel, (3) Filtrage des motifs pertinents et fréquents à partir du vocabulaire, (4) Construction de la table binaire modélisant les relations de $\{Présence/Absence\}$ entre symboles et attributs, (5) Construction du treillis de Galois et reconnaissance du symbole requête et (6) Reconnaissance du symbole requête.

Construction du dictionnaire visuel fréquent

Dans le cadre de l'amélioration de la qualité des caractéristiques visuelles représentant les objets graphiques, [Gilbert *et al.*2009], [Yuan and Wu2008] et [Fernando *et al.*2012] se sont basés sur la fouille des motifs fréquents du vocabulaire visuel. Généralement, ces approches visent à trouver des caractéristiques fréquentes en termes de leur *pertinence* et de leur *informativité*. L'objectif de ces travaux est d'extraire les caractéristiques ou les motifs de plus haut niveau qui sont sémantiquement les plus significatifs et les plus déterminants parmi l'ensemble des motifs. Dans le même esprit, d'autres travaux choisissent d'ajouter plus d'information géométrique locale à leurs algorithmes de génération du vocabulaire visuel. Par exemple, [Lazebnik *et al.*2004] construit un dictionnaire visuel en se basant sur des groupes de motifs voisins dont l'apparence et la configuration spatiale se reproduisent dans l'ensemble d'apprentissage. En outre, il a été observé dans [Perronnin and Dance2007] que dans le cadre des approches classiques à partir de sac de mots avec des bases de données contenant un nombre restreint de catégories d'images, l'utilisation d'un nombre important de mots visuels permet d'obtenir de meilleures performances de classification. Néanmoins, l'augmentation excessive du nombre de mots visuels peut provoquer la saturation du système et cause ainsi la dégradation des performances du classifieur. Notre constat global est que trouver la taille optimale du dictionnaire visuel qui garantit de meilleurs résultats de classification, reste une tâche complexe à résoudre. Dans ce cadre et dans le but de surmonter ces problèmes, nous présentons une méthode efficace pour obtenir des mots représentatifs de la classe visuelle. Cela permet entre autre de déterminer la taille optimale du dictionnaire permettant l'obtention de meilleurs taux de reconnaissance. La répartition

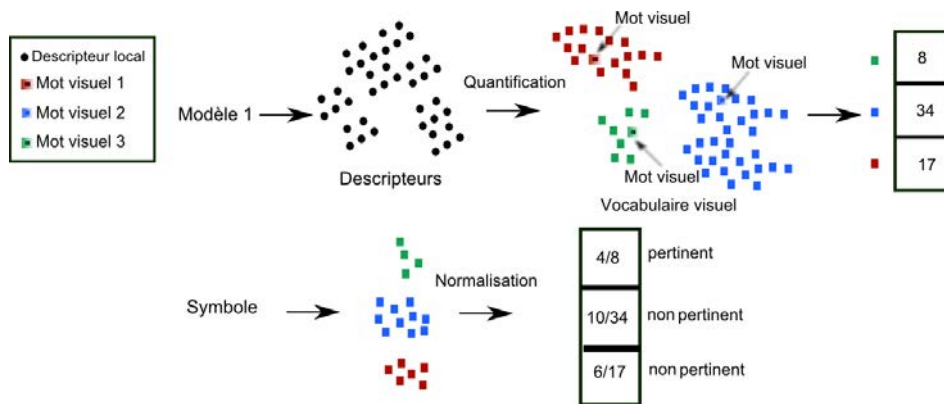


FIGURE 2.3 – Approche adoptée pour le filtrage du vocabulaire visuel en se basant sur les probabilités normalisées (dans cet exemple, le seuil de comparaison est fixé à 50%).

des mots visuels dans l'espace de données n'est pas uniforme, ce qui signifie qu'il y a un plus grand pourcentage de motifs visuels non informatifs et qui ne sont pas assez fréquents ou pertinents et peuvent apporter de grandes incertitudes et ambiguïtés et par la suite diminuer la capacité de discrimination du vocabulaire visuel et nuire à la phase de la reconnaissance. La fréquence d'apparition des mots du vocabulaire diffère d'un mot visuel à un autre. Une solution pour pallier à cette ambiguïté serait d'extraire uniquement les mots visuels les plus répandus dans chaque modèle de symboles ayant un score de *pertinence* normalisé (voir Figure.2.3) supérieur à un certain seuil θ (voir Figure.2.4). Autrement dit, le dictionnaire visuel est susceptible d'être beaucoup moins ambiguï et plus discriminant en éliminant les motifs n'apportant pas assez d'information. Par conséquent, il nous est

paru qu'il est intéressant de découvrir automatiquement les motifs visuels fréquents afin de construire un Treillis de Galois Fréquent ayant une taille raisonnable et un taux de reconnaissance assez élevé approuvé durant les études expérimentales. En analysant le dic-

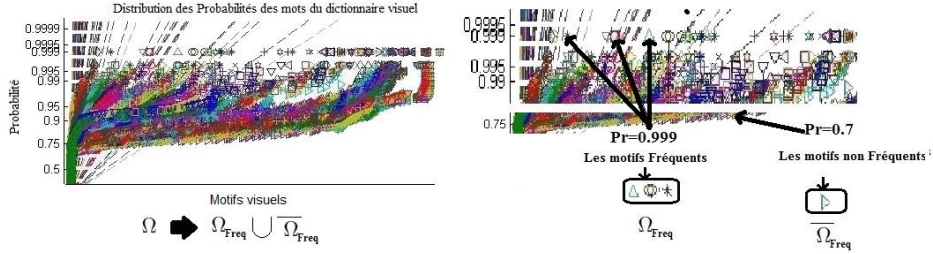


FIGURE 2.4 – Le vocabulaire visuel et la technique de seuillage utilisée pour le partitionnement du dictionnaire visuel : $\Omega = \Omega_{Freq} \cup \Omega_{NonFreq}$.

tionnaire visuel Ω ($\Omega_{Freq} \cup \Omega_{NonFreq}$) (voir Figure.2.4) et en le filtrant en fonction des scores de *pertinence* de ses motifs, on obtient un ensemble de motifs visuels significatifs que nous appelons Ω_{Freq} .

Nous présentons une méthode efficace pour obtenir l'ensemble des mots visuels les plus *pertinents* en termes de leur pouvoir d'*informativité* et de *discriminativité*. Alors, une caractéristique (mot visuel) w_i d'un cluster c_j du vocabulaire visuel (avec $j \in [1, \Omega_{NbClusters}]$) est considérée comme *fréquente* si elle est bien répartie et reproduite entre les symboles d'une même classe k . Cette approche garantit un niveau de discrimination de caractéristiques assez élevé et permet d'éliminer les motifs visuels nuisant à l'ensemble des caractéristiques les plus *pertinentes* et les plus riches en information. Comme nous l'avons détaillé dans l'algorithme.7, dans notre approche nous proposons un nouveau critère pour le filtrage des motifs du vocabulaire les plus représentatifs d'une classe de symboles. Ce critère correspond à un score de *pertinence* que nous appelons $\mathcal{R}_k^{w_i}$:

- $\mathcal{R}_k^{w_i}$: correspond au score de *pertinence* du motif w_i dans l'ensemble des symboles de la classe k .

$$\mathcal{R}_k^{w_i} = \frac{|w_{ik}|}{\sum_{j=1}^{|\Omega|} |w_{jk}|} \quad (2.1)$$

où $|w_{ik}|$ est la fréquence d'apparition du motif w_i d'un cluster c_j du vocabulaire Ω où $j \in [1, \Omega_{NbClusters}]$ dans un modèle k des symboles et $|w_{jk}|$ est le nombre des mots visuels w_j du vocabulaire Ω caractérisant les symboles de la classe k où $k \in [1, NbSymbClasses]$ et $j \in [1, \Omega]$.

En effet, si $\mathcal{R}_k^{w_i} = 1$ alors le motif w_i apparaît fréquemment dans chaque symbole de la classe k et réciproquement si $\mathcal{R}_k^{w_i} = 0$ la classe k n'est pas caractérisée par le motif w_i , néanmoins ce dernier peut apparaître dans d'autres classes de symboles. Notre approche a pour objectif de trouver les motifs du vocabulaire Ω les plus informatifs et les plus représentatifs des classes de symboles :

$$\Omega_{Freq} = \left\{ \begin{array}{l} \{w_{ik} \in \Omega, i \in [1, |\Omega|], k \in [1, NbSymbClasses]\} \\ \mathcal{R}_k^{w_i} \geq \theta \end{array} \right.$$

Si $\mathcal{R}_k^{w_i} \geq \theta$ alors w_{ik} est considéré comme un attribut *fréquent*.

En ce qui concerne les phases de construction et de classification avec le treillis de concepts, nous avons repris les mêmes algorithmes et les mêmes stratégies décrits dans le chapitre

Algorithm 7 Algorithme Proposé

1. *Filtrage du Vocabulaire Visuel*

Entrées w_{ik} un mot visuel

Ω le dictionnaire visuel

S la base de symboles

$\mathcal{R}_k^{w_i}$ la fréquence d'apparition du mot visuel w_{ik} dans la classe de symboles k

θ un certain seuil représentant x% de symboles du modèle k caractérisés par le motif w_i

$\forall w_{ik} \in \Omega$ Faire

si $\mathcal{R}_k^{w_i} \geq \theta$ alors

$\Omega_{Freq} = \text{Empiler}(w_{ik})$

sinon $\Omega_{NonFreq} = \text{Empiler}(w_{ik})$

Finsi

2. *Construction de la table binarisée du treillis de Galois*

Si un symbole S est caractérisé par w_{ik} alors

alors $R(S, w_{ik}) = 1$

sinon $R(S, w_{ik}) = 0$

Finsi

3. *Construction du diagramme de Hasse du treillis de Galois G*

Pour $k=1$ à $NbSymbCat$

Pour $i=1$ à $NbSymbCat$ $\mathcal{C}[k]$

Pour $j=1$ à $NbConcepts$

Pour $j=1$ à $NbAttr$

Algorithme du chapitre.1.6($G, (\mathcal{C}[k], w_{ik})$)

FinPour

FinPour

FinPour

Sorties Le Treillis de Galois Fréquent TGF : $\{Symboles, Motifs Visuels Fréquents\}$

précédent. Les résultats expérimentaux obtenus seront détaillés dans la section suivante.

2.3 Résultats Expérimentaux

Nous étudions les performances de notre système de reconnaissance de symboles graphiques basé sur l'approche décrite dans la section précédente. Un mot visuel w_i est considéré comme un motif fréquent si ce mot apparaît dans un ensemble de symboles S_k , avec une probabilité $\Pr_{i,k}(w_i, S_k) \geq \theta$ avec θ est le pourcentage de symboles du modèle k caractérisés par w_i . En d'autres termes, un mot est dit *pertinent* dans une classe k s'il caractérise au moins $\theta\%$ des symboles de la classe k . L'ensemble des motifs graphiques fréquents est Ω_{Freq} . Dans la Figure.2.5, nous présentons les résultats de la reconnaissance obtenus par notre Treillis de Galois Fréquent combiné premièrement, à la totalité du dictionnaire visuel Ω et deuxièmement au dictionnaire visuel filtré Ω_{Freq} représentant uniquement les motifs les plus fréquents. Ces résultats permettent de juger des améliorations très significatives apportées par l'extraction des mots visuels les plus représentatifs d'une classe de symboles. Les taux de reconnaissance obtenus montrent clairement la robustesse et le pouvoir de l'étape de filtrage du dictionnaire visuel. L'expérimentation présentée dans la Figure.2.5

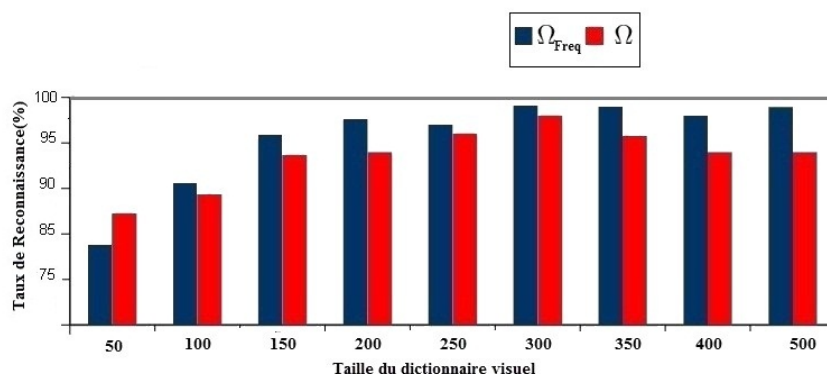


FIGURE 2.5 – Évolution du taux de reconnaissance en utilisant le Treillis de Galois combiné à Ω et Ω_{Freq} en fonction du nombre de caractéristiques. Dans cette expérimentation, la base de symboles GREC2003 est partitionnée en un ensemble d'apprentissage constitué de 1000 instances de symboles et un ensemble de test comportant 500 symboles. $|\Omega|$ varie entre 50 et 500 mots visuels.

permet d'évaluer :

1. Le taux de reconnaissance obtenu par le $TGF_{\Omega_{Freq}}$ lorsqu'on considère seulement Ω_{Freq} qui correspond au dictionnaire filtré en termes des mots les plus fréquents (voir l'histogramme bleu de la Figure.2.5) : les résultats fournis en combinant le treillis de Galois avec le dictionnaire filtré Ω_{Freq} sont plus élevés que ceux fournis par le dictionnaire Ω . Cela peut être expliqué par la suffisance d'information apportée par les motifs fréquents du dictionnaire indépendamment de ceux dont la fréquence d'apparition dans la base de symboles ne dépasse pas le seuil θ , ce qui prouve l'impact des motifs fréquents sur les performances du classificateur. En effet, un constat global que nous pouvons faire après cette analyse est que la capture des mots pertinents ou fréquents rend la discrimination de la forme de plus en plus fine. Les mots fréquents

s'avèrent suffisants pour reconnaître les objets graphiques.

2. L'évolution du taux de reconnaissance en fonction de la variation de la taille du dictionnaire visuel (voir axe x). Le filtrage du dictionnaire visuel peut conduire à la réduction de la taille du vocabulaire par rapport à sa taille originale (de 10% à 90%) en éliminant les mots les moins informatifs en terme de la capacité de discrimination afin d'évaluer les performances de classification dans chaque expérience (l'axe des abscisses x montrant la taille du dictionnaire obtenue suite à plusieurs filtrages). Nous constatons que le taux de reconnaissance s'améliore avec l'augmentation du nombre de motifs visuels. Lorsque la taille du vocabulaire est fixée à 50 mots visuels (ça correspond à 10% de la taille du dictionnaire), le taux de reconnaissance est de 81% en utilisant notre approche $TGF_{\{Symboles, \Omega_{Freq}\}}$ au lieu de 84% pour le $TG_{\{Symboles, \Omega\}}$. Ceci peut être expliqué par le fait qu'il n'y a pas un nombre suffisant de motifs fréquents qui peuvent être extraits à partir du dictionnaire Ω mais durant le reste de l'expérimentation, le taux de reconnaissance augmente beaucoup plus rapidement lorsque le nombre des caractéristiques fréquentes est en augmentation et atteint 98,04% pour le $TGF_{\Omega_{Freq}}$ en comparaison avec 94,6% pour le TG_{Ω} .

2.3.1 Comparaison du TGF avec les autres approches

Comme nous l'avons détaillé dans les sections précédentes, afin de sélectionner les mots pertinents du vocabulaire, nous nous sommes efforcés de prendre en compte la fréquence d'apparition de chaque mot dans chaque classe d'objets. Cette probabilité est comparée à un certain seuil qui indique la borne minimale atteinte par chaque mot pour qu'il soit considéré comme *fréquent* ou *pertinent*. Pour cela, nous avons choisi d'évaluer le comportement du classifieur en fonction de ce critère dont nous avons eu recours durant toutes nos expérimentations. Nous cherchons ainsi à trouver la valeur optimale de ce paramètre pouvant aboutir à de très bonnes performances de reconnaissance.

L'expérimentation illustrée dans la Figure.2.6 vise à varier le paramètre θ afin d'évaluer l'impact de ce dernier sur le taux de reconnaissance retourné par le TGF. Nous remarquons qu'à partir d'un certain seuil atteint par θ ($\simeq 2/3$), le taux de reconnaissance commence à augmenter considérablement. Cette expérimentation suggère de rechercher la valeur optimale du paramètre θ à utiliser comme seuil pour élaguer le dictionnaire et construire un classificateur performant. Nous examinons le paramètre optimal en fonction de la taille du vocabulaire. Nous avons constaté que si la fréquence d'apparition d'un mot est supérieure à θ (en d'autres termes, si le mot visuel se reproduit dans au moins $\theta\%$ des symboles d'un modèle k), le taux de reconnaissance du treillis fréquent est stable. Les mots fréquents extraits du dictionnaire visuel et qui ont la fréquence d'apparition la plus importante dans chaque classe de symboles sont nettement meilleurs et efficaces que le reste des attributs. Nous concluons que la taille du vocabulaire peut être diminuée jusqu'à un certain seuil, sans perte de précision et sans dégrader son pouvoir de discrimination, ce qui implique que le pourcentage des motifs informatifs dans le dictionnaire visuel est beaucoup plus grand que les motifs non informatifs ce qui a une influence directe sur les résultats fournis par le treillis de Galois. Les résultats ont aussi montré que les résultats de classification peuvent se dégrader si un grand nombre de mots visuels sont éliminés du vocabulaire. Comme nous l'avons évoqué dans l'introduction de ce chapitre, cette approche est une extension de la chaîne de traitement de reconnaissance que nous avons exposée dans le premier chapitre de cette partie. Pour cela, nous avons choisi de comparer les deux approches en vue d'obtenir une évaluation objective des performances afin de se rendre compte de l'impact de l'amé-

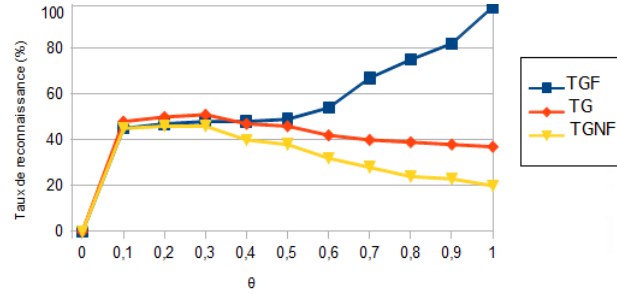


FIGURE 2.6 – Comparaison entre les taux de reconnaissance des classifieurs suivants : $TGF_{\Omega_{Freq}}$, $TG_{\Omega_{NonFreq}}$ et TG_{Ω} en fonction de θ (le seuil à partir duquel nous pouvons parler de la notion de *pertinence* donc possibilité de filtrage du vocabulaire sans perte de discrimination). Dans cette expérimentation, nous avons choisi une taille du vocabulaire $|\Omega|$ entre 50 et 500 mots visuels. Un mot visuel w_i est considéré comme *pertinent* et $w_i \in \Omega_{Freq}$ si sa probabilité d'apparition dans un modèle de symboles k dépasse le seuil θ , sinon $w_i \in \Omega_{NonFreq}$.

lioration apportée grâce au filtrage du vocabulaire. L'objectif de l'expérimentation dont les résultats sont illustrés dans la Figure.2.7 est de positionner cette deuxième contribution par rapport à notre première approche ainsi que par rapport au classifieur de Bayes et au SVM en faisant varier la taille du vocabulaire visuel utilisé avec le treillis. Dans la Figure.2.7(a), nous constatons que le TG_{Ω} est plus performant que le $TGF_{\Omega_{Freq}}$ et cela est dû à l'absence d'un nombre satisfaisant de motifs pertinents pouvant apporter une description fine des symboles d'une classe donnée. Le classifieur SVM s'avère le plus performant dans cette série d'expérimentation ainsi que le réseau Bayésien. Néanmoins dans la Figure.2.7(b), lorsque la taille du dictionnaire est augmentée, la performance de $TGF_{\Omega_{Freq}}$ est significative en comparaison avec le TG_{Ω} ainsi qu'aux autres classifieurs. Ce résultat prouve que

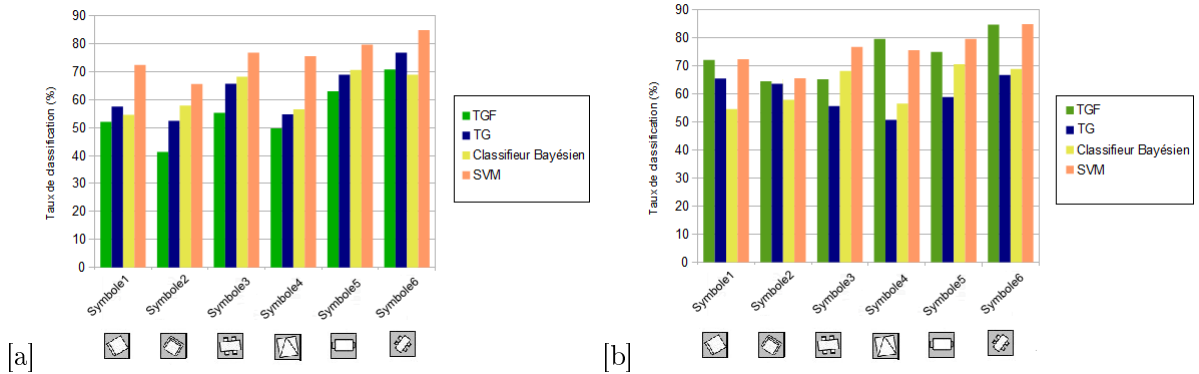


FIGURE 2.7 – Évaluation du taux de reconnaissance en utilisant le TG_{Ω} , $TGF_{\Omega_{Freq}}$, le classifieur Bayésien et le classificateur SVM (dans cette expérimentation, $\theta=2/3$, $|\text{NbSymb}|=500$ et Ω_{Freq} est l'ensemble des mots visuels fréquents. Nous avons testé les classifieurs sur 6 modèles de symboles : (a) $\Omega=50$ mots visuels et (b) $\Omega=100$ mots visuels. Le noyau utilisé pour le classifieur SVM est linéaire.

le fait que le $TGF_{\{Symboles, \Omega_{Freq}\}}$ est significativement meilleur que le $TG_{\{Symboles, \Omega\}}$ cela implique que les mots visuels fréquents largement répandus entre les symboles sont plus informatifs que les mots visuels rares qui possèdent un pouvoir discriminatif assez limité.

Ces résultats prouvent que l'étape de sélection des caractéristiques à partir des symboles graphiques doit fournir les motifs ayant une capacité discriminative assez élevée afin de garantir un taux élevé de reconnaissance. Un constat que nous pouvons faire est qu'en effet, il est fréquent que des mots visuels soient présents dans certains modèles mais ils ne véhiculent pas d'information concernant le modèle auquel ils appartiennent.

La Figure.2.8 montre une évaluation de la capacité de reconnaissance du Treillis de Galois Fréquent en comparaison avec d'autres classificateurs usuels utilisés dans la littérature en fonction de la taille du vocabulaire visuel. Chaque classificateur est appris afin de reconnaître une instance d'un symbole requête. Le taux de reconnaissance fourni par le $TGF_{\{Symboles, \Omega_{Freq}\}}$ est globalement plus élevé en comparaison avec les autres classificateurs comme le classifieur de Bayes et le k -ppv pour une certaine taille du vocabulaire. À partir de 100 mots visuels, le $TGF_{\Omega_{Freq}}$ présente des taux de reconnaissance plus élevés que le reste des classifieurs et cela peut être expliqué par la suffisance du nombre des motifs visuels du vocabulaire et la possibilité de l'extraction des motifs les plus informatifs. Tandis que pour une taille du vocabulaire du 50 mots visuels, le TGF_{Ω} s'avère plus performant par rapport aux autres classifieurs et plus particulièrement par rapport au $TGF_{\Omega_{Freq}}$ car pour ce nombre de motifs, on ne peut pas parler de notion de *pertinence* mais plutôt de *complémentarité* entre les différents motifs du vocabulaire afin de garantir de bons taux de reconnaissance. Comme nous pouvons le voir dans la Figure.2.8, les performances de

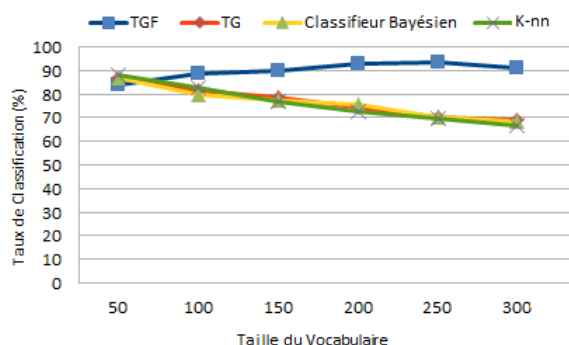


FIGURE 2.8 – Évaluation du taux de reconnaissance (%) des différents classificateurs en fonction de la taille du vocabulaire visuel.

notre Treillis de Galois Fréquent $TGF_{\{Symboles, \Omega_{Freq}\}}$ résultant de notre approche, sont plus élevées (98.4% avec $|\Omega|=250$) par rapport au treillis de Galois de la première contribution $TG_{\{Symboles, \Omega\}}$ (76.08%) et supérieures aussi aux classifieurs de Bayes et k -ppv.

L'évaluation de la complexité d'un tel classificateur en termes du nombre de nœuds, du temps d'apprentissage et du temps de classification est également présentée dans la Table.2.1. Comme nous pouvons le voir, le principal inconvénient d'un treillis de Galois est sa taille qui peut être très importante dans des applications réelles. Dans notre approche, nous proposons de réduire la taille de ce graphe en simplifiant les caractéristiques décrivant les objets tout en essayant de maintenir sa structure et sa capacité de classification. Le TGF est meilleur que le TG de notre première approche en terme de la taille et de la complexité algorithmique. Dans cette série d'expérimentations, nous évaluons les taux de reconnaissance obtenus par le TGF, le TG représenté dans le chapitre précédent ainsi que d'autres classifieurs usuels. Cette expérience indique que la performance du classifieur est fortement dépendante de l'étape de filtrage du dictionnaire. Par exemple, comme le montre

(la ligne 4 du tableau.2.1), nous constatons que le taux de reconnaissance obtenu en utilisant le TGF (89.05%) est nettement meilleur que le taux présenté par le TG de notre première contribution (75.3%). En outre, notre approche garantit un temps de traitement réduit (87.04s au lieu de 253.58s pour le treillis de Galois basé sur tout le dictionnaire). Le temps nécessaire durant l'étape de classification est plus faible dans notre approche parce que le nombre de nœuds est considérablement réduit.
















Modèles	k-ppv (k=4)	Svm (noyau linéaire)	Réseau Bayésien	Arbre de décision	TG	TGF
	74.3%	78.5%	83.7%	55.3%	72.0%	78.9%
	70.3%	73.6%	67.2%	74.3%	42.3%	60.3%
	45.6%	66.1%	67.7%	50.1%	81.02%	87.5%
	65.9%	70.6%	93.0%	76.4%	89.05%	75.3%
	53.0%	63.1%	98.5%	20.3%	79.6%	77.4%
	82.7%	70.6%	80.3%	36.7%	85.3%	95.7%
	85.6%	75.0%	72.4%	83.3%	92.4%	91.2%
	60.04%	56.4%	82.0%	36.7%	79.0%	84.7%
	76.6%	76.4%	87.4%	60.8%	89.87%	83.8%
	74.8%	84.0%	61.4%	55.6%	97.8%	85%
	87.1%	47.9%	47.5%	78.1%	93.65%	85.2%
	68.5%	78.8%	97.5%	37.7%	68.09%	91.5%
	90.75%	77.5%	89.3%	49.6%	80.32%	76.7%
	67.8%	83.2%	70.0%	31.8%	52.0%	63.4%
	33.9%	95.4%	59.0%	70.6%	61.3%	65.8%
Moyenne	69.12%	73.14%	77.12%	54.48%	77.58%	80.16%
Nb _{concept} (Treilliscomplet)	-	-	-	-	2270	1140
Nb _{concept} (GenCon)	-	-	-	-	202	140
Temps _{supp} (Treilliscomplet)(sec)	-	-	-	-	253.58	87.04
Temps _{supp} (GenCon)(sec)	-	-	-	-	1.7	0.9
Temps _{classif} (Treilliscomplet)(sec)	-	-	-	-	26.08	2.6
Temps _{classif} (GenCon)(sec)	-	-	-	-	4.7	2

TABLE 2.1 – Évaluation des performances de la reconnaissance de différents classificateurs : Nb_{concept} (Treillis complet) : le nombre de concepts générés dans le treillis de Galois généré entièrement, Nb_{concept} (GenCon) : nombre de concepts dans le treillis où seulement les concepts nécessaires à la classification sont générés, Temps_{classif} (Treillis complet) : le temps de classification nécessaire pour la recherche de la classe du symbole requête dans le treillis complet, Temps_{classif} (GenCon) : le temps de classification nécessaire pour la recherche de la classe du symbole requête dans le treillis où seulement les concepts nécessaires à la classification sont générés, Temps_{supp} (Treillis complet) : le temps nécessaire pour la construction du treillis complet, Temps_{supp} (GenCon) : le temps de construction nécessaire pour la construction du treillis où seulement les concepts nécessaires à la classification sont générés. La taille du vocabulaire visuel est $|\Omega|$ est fixé à 80. Nous avons utilisé le même protocole d'expérimentation de l'expérience présentée dans le chapitre précédent (le nombre de symboles d'apprentissage et le nombre de symboles test est le même que dans le chapitre précédent pour obtenir une évaluation réaliste). Pour l'algorithme *k-ppv*, nous avons choisi $k=4$ et pour le classifieur SVM le noyau choisi est le noyau linéaire.

L'association entre une partie du dictionnaire des mots visuels Ω_{Freq} au lieu de l'ensemble des primitives représentées par Ω avec le treillis de Galois comme un classificateur, fournit un taux de reconnaissance élevé, en outre, cette approche réduit la taille du classificateur (voir Figure.2.9). Nous pouvons aussi constater dans la Table.2.2 que la méthode utilisée

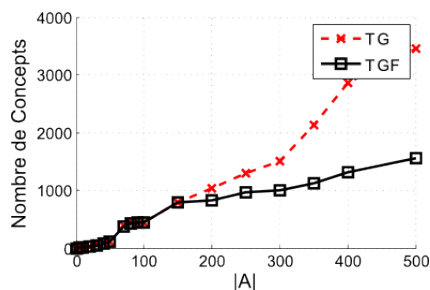


FIGURE 2.9 – Comparaison entre la taille du $TGF_{Symboles, \Omega_{Freq}}$ et le $TG_{Symboles, \Omega}$ en fonction de la taille du dictionnaire visuel.

pour représenter les caractéristiques visuelles d'un symbole graphique montre de bons résultats de classification et de recherche de symboles en termes de *Précision/Rappel*. Cela montre que la technique adoptée pour l'extraction de caractéristiques et leurs descriptions est efficace dans la classification de symboles. Notre approche basée sur le descripteur SIFT appliqué à des régions stables choisis aléatoirement (98,4%) est la plus performante, suivie du descripteur de forme appelé le contexte de forme [Belongie *et al.*2000] (95,6%) et du SIFT classique (94,4%), tandis que le descripteur de Fourier est le plus faible en comparaison avec les autres méthodes (89,3%).

Dans une seconde série d'expérimentations, nous avons choisi d'évaluer notre approche de

TABLE 2.2 – Comparaison entre les différentes méthodes de description des primitives.

Descripteur/Critère	Taux de reconnaissance	Précision	Rappel
Notre approche	98.4 %	0.97	0.52
SIFT	94.4%	0.856	0.41
Contexte de Forme	95.6%	0.9	0.27
Descripteur de Fourier	89.3%	0.75	0.32

représentation de symboles graphiques avec ceux de l'état de l'art. Pour cela, nous avons choisi d'analyser la courbe de *Précision*¹⁶/*Rappel*¹⁷. Cette expérience vise à tester si les primitives proposées pour représenter les symboles graphiques sont suffisamment discriminantes dans le cadre du système de reconnaissance proposé. La performance du système dépend du descripteur choisi dans l'étape de segmentation. Nous avons analysé le rendement de quelques descripteurs présentés dans l'état de l'art à savoir le descripteur *CFPI* utilisé dans [Nguyen *et al.*2009] qui se base sur le calcul du *contexte de forme* au niveau des points d'intérêt détectés par le détecteurs SIFT (plus de détails sur ce descripteur seront étalés dans la troisième partie de ce manuscrit). Nous avons comparé nos résultats avec le descripteur *GFD*[Zhang and Lu2002] et la *R*-signature[Tabbone *et al.*2006] (nommée *RS*

16. $Précision = \frac{|Nb_{SP}|}{|Nb_S|}$ où $|Nb_{SP}|$ est le nombre de symboles pertinents retrouvés et $|Nb_S|$ est le nombre de symboles présents dans la base.

17. $Rappel = \frac{|Nb_{SP}|}{|Nb_{ST}|}$ sachant que $|Nb_{SP}|$ est le nombre de symboles pertinents retrouvés et $|Nb_{ST}|$ est le nombre de total des symboles retournés.

sur les figures). Notre approche repose sur le calcul de SIFT au voisinage stable des points détectés aléatoirement (sur les courbes des Figures.2.10 et 2.11, notre approche est marquée par SIFT).

2.3.2 Invariance aux transformations : Bases GREC_{scale} , GREC_{rot} et $\text{GREC}_{scale+rot}$

Cette partie est dédiée à l'évaluation de l'invariance des descripteurs aux transformations linéaires (transformation d'échelle et de rotation). La Figure.2.10 montre que nos résultats sont prometteurs. Notre méthode qui extrait des points du document d'une façon aléatoire afin de calculer les descripteurs SIFT au niveau de leurs ellipses englobantes est beaucoup plus précise que l'approche [Nguyen *et al.*2009] fondée sur le détecteur SIFT (DoG).

Nous constatons que notre descripteur et le descripteur GFD surpassent les autres descripteurs. RS montre des résultats moins bons en le comparant aux autres méthodes. Le descripteur $CFPI$ bien qu'il s'avère bien performant et invariant aux transformations d'échelle, aux rotations et aux déformations en le comparant à notre approche, notre constat global est qu'il présente quelques instabilités vu qu'il se base sur le détecteur SIFT pour l'extraction des points d'intérêt à partir des images ce qui engendre une mauvaise représentation des points de l'image.

La répétition est alors dégradée ce qui rend la description avec le *contexte de forme* assez difficile surtout pour les symboles possédant un nombre limité de points ce qui est contrairement le cas quand nous appliquons notre détecteur semi-aléatoire $Random$ ¹⁸ qui permet d'extraire les points partout dans l'image. Les points aléatoires retournés offrent étonnement une bonne représentation de l'image car ils sont bien répandus et répartis sur toute l'image (il ne s'agit pas uniquement des coins de Harris ou les maxima et minima locaux de $DoG...$).

La combinaison entre l'approche aléatoire de détection des points à partir des graphiques et l'extraction du voisinage stable des points en se basant sur le détecteur $mser$ permet d'offrir un ensemble de primitives porteuses d'une information très riche et permettra ainsi une description et une modélisation très fines de celle-ci. Cette invariance peut s'expliquer par la robustesse du détecteur $mser$ qui s'est avéré très adapté au contexte de la reconnaissance.

2.3.3 Invariance aux déformations ou occlusions : Bases GREC_{degr1} , GREC_{degr2} et GREC_{dist1}

D'après la Figure.2.11, notre constat global est que le recours à l'extraction du voisinage stable autour des points aléatoires a permis de renforcer le pouvoir discriminant et modélisant du descripteur. Nous remarquons que le descripteur GFD fournit des résultats assez similaires et parfois dépasse notre descripteur dans certains cas. Toutes ces évaluations permettent d'affirmer que la qualité de l'étape d'extraction des primitives est cruciale pour obtenir de bonnes performances de reconnaissance.

18. Pour des raisons de clareté, l'approche de segmentation et de description des symboles utilisée dans ce chapitre, sera détaillée dans le 2^{ème} chapitre de la 3^{ème} partie de ce manuscrit

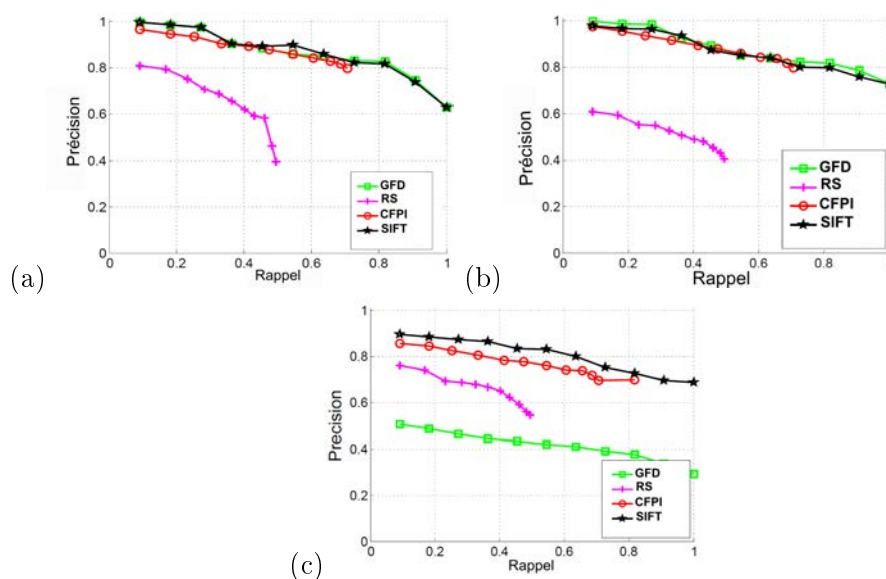


FIGURE 2.10 – Courbes de *Précision/Rappel* pour la reconnaissance de symboles. Dans cette expérimentation, nous avons utilisé les bases de modèles de symboles suivantes : (a) GREC_{scale} , (b) GREC_{rot} et (c) $\text{GREC}_{scale+rot}$ en vue d'évaluer les performances des descripteurs vs notre approche (pour la description des bases voir le chapitre précédent). Nous avons effectué les tests de recherche des symboles similaires à une requête avec les 50 modèles de chaque base. Le nombre moyen d'occurrences de chaque symbole dans ces trois bases varie entre 0 et 13 occurrences.

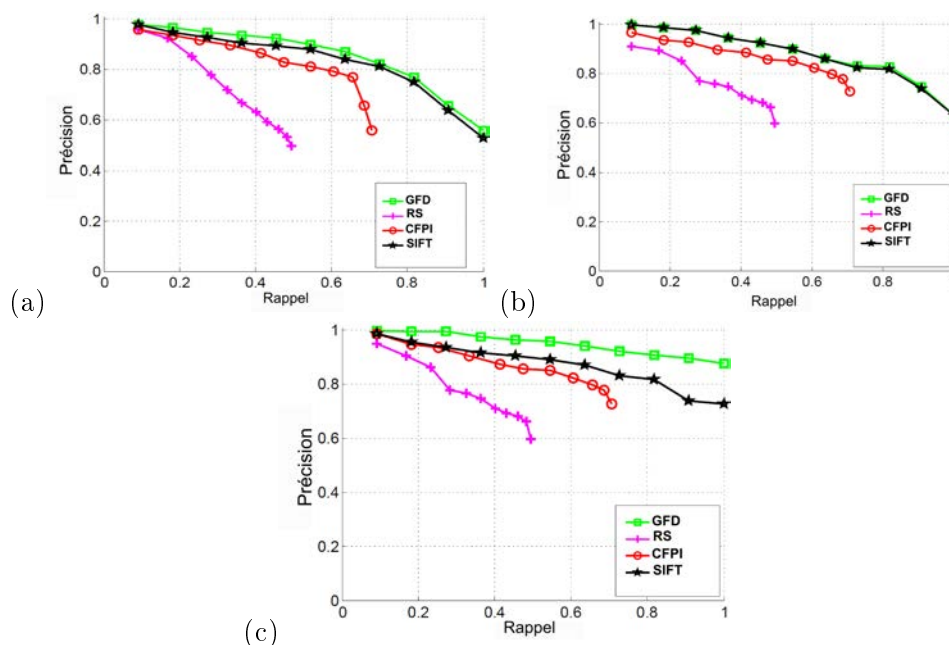


FIGURE 2.11 – Courbes *Précision/Rappel* pour la reconnaissance de symboles. Dans cette expérimentation, nous avons utilisé les bases de modèles de symboles suivantes : (a) GREC_{degr1} , (b) GREC_{dist1} et (c) GREC_{dist2} en vue d'examiner le comportement des descripteurs pour les symboles déformés. Nous avons effectué les tests de recherche des symboles similaires à une requête avec les 50 modèles de chaque base. Le nombre moyen d'occurrences de chaque symbole dans ces bases varie entre 0 et 11 occurrences.

2.3.4 Évaluation des techniques de sélection de caractéristiques

Afin de mener à bien nos expériences, nous avons choisi de positionner notre approche par rapport à la contribution présentée dans le chapitre précédent. En effectuant des expériences avec différentes tailles du dictionnaire visuel, nous étudions le comportement du treillis quand il est combiné avec (a) la totalité du dictionnaire $|\Omega|$, (b) le dictionnaire filtré Ω_{Freq} et (c) le dictionnaire $\Omega_{NonFreq}$. La Figure.2.12 montre les résultats de comparaison de différentes méthodes utilisées : la sélection des variables des descripteurs avec LASSO, la méthode sans sélection de caractéristiques et la sélection des motifs pertinents du vocabulaire visuel décrite dans ce chapitre. Nous constatons que la sélection des mots pertinents en se basant sur notre approche garantit de bons taux de classification en comparaison à la méthode de sélection des variables des descripteurs ainsi que la méthode sans la réduction des caractéristiques. Nous pouvons dire que les motifs pertinents sélectionnés par notre méthode se sont avérés les plus discriminants et les plus informatifs parmi l'ensemble des mots du vocabulaire visuel. En outre, la réduction des composantes des descripteurs peut parfois être moins performante que la sélection des mots visuels du dictionnaire comme est le cas de notre approche. Dans cette expérimentation, la technique utilisée pour filtrer le vocabulaire visuel s'avère la plus performante par rapport à la méthode LASSO ou l'ACP. L'ACP permet de réduire la dimension de l'espace des caractéristiques en proposant des combinaisons linéaires des variables initiales tandis que LASSO permet de sélectionner les variables. Ces deux techniques se sont avérées moins performantes par rapport à l'approche décrite dans ce chapitre ce qui peut être dû à l'absence d'un critère de contrôle du processus de la sélection comme est le cas de notre méthode se basant sur le score de *pertinence* des motifs dans les modèles des symboles graphiques. Les résultats obtenus montrent que la sélection des caractéristiques du vocabulaire visuel indépendamment de la méthode retenue, permet d'améliorer les taux de classification. La réduction de l'espace des caractéristiques permet de gagner en espace ainsi qu'en précision.

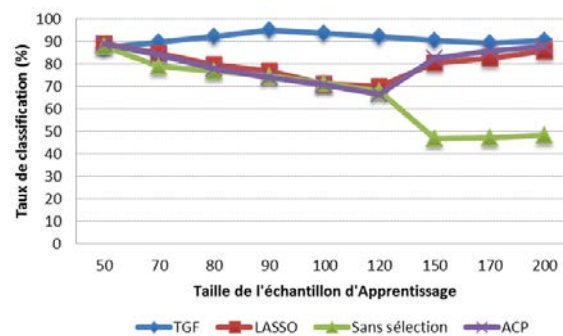


FIGURE 2.12 – Évaluation des techniques de sélection de caractéristiques dans le cadre de la reconnaissance des symboles graphiques en se basant sur le treillis de Galois.

2.3.5 Évaluation du taux de reconnaissance du treillis de Galois en fonction de la taille du dictionnaire

Nous allons étudier l'influence de l'étape de filtrage du dictionnaire visuel sur les performances du treillis de Galois dans le cadre de notre chaîne de traitement de reconnaissance. Cette évaluation fournit une comparaison entre les différentes stratégies adoptées pour la conception du treillis de concepts. Dans notre approche, nous avons opté pour une heuris-

tique de mesure d'informativité et de pertinence de chaque mot visuel afin d'éliminer les mots les moins informatifs avant de procéder à l'étape de construction du classifieur.

Les résultats illustrés dans la Figure.2.13, montrent que la taille du treillis est déterminante pour obtenir de bonnes performances de reconnaissance. Dans le cas où la taille du vocabulaire visuel est inférieure à 100 mots visuels, nous constatons d'après la Figure.2.13(a) et (b) que le taux de reconnaissance du TGF est moins bon que le taux fourni par le treillis de Galois proposé dans notre première contribution. Ceci peut être expliqué par l'insuffisance des attributs dans le dictionnaire ainsi que l'insuffisance du pouvoir discriminant souhaité dans l'étape de pré-traitement qui précède l'étape de classification. La taille réduite du vocabulaire n'assure pas une représentation fidèle de l'information contenue dans les classes de symboles ce qui dégrade les taux de reconnaissance fournis par le classifieur. Néanmoins nous constatons que quand le treillis de concepts est combiné avec la totalité du dictionnaire Ω , le système peut reconnaître les symboles efficacement.

Nous pouvons conclure que si la taille du vocabulaire est trop petite, l'étape de sélection des mots en termes de *pertinence* aboutit à une perte d'information vu que tous les motifs du vocabulaire se complètent pour fournir une description fine de l'information sur la forme de l'objet graphique. Dans ce cas là, nous ne parlons plus de *pertinent* et *non pertinent*, mais nous parlons plutôt de *complémentarité* entre mots. Quand la taille du vocabulaire $|\Omega|$ est supérieure à 300, le TGF basé sur les mots pertinents ($\{\Omega\} \setminus \{\Omega_{NonFreq}\}$) montre de meilleurs résultats en comparaison aux autres méthodes. Ceci pourra être expliqué par le fait que les mots retenus représentent fidèlement et suffisamment les classes des symboles (voir Figure.2.13(c) et (d)).

Nous remarquons également que lorsque $|\Omega|$ varie de 500 à 800, la performance de la TGF reste stagnante. Le TG basé sur les mots non pertinents donne de mauvais résultats. Donc un constat global après cette évaluation est que lorsque la taille du vocabulaire visuel est petite, les mots non fréquents participent d'une façon complémentaire à l'apport et la représentation de l'information sous la forme demandée par le classifieur. Pour cette raison, il sera préférable de garder l'ensemble du dictionnaire et de le céder directement à la structure du treillis de concepts. Il existe inévitablement un biais et un écart entre la performance du treillis de Galois basé sur des attributs fréquents et le treillis de Galois basé sur les attributs non fréquents. Ceci pourra être expliqué par le fait que les mots fréquents sont largement répandus dans la base des symboles et apportent une information suffisante sur les modèles des symboles. Ils représentent une sorte d'*identité* de chaque modèle. Le taux de reconnaissance du classifieur augmente considérablement lorsqu'il est combiné à ces motifs fréquents Ω_{Freq} .

2.4 Conclusion

Dans ce chapitre, nous avons proposé une approche originale pour la reconnaissance de symboles graphiques basée sur le Treillis de Galois Fréquent combiné à un dictionnaire visuel filtré. Le fort essor qu'a connu l'Analyse Formelle de Concepts depuis ces dernières décennies, nous a incité à favoriser et intégrer le treillis de Galois dans le cadre des systèmes de reconnaissance de symboles. L'utilisation de cette structure pour la reconnaissance d'objets graphiques a montré des résultats prometteurs. Le principe de notre système est basé sur une phase de pré-traitement très sélective. Le vocabulaire visuel est filtré en fonction des mots les plus pertinents. Nous avons choisi de capturer les motifs qui apportent le plus d'information sur les différents modèles de symboles de sorte à affiner la discrimination et

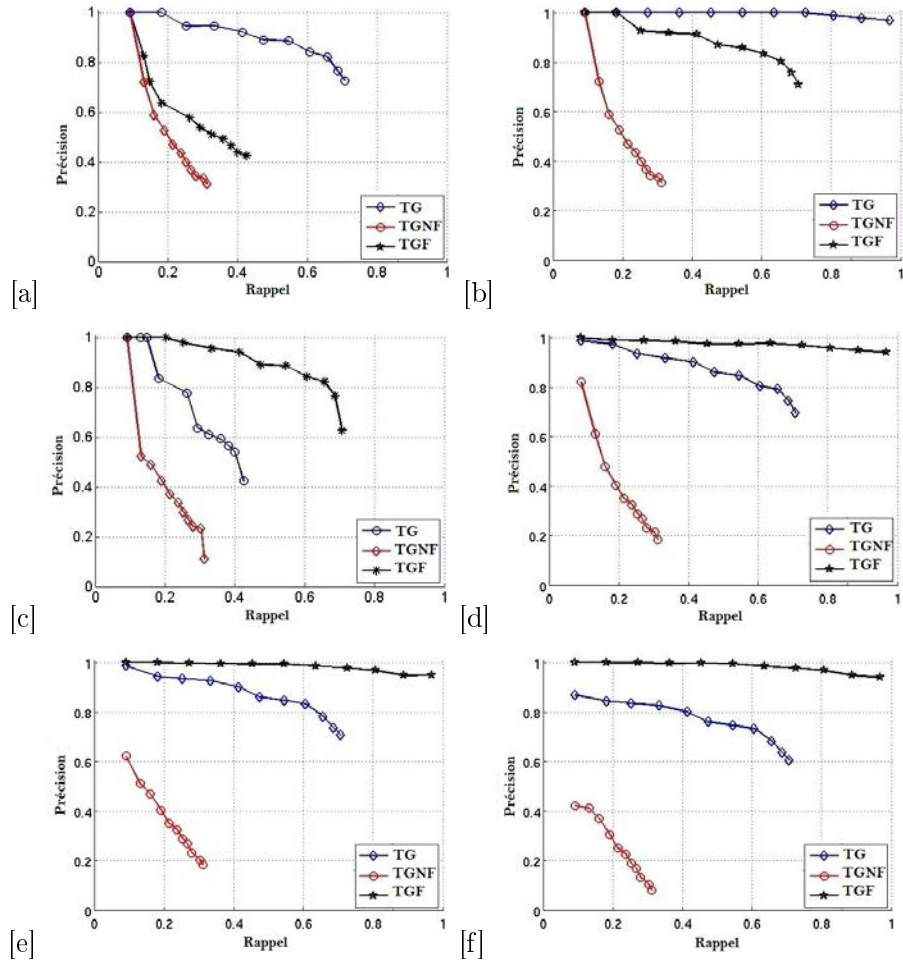


FIGURE 2.13 – Comparaison des performances obtenues par les trois structures du treillis de Galois suivants : $TGF_{\Omega_{Freq}}$, $TG_{\Omega_{NonFreq}}$ et le TG_{Ω} en fonction de la taille du vocabulaire visuel utilisé (a) $\|\Omega\|=50$, (b) $\|\Omega\|=100$, (c) $\|\Omega\|=300$, (d) $\|\Omega\|=400$, (e) $\|\Omega\|=500$, (f) $\|\Omega\|=800$.

la description de la forme des objets. En s'appuyant sur cette heuristique, le vocabulaire est réduit en termes de ses mots les plus pertinents. Ce vocabulaire filtré s'avère ainsi très suffisant pour décrire les symboles de la base. Nous avons souhaité évaluer le choix de cette stratégie et son impact sur les performances du treillis de concepts choisi comme classifieur. Ainsi, les caractéristiques visuelles qui apparaissent fréquemment dans les modèles des symboles graphiques sont les attributs qui constituent l'entrée du treillis de Galois. Une comparaison a été proposée entre l'utilisation d'un dictionnaire visuel filtré en termes des motifs les plus fréquents pour la construction du treillis de Galois et l'approche basée sur le dictionnaire entier qui représente le cœur de notre première contribution exposée dans le chapitre précédent. Les résultats expérimentaux montrent la pertinence et la robustesse de notre approche. La principale nouveauté de ce travail est l'utilisation des motifs graphiques fréquents pour construire notre classificateur ce qui garantit un taux de reconnaissance assez élevé et une taille très réduite ainsi qu'un temps de traitement raisonnable durant le processus de classification. Notre travail élargit et étend le domaine de l'Analyse Formelle de Concepts en démontrant que le treillis de Galois peut être utilisé pour la description et la reconnaissance des symboles segmentés. Notre défi est d'appliquer ce genre de démarches pour reconnaître les symboles non segmentés dans le contexte de la localisation de symboles. Cela fera l'objet de la 3^{ème} partie de ce manuscrit. De plus, nous avons essayé d'explorer la capacité du treillis de Galois dans d'autres applications de la vision par ordinateur et plus particulièrement dans une application de classification d'images. Ce travail peut nous ouvrir la voie à mieux explorer la structure dans plusieurs contextes et dans plusieurs applications.

3

Application à la Reconnaissance d'Images

"Tous les modèles sont faux, mais certains sont utiles".

Alain Faure

Sommaire

3.1	Introduction	119
3.2	État de l'Art	121
3.3	Le Formalisme du Treillis de Galois	123
3.4	Formulation du Problème	123
3.5	Approche Proposée	124
3.6	Résultats Expérimentaux	126
3.6.1	Comparaison avec les classifieurs de l'état de l'art	127
3.7	Évaluation des performances	133
3.7.1	Évaluation des performances du treillis en fonction du nombre d'attributs	133
3.7.2	Évaluation du taux de reconnaissance en fonction de la taille de l'échantillon d'apprentissage	134
3.8	Étude de la Complexité du treillis	134
3.9	Conclusion	135

3.1 Introduction

Les évaluations mises en place dans les deux contributions exposées dans les chapitres précédents, ont permis d'ouvrir la voie à tester le formalisme du treillis de concepts dans un contexte plus général. Pour ce faire et à l'égard des raisons évoquées auparavant, nous avons choisi de l'appliquer dans le cadre de la reconnaissance des images comme application à ce classifieur. La classification des images reste un défi majeur pour la communauté de vision par ordinateur. L'un des développements les plus significatifs de la dernière décennie est l'application des caractéristiques locales pour la classification d'images, y compris l'introduction de la représentation par sac-de-mots-visuels qui inspire et initie beaucoup d'efforts de recherche.

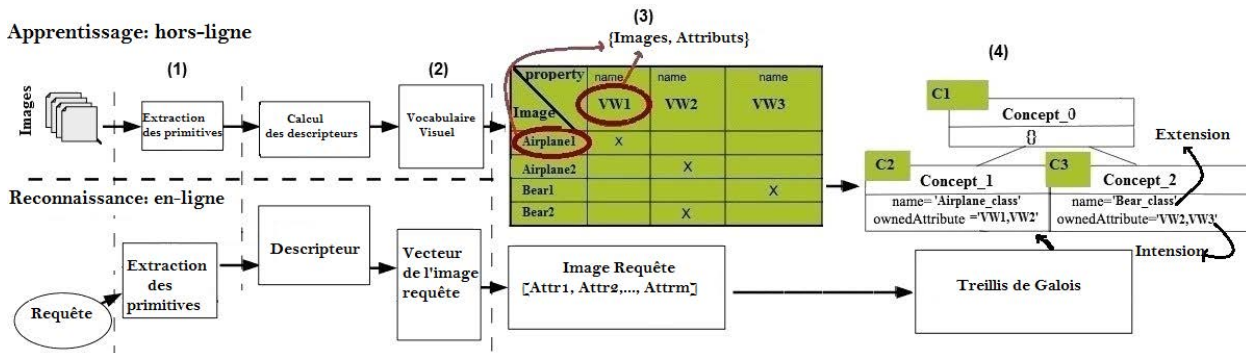


FIGURE 3.1 – L’architecture proposée pour la chaîne de traitement de reconnaissance d’images en se basant sur le treillis de Galois_{image}.

La plupart des méthodes existantes reposent sur la même architecture : (a) les caractéristiques les plus discriminantes sont extraites d’une image requête ; (b) puis un classifieur est conçu pour assigner une classe à l’image requête. Cette contribution est définie dans la perspective d’appliquer les chaînes de traitements proposées dans les chapitres précédents au contexte de l’image. Nous nous sommes intéressés au problème de la recherche d’un ensemble d’images similaires à une image requête organisées dans une structure de treillis de Galois_{image}.

La motivation principale de cette approche est de montrer la capacité du treillis de Galois_{image} à être appliqué dans le contexte de la classification des images. Pour nous positionner par rapport aux travaux de la littérature, nous avons choisi de nous comparer aux approches de l’état de l’art qui sont les plus proches tout en tentant d’utiliser les mêmes protocoles d’expérimentations adoptées dans ces travaux. L’architecture de notre chaîne de traitement est illustrée dans la Figure 3.1 :

1. Extraction et description de l’ensemble des images fournissant un ensemble de descripteurs caractéristiques.
2. Quantification des signatures en utilisant l’algorithme de *k-means* afin de construire le vocabulaire visuel.
3. Sélection des mots visuels les plus pertinents en se basant sur l’algorithme présenté dans le chapitre précédent.
4. Construction de la table du treillis de Galois basée sur la relation binaire entre $\{Images, Caractéristiques\}$.
5. Classification d’une image requête en retournant l’ensemble des concepts qui sont susceptibles de contenir les occurrences similaires à l’image requête.

Ce chapitre est organisé comme suit. Dans la section 2, nous présentons brièvement quelques travaux s’inscrivant dans le contexte de la classification d’images. Dans la section 3, nous exposons une description détaillée des étapes de la chaîne de traitement mise en place. Ensuite, les résultats des expérimentations et l’évaluation des performances de notre approche basée sur le treillis de Galois sont étalés. L’étude de la complexité spatiale et temporelle de notre outil sera présentée dans la section suivante.

3.2 État de l'Art

En vision par ordinateur, de nombreux efforts de recherche ont été consacrés à la catégorisation d'objets. Ces approches peuvent être classées en trois approches complémentaires : (a) les systèmes de recherche et d'indexation d'images basés sur un scénario entre machine-utilisateur. Ce dernier introduit sa requête auprès du système qui se charge d'assigner une classe candidate à la requête ainsi que l'ensemble des occurrences des images qui lui sont similaires. Ce processus est précédé d'une étape de description et de spécification de caractéristiques visuelles des images ; (b) les systèmes interactifs avec contrôle de pertinence où l'utilisateur est impliqué dans un processus itératif en indiquant si les images retournées par le système sont pertinentes et correspondent à l'image mentale qu'il a de son image ou pas. Le système se base sur le retour de l'utilisateur pour affiner sa recherche ; (c) et enfin les systèmes de recherche basés sur la navigation dans un espace organisé. Dans ce contexte, une ligne de recherche émergente [Boiman *et al.*2008], [Bosch *et al.*2007], [Duchenne *et al.*2011], [Yang *et al.*2009] et [Gehler and Nowozin2009] consiste à concevoir des classificateurs dédiés à cet objectif et qui ont montré des résultats de classification prometteurs. Les classificateurs ont récemment reçu une attention accrue [Bosch *et al.*2007], [Im and Cho2006], [Boiman *et al.*2008], et ont montré de bonnes performances dans la tâche de classification d'images. Néanmoins ces classifieurs qui reposent sur des modèles paramétriques [Boiman *et al.*2008] nécessitent une phase d'apprentissage intensive précédée d'une étape de configuration de paramètres relatifs au nombre de classes (par exemple, la configuration des paramètres de SVM [Bosch *et al.*2007], [Boiman *et al.*2008], [Yang *et al.*2009], [Duchenne *et al.*2011], [Gehler and Nowozin2009], arbres de décision et les réseaux Bayésiens [Im and Cho2006]).

Pour pallier à ces inconvénients, nous avons choisi d'appliquer la structure du treillis de Galois qui ne nécessite aucun paramétrage précédant la phase de classification. Le dispositif utilisé dans notre approche est adapté à ce problème en raison de l'absence de méthodes à base de poids, et même des distances utilisées habituellement dans d'autres types de graphes [Im and Cho2006]. En tant que structure de navigation, notre intérêt au treillis de Galois se justifie par les nombreux avantages qui les présente : (a) tout d'abord, il est très rapide pour naviguer à travers une structure de graphe : le temps nécessaire de navigation d'un nœud à un autre est optimal, c'est à dire en $O(1)$, (b) la distance qui sépare le nœud *inférieur* et le nœud *supérieur* du graphe est au plus polynomiale en fonction du nombre de propriétés utilisées (dans notre cas, ces propriétés correspondent à la taille du vocabulaire visuel) ; (c) tout en tentant de conserver l'aspect sémantique entre $\{Images, Attributs\}$, le treillis de Galois est par défaut le meilleur outil qui garantit un tel aspect vu sa structure dédiée principalement à la structuration et la hiérarchisation des relations entre l'ensemble d'images et l'ensemble de leurs caractéristiques visuelles, d'ailleurs c'est ce qui le différencie des autres structures de graphes.

Comme il est de coutume de positionner le travail proposé par rapport aux contributions de la littérature s'inscrivant dans le même contexte, nous nous sommes efforcés à fouiller dans l'état de l'art pour trouver les travaux qui s'avèrent liés à ce que nous proposons, néanmoins un constat global est que malgré la prolifération de cette structure dans le domaine de la fouille ces dernières années, il existe peu de travaux qui y sont dédiés. Nous avons pu repérer deux travaux qui sont adéquats avec notre approche à savoir l'approche proposée par [Emmanuel and Samuelides2004] et [Martinez and Loisant2002] qui abordent le treillis comme une structure de navigation dans les bases d'images plutôt que de le considérer comme classifieur. En effet, la structure du treillis de Galois offre naturellement une

manière efficace, efficace, facile et intuitive pour naviguer. Mais, cette structure mathématique n'a pas été réellement appliquée et évaluée dans le cadre de la classification des images. En raison de l'absence des travaux de la littérature fondés sur cette structure, nous comparons notre approche à celles proposées dans [Im and Cho2006], [Bosch *et al.*2007], [Boiman *et al.*2008], [Zhou *et al.*2010], [Yang *et al.*2009], [Gehler and Nowozin2009] où les auteurs présentent une approche de classification des images en fonction du contenu visuel.

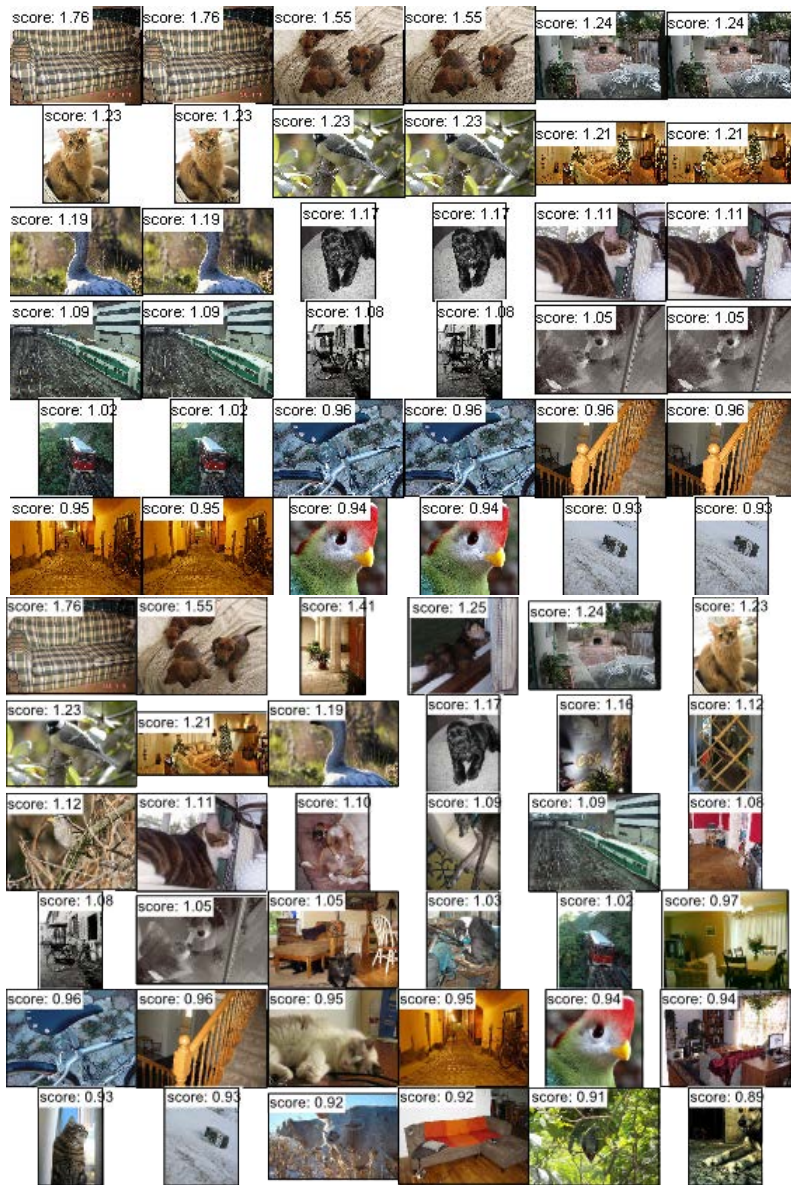


FIGURE 3.2 – Exemples des résultats obtenus dans le cadre d'un système de reconnaissance d'images.

3.3 Le Formalisme du Treillis de Galois

Les principes et les algorithmes de construction de la structure du treillis de Galois ont été largement exposés dans le chapitre *Treillis de Galois et Analyse Formelle de Concepts* et le lecteur peut s'y référer pour une présentation formelle détaillée. Comme nous l'avons signalé, la performance du treillis de Galois dépend généralement des algorithmes utilisés soit pour la construction et l'apprentissage, soit pour la classification. Le regroupement conceptuel sur lequel repose le treillis concerne la structuration et la hiérarchisation des images non étiquetées en différentes classes en fonction de leurs descriptions. Une caractéristique importante des méthodes de classification conceptuelles est que toute classe en plus d'être caractérisée par sa partie *extension* (l'ensemble des objets d'une même classe \mathcal{C}_k), est également caractérisée par une partie *intension* (désigne la description des images résultante de la phase de représentation). Une autre caractéristique souvent souhaitable dans n'importe quel classifieur possédant la structure de graphes, est que le processus de formation des classes soit progressif, c'est-à-dire le traitement du $n^{\text{ième}}$ objet ne nécessite pas le traitement des $(n-1)$ objets qui le précèdent dans le graphe.

3.4 Formulation du Problème

Notre travail consiste à projeter le treillis dans une chaîne de traitement dans le contexte de la classification des images, pour cela nous avons appelé la structure utilisée par le treillis de Galois_{image} permettant de regrouper des images partageant les mêmes descriptions dans une même catégorie (voir la Figure.3.3).

La Figure.3.3 présente un exemple illustratif de l'architecture du système proposé : le concept *supérieur* du treillis de Galois est $C_0 = \{\text{img1}, \text{img2}, \text{img3}, \text{img4}\}$, possède deux descendants qui sont respectivement $C_1 = \{(\text{img1}, \text{img2}), (\text{Attr1})\}$ et $C_2 = \{(\text{img1}, \text{img3}, \text{img4}), (\text{Attr2})\}$. Le processus au fur et à mesure qu'on descend dans l'arborescence, consiste à regrouper des images partageant les mêmes descriptions ou les mêmes caractéristiques dans un même nœud ou un même concept. Intuitivement, nous nous intéressons à l'ensemble des images qui partagent exactement la même description des caractéristiques ainsi que celles partageant le plus grand nombre de descriptions. Une étape de classification consiste à récupérer la classe d'une image requête à l'aide du treillis de Galois_{image} (voir Figure.3.2). La partie la plus encourageante de cet algorithme est que les caractéristiques extraites peuvent facilement "alimenter" les *intensions* des nœuds du classifieur utilisé. La stratégie de classification adoptée est détaillée dans la section suivante. La partie *extension* des concepts retenus lors de la phase de classification comprend l'ensemble des images partageant les mêmes propriétés que l'image requête. Les catégories des images sont indexées par une description conceptuelle, et reliées au moyen d'une relation de {généralisation/spécification}.

En outre, la structure de navigation que nous utilisons est flexible ce qui la rend très intéressante dans ce contexte. Les autres types de graphes permettent un regroupement hiérarchique strict où chaque classe possède exactement un parent, à l'encontre de la structure du treillis où plusieurs chemins peuvent mener à une même classe ce qui diminue la probabilité de tomber sur des faux concepts. Deuxièmement, la même catégorie est souvent pertinente pour deux ou plusieurs requêtes qui arrivent à avoir des descriptions comparables. Par conséquent, la capacité de faire face à des classes non-disjointes est une caractéristique importante des systèmes de recherche. Cette capacité à intégrer des connaissances sur les

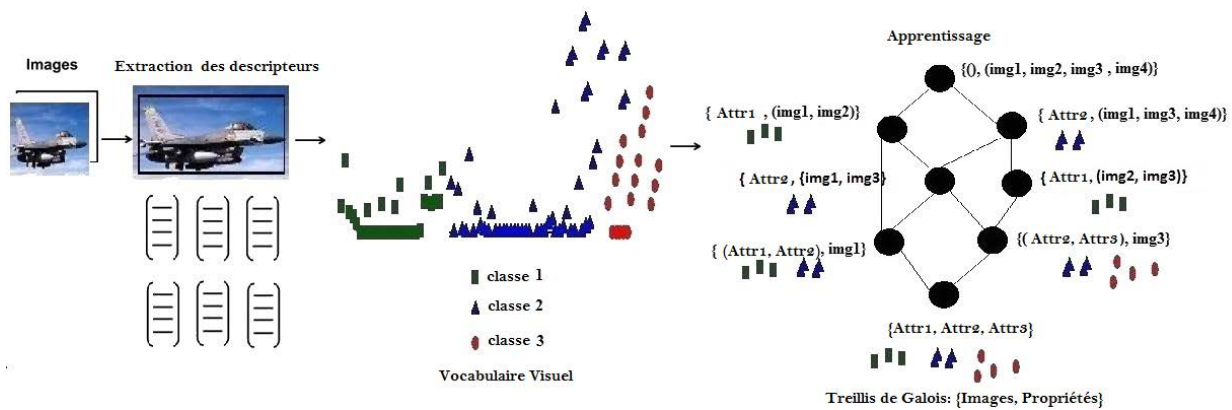


FIGURE 3.3 – Les différentes étapes de la construction du treillis de Galois_{image} basé sur la représentation par sac de mots.

objets (images semblables sont regroupées dans le même nœud) facilite la mesure de similarité et réduit la complexité en temps : il y a un nombre restreint de nœuds visités au cours de l'étape de classification qui peut améliorer considérablement les performances de reconnaissance du classifieur. La plupart des systèmes de classification peuvent traiter certains de ces problèmes, mais pas tous. La motivation du choix du treillis est sa capacité à regrouper un grand nombre de descriptions qui découlent de l'étape de description des catégories des images.

3.5 Approche Proposée

Comme mentionné précédemment, le système de reconnaissance d'images proposé en utilisant le treillis de concepts vise à construire un tableau binaire où les images correspondent aux lignes de la table et les caractéristiques résultant du vocabulaire visuel correspondent aux colonnes (voir Table.3.1). Une valeur booléenne dans une cellule donnée indique si une image donnée possède un certain attribut ou non. Nous appliquons la même stratégie décrite dans le chapitre précédent. La relation entre {images, attributs} est une relation de {Présence/Absence}. La construction du classifieur est abordée de la même façon détaillée dans le chapitre de la classification de symboles.

La classification d'une image requête revient à calculer les similarités entre cette dernière est celles apprises dans le treillis. Une comparaison entre les parties *intensions* des concepts existants est mise en place. Les concepts partageant les mêmes attributs que l'image requête sont susceptibles de contenir des instances des images semblables à l'image passée en requête. Ces étapes seront détaillées dans les sections suivantes. Vu que l'algorithme de construction de cette structure est déterminant pour les performances de la classification et joue un rôle essentiel dans l'intégration du treillis de concepts dans différentes applications, nous avons opté pour les raisons évoquées ultérieurement pour l'algorithme de Norris. L'efficacité et la performance des algorithmes de treillis de concepts sont très différentes l'un par rapport à l'autre. La génération des concepts nécessaires à la classification uniquement, a permis de gagner en temps, en espace et en performance vu que le nombre de concepts du treillis est considérablement réduit en comparaison avec la génération de l'ensemble des concepts ce qui a offert aussi une bonne navigation dans la structure tout le long de la

Catégories	Attributs															
	Attr ₁	Attr ₂	Attr ₃	Attr ₄	Attr _m	Attr _{m+j}	Attr _{n-1}	Attr _n	
image ₁	X			X	X		X	X	X	X	X	X	X		X	
image ₂	X	X					X	X	X	X	X	X	X		X	
image ₃		X					X		X		X	X	X	X		X
image ₄	X		X	X			X	X	X	X	X	X	X		X	
image ₅	X			X			X	X	X	X	X	X	X		X	
image ₆		X		X			X		X		X	X	X		X	
image ₇	X		X				X		X	X	X	X	X		X	
image ₈	X	X					X	X	X		X	X	X		X	
image ₉	X		X				X	X	X		X	X	X		X	
image ₁₀		X		X			X	X	X	X	X	X	X		X	
image ₁₁		X		X			X	X	X	X	X	X	X		X	
...	X			X	X		X	X	X		X	X	X		X	
image _p	X		X	X			X	X	X	X	X	X	X		X	
image _{p+1}	X						X	X	X	X	X	X	X		X	
...		X	X				X	X	X		X	X	X	X		X
...	X		X	X			X	X	X	X	X	X	X		X	
image _{p+m}		X	X	X	X		X	X	X	X	X	X	X		X	

TABLE 3.1 – La table binaire du treillis de Galois : le contexte \mathcal{K} est le triplet $(\mathcal{O}, \mathcal{A}, \mathfrak{S})$ où \mathcal{O} est l'ensemble d'images, \mathcal{A} est l'ensemble d'attributs et \mathfrak{S} une correspondance de $\{\mathcal{O} \times \mathcal{A}\}$ dans $\{0, 1\}^{<\mathcal{O} \times \mathcal{A}>}$ où $\mathfrak{S}(\mathcal{O}, \mathcal{A}) = 1$ si \mathcal{O} est caractérisé par un attribut \mathcal{A} sinon $\mathfrak{S}(\mathcal{O}, \mathcal{A}) = 0$.

phase de classification.

Classifier une image avec le treillis de Galois

L'objectif principal de notre chaîne de classification consiste à retourner les images qui sont les plus vraisemblables à une image requête en termes de leurs caractéristiques contenues dans les parties *intension* des concepts du classifieur. Pour affecter une classe à une image requête, le système effectue un calcul de similarité entre le nouvel objet et chaque concept résultant de l'étape d'apprentissage. La similarité entre un objet et un concept est le nombre de propriétés communes entre ces derniers. Le système attribue alors la classe de l'objet du concept le plus similaire.

Comme nous l'avons expliqué, le treillis est complet pour cette raison il y aura exactement un seul concept qui partage exactement les mêmes attributs que la requête. La liste des concepts sont retournés dans leur ordre décroissant de pertinence vu que la navigation au cours de la phase de classification commence du bas vers le haut. Donc en partant vers le haut (vers le *top*), le nombre des attributs partagés est moindre donc les concepts trouvés au niveau inférieur sont plus pertinents que ceux en haut de l'arborescence.

Dans le cas de multi-concepts, le système offre également un processus empirique qui assigne à chaque objet la classe majoritaire parmi les concepts pertinents retenus. La pertinence d'un concept est mesuré en terme du nombre des attributs partagés avec la requête. Le treillis de concepts peut être considéré comme un espace de recherche dans lequel nous évoluons par la validation des mots visuels partagés issus de l'étape de génération du dictionnaire visuel.

Au cours de la phase de classification, l'ensemble des mots $V_{req} = (V_1, \dots, V_n)$ de l'objet à reconnaître sont introduits dans le treillis de concepts. Notons que toutes les classes d'images sont candidates aucune *intension* n'est encore validée. Le nombre des attributs communs entre $intension(V_{req}) = (V_1, V_2, \dots, V_m)$ et chaque *intension* du concept $V_j = (V_1, V_2, \dots, V_n)$ est calculé. Notons que la recherche du concept similaire est de bas en haut (du nœud *bottom* (\emptyset, \mathcal{A}) vers le nœud *top* (\mathcal{O}, \emptyset)).

$$Sim(intension(V_{req}), intension(V_j)) = \frac{intension(V_{req}) \cap intension(V_j)}{intension(V_{req}) \cup intension(V_j)} \quad (3.1)$$

Dans le cas où plusieurs concepts sont retenus, ils sont en ordre décroissant de leur pertinence cela veut dire qu'un concept partageant m attribut avec la requête est dit plus pertinent que le concept C_j lui aussi partageant n attributs avec la requête *si et seulement si* $m > n$. Ce nombre est ensuite utilisé pour calculer un score qui servira pour le processus de vote qui a pour objectif le choix d'un seul concept parmi la liste des concepts retenus. Un concept ayant m primitives dans l'*intension* doit recevoir 2^m-1 votes si tous ses m attributs ont été correctement identifiés. Le score obtenu est utilisé pour le vote afin de choisir un seul nœud résultat et retourner ainsi son *extension* comme l'ensemble des images appartenant à la même catégorie que l'image requête.

Algorithm 8 Algorithme de classification utilisé

- 1: **Données**
- 2: \mathcal{P} : l'ensemble des concepts du treillis \mathcal{G}
- 3: x_k : une image requête appartenant à une catégorie k
- 4: \mathcal{A}_k : les attributs de l'image requête
- 5: Chercher l'ensemble des images x_i où $i \in [1..n]$ partageant les mêmes attributs \mathcal{A} que x_k
- 6: **Sortie** : la partie *extension* des concepts retenus $\in \mathcal{P}_i$
- 7: **Début**
- 8: Pour $j = 1$ à NbConcepts
- 9: Pour $i = 1$ à NbAttrConcept $\mathcal{O}[j]$
- 10: Durant la phase de classification, le principe est de comparer les attributs de x_k avec la partie *intension* des concepts du treillis de Galois

$$Sim(intension(x_{ki}), intension(C_{ji})) = \frac{intension(x_{ki}) \cap intension(\mathcal{P}_{ji})}{intension(x_{ki}) \cup intension(\mathcal{P}_{ji})} \quad (3.2)$$

où *Sim* est le nombre d'attributs communs et vérifiés au cours du processus de navigation de bas en haut en passant d'un niveau à un autre dans l'arborescence du treillis

- 11: ConceptsRetenus=List($\{\mathcal{P}[i], \mathcal{P}[i+1], \dots, \mathcal{P}[n]\}$) partageant le plus grand nombre d'attributs communs avec la requête x_k .
 - 12: si $|\text{List}|=1$ cela veut dire que le processus de classification a abouti à un seul concept partageant **exactement** les mêmes attributs que x_k . Dans ce cas retourner ce concept dont la partie *extension* contient l'ensemble des images similaires à la requête.
 - 13: sinon si $|\text{List}| > 1$, dans ce cas les concepts retenus sont dans l'ordre décroissant de leur pertinence. Le concept le plus pertinent (partageant le plus grand nombre d'attributs avec l'image requête) est retourné.
 - 14: Un vote est procédé sur la liste : List($\{\mathcal{P}[i], \mathcal{P}[i+1], \dots, \mathcal{P}[n]\}$)
 - 15: Concept_{retenu}= $\mathcal{P}[\text{max}]$ ayant reçu le vote maximal : si $Sim(\mathcal{P}[i], x_k) = m$ alors $\mathcal{P}[i]$ reçoit 2^m-1 votes
 - 16: FinPour
 - 17: FinPour
 - 18: Retourner *extension*($\mathcal{P}[\text{max}]$)
-

3.6 Résultats Expérimentaux

La catégorisation d'images a connu un fort essor dans la communauté de vision depuis ces dernières décennies. À l'égard de ces raisons, nous avons choisi afin de tester la robustesse

de notre système d'utiliser les mêmes protocoles d'expérimentations adoptés dans la littérature. L'évaluation de la robustesse d'un Système de Recherche d'Images SRI revient à tester les pertinences des réponses retournées suite à une interrogation de la base d'images.

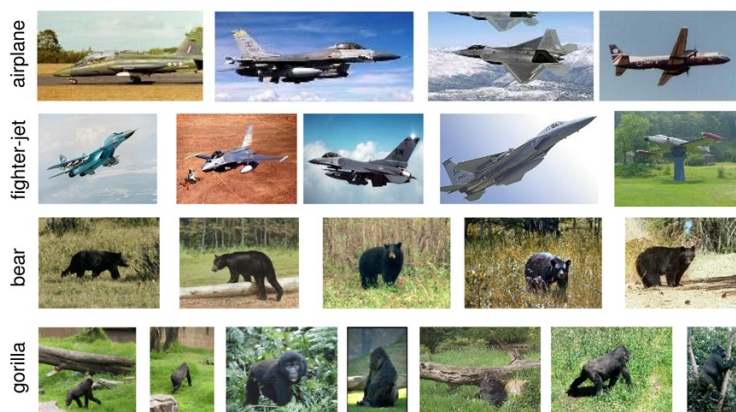


FIGURE 3.4 – Échantillons des images de la base Caltech101.

3.6.1 Comparaison avec les classifieurs de l'état de l'art

Pour la première série d'expériences, et afin de positionner notre travail dans la littérature, nous avons choisi de le comparer par rapport aux classifieurs usuels de l'état de l'art. De nombreux travaux se sont intéressés au problème de la recherche d'images par contenu en se basant sur des classifieurs très répandus dans la littérature comme par exemple les réseaux Bayésien, les arbres de décision, l'algorithme de k -ppv [Boiman *et al.*2008], les machines à vecteur de support (SVM) [Bosch *et al.*2007] et l'algorithme AdaBoost [Kawaguchi and Nishii2007]. Pour la deuxième série d'expériences, nous avons choisi d'étendre l'évaluation du système pour d'autres bases d'images à savoir Caltech101, Caltech256 et PASCAL VOC 2007² qui sont des jeux de données de référence dans le contexte de la classification d'images. Quelques échantillons de ces bases d'images sont présentés dans Figure.3.4. Pour une évaluation objective, nous avons eu recours au même protocole d'expérimentations retenu par les concepteurs de ces bases.

Nous procédons les expérimentations en utilisant les 102 classes de la base Caltech101 et les 256 classes de la base Caltech256. Le nombre d'images dans l'ensemble d'apprentissage est respectivement 5, 10, 15, 20, 25 et 30 images par catégorie et jusqu'à 50 images tests. Pour Caltech 256, 25 images de test sont utilisées. Notre classificateur ne nécessite aucun paramétrage. Un treillis est construit pour chaque catégorie d'images pour réduire le nombre de concepts. Le vocabulaire visuel utilisé est fixé à 100 mots visuels.

Caltech 101 La Figure.3.5 donne les résultats obtenus pour la recherche d'images lors de l'interrogation de la base Caltech101¹.

Nous constatons que notre algorithme est le deuxième, derrière SVM en termes de perfor-

2. <http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/>

1. La base Caltech101 contient 101 catégories d'images. Chaque catégorie contient environ 40 à 800 images. La plupart des catégories ont environ 50 images. La taille de chaque image est d'environ 300 x 200 pixels.

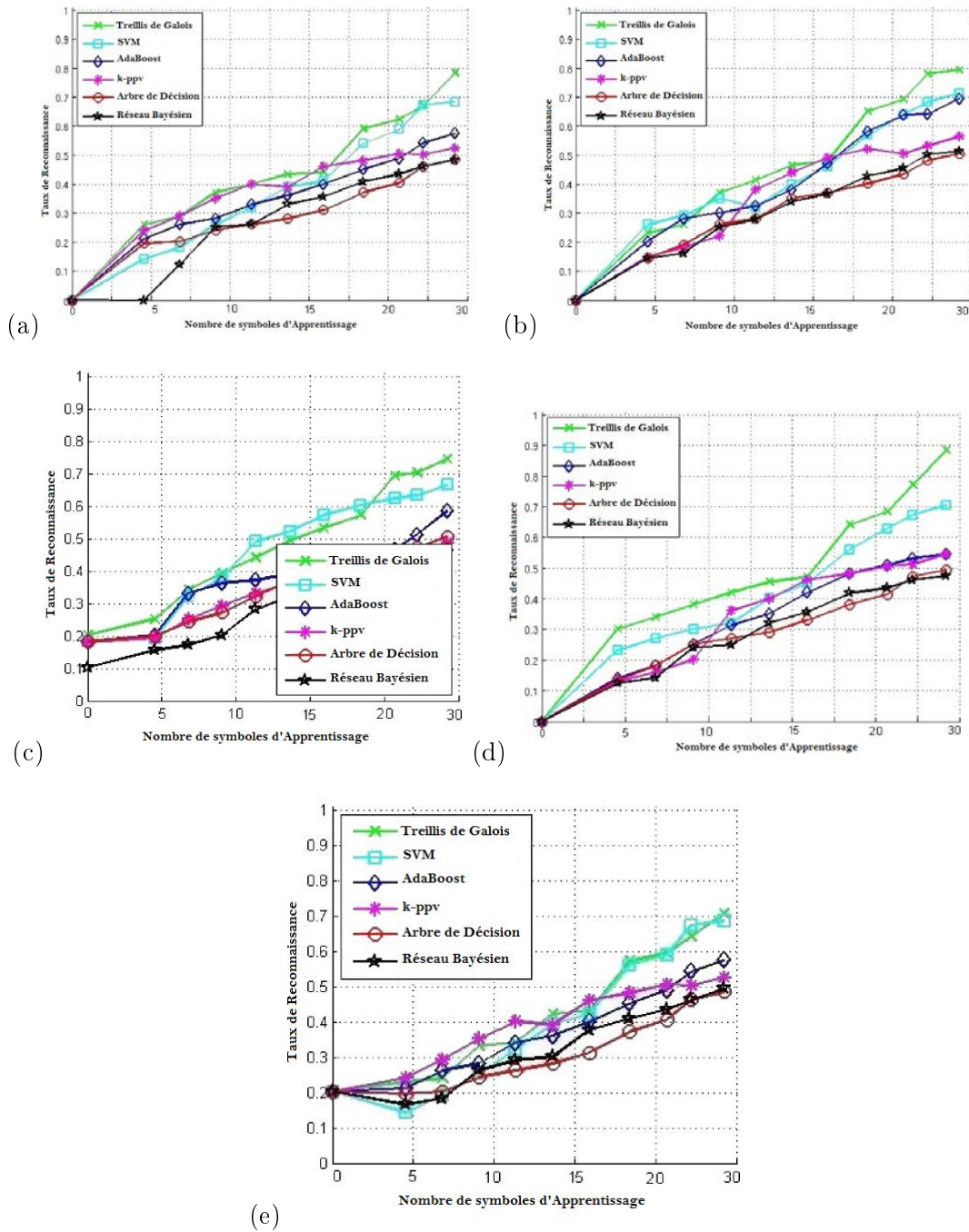


FIGURE 3.5 – Le taux de reconnaissance des classifieurs usuels en comparaison avec le treillis de Galois, en fonction du nombre de symboles de l'ensemble d'apprentissage. Le nombre d'images d'apprentissage varie entre 0 et 30 images. Les courbes montrent les performances des classifieurs en utilisant un seul descripteur (le descripteur SIFT est utilisé dans cette expérimentation) dans le cadre de la classification d'images provenant des bases Caltech101 : (a) Aeroplane, (b) Person, (c) Cars, (d) Boat, (e) Birds. Notre approche n'a pas besoin d'étape préalable de configuration de paramètres à l'encontre du SVM et le réseau Bayésien. En ce qui concerne le classifieur SVM, nous avons utilisé le noyau linéaire obtenu comme suit : $\mathcal{K}(x, x_i) = x \cdot x_i$. Pour le k -ppv nous avons choisi $k=4$.

mances pour les expériences employant 15 images d'apprentissage. Cela peut s'expliquer par le pouvoir discriminant des attributs hiérarchisés dans la structure du treillis ce qui renforce le regroupement des images partageant les mêmes attributs dans les mêmes concepts et permet de réduire la confusion entre les classes au cours de l'étape de classification. On peut observer que nous avons obtenu de bonnes performances par rapport aux classifieurs usuels. En comparaison à ces approches en particulier pour les 30 exemples d'apprentissage, nous notons que concrètement il y a une augmentation significative des performances pour de nombreuses catégories en raison de la capacité du treillis de Galois à filtrer les propriétés les plus fréquentes partagées entre les images et de les regrouper dans un même nœud facilitant ainsi la reconnaissance.

À partir des résultats obtenus, nous pouvons tirer plusieurs conclusions. Dans de nombreux cas, le treillis de Galois_{image} surpasse les méthodes classiques en terme du taux de reconnaissance de nombreuses catégories d'images. Nous notons que le classifieur SVM (noyau linéaire) montre de bonnes performances dans certains cas, par rapport à notre approche (voir Figure.3.5.(c) et (e)) ce qui pourra être expliqué par l'absence d'un nombre suffisant de mots visuels représentés dans le treillis de Galois et l'insuffisance de leurs pouvoirs discriminants ce qui dégrade la représentation de l'information issue des images. Il y aura donc une forte probabilité de regroupement des images ne partageant pas les mêmes attributs dans le même concept du treillis vu l'absence d'une description fine des images. Notons également que l'algorithme AdaBoost et le réseau Bayésien présentent de bonnes performances et qui s'avèrent meilleures que la méthode de *k-ppv* et l'arbre de décision. Pour conclure cette expérimentation, nous constatons que le treillis de concepts présente des résultats prometteurs en comparaison avec les autres classifieurs de l'état de l'art et qui sont utilisés depuis des décennies ayant souvent prouvé leurs robustesses dans le contexte de la recherche d'images. Donc ces performances peuvent ouvrir une nouvelle voie pour mieux explorer le treillis et l'adapter au contexte de la reconnaissance d'images. En effet, si le treillis est bien intégré dans la chaîne de traitement, il assure de hautes performances de classification pouvant parfois dépasser les classifieurs très largement répandus. En outre, ses performances dépendent essentiellement du nombre d'attributs caractérisant les images. L'originalité de notre approche réside dans l'absence d'une étape de configuration de paramètres comme c'est le cas pour [Bosch *et al.*2007] ou [Boiman *et al.*2008] nécessitant un paramétrage précédant la phase de classification. Cette étape a certainement un impact sur les taux de reconnaissance retournés par les classifieurs.

Caltech 256³ Plus précisément, dans Caltech-256, les images sont acquises dans des conditions d'éclairage difficiles, à différentes échelles, et à partir de divers points de vue. Pour des raisons de défi à la classification, Caltech-256 contient les mêmes catégories et des sous-catégories identiques qui peut expliquer la baisse des performances globales de toutes les approches en comparaison à Caltech101. Nos résultats ne sont pas meilleurs mais ils sont cependant similaires à ceux obtenus par les autres méthodes. Nous présentons les résultats sur 4 ensembles d'apprentissage (5, 10, 15, 25 et 30 images) dans la Figure.3.6. Notons que nous comparons notre approche par rapport aux classifieurs usuels de la littérature. Les résultats obtenus montrent que les taux de classification obtenus par le SVM et le réseau Bayésien sont meilleurs que notre approche dans le cas de 5 et 10 images d'apprentissage. Les autres approches semblent aussi performantes et présentent des taux

3. Caltech256 se compose de 256 catégories d'images. Chaque catégorie possède un minimum de 80 images à l'encontre de Caltech 101 où certaines classes regroupent moins de 31 images.

de reconnaissance assez proches. Dans le cas de 30 images d'apprentissage, les résultats obtenus mettent en évidence la robustesse du dispositif mis en place dans le cadre de la classification l'emmenant parfois à dépasser les approches de l'état de l'art.

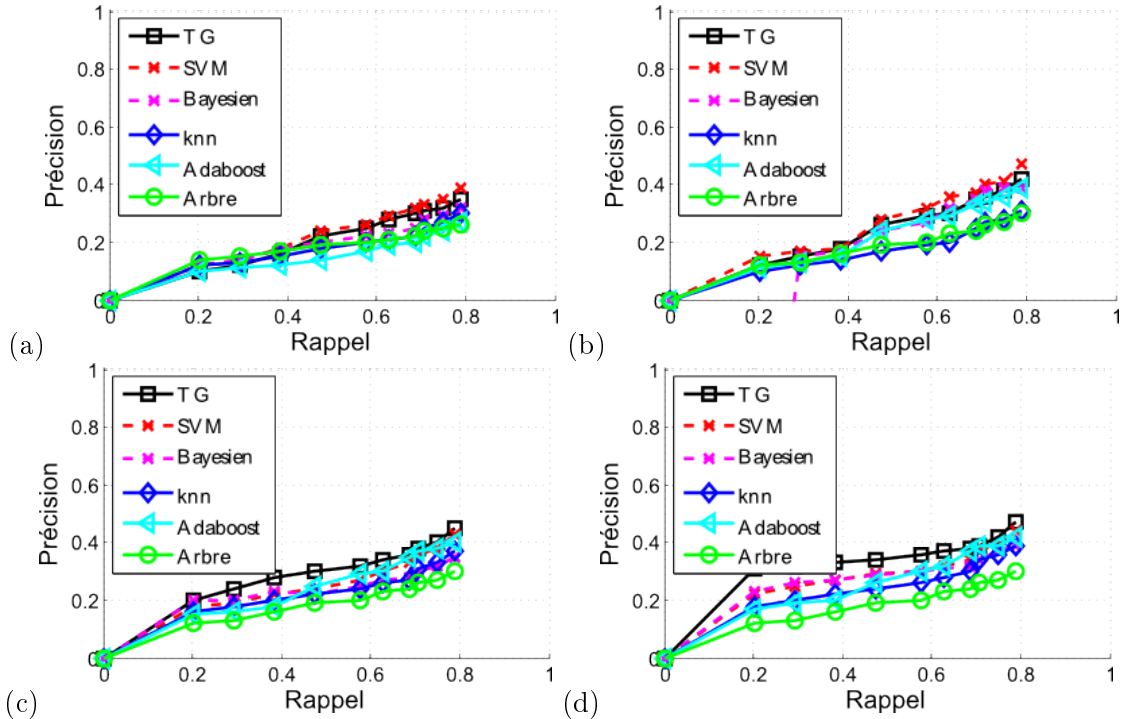


FIGURE 3.6 – Le taux de reconnaissance des classifieurs usuels en comparaison avec le treillis de Galois, en fonction du nombre de symboles de l'ensemble d'apprentissage. Le nombre d'images d'apprentissage varie entre 0 et 30 images. Les courbes montrent les performances des classifieurs en utilisant un seul descripteur (le descripteur SIFT est utilisé dans cette expérimentation) dans le cadre de la classification d'images provenant des bases Caltech256 : (a) $|\text{Échantillon}_{\text{apprentissage}}|=5$, (b) $|\text{Échantillon}_{\text{apprentissage}}|=10$, (c) $|\text{Échantillon}_{\text{apprentissage}}|=20$ et (d) $|\text{Échantillon}_{\text{apprentissage}}|=30$. Notre approche n'a pas besoin d'étape préalable de configuration de paramètres à l'encontre du SVM et le réseau Bayésien. En ce qui concerne le classifieur SVM, nous avons utilisé le noyau linéaire obtenu comme suit : $\mathcal{K}(x, x_i)=x^*x_i$. Pour le k -ppv nous avons choisi $k=4$.

Pascal VOC 2007 Dans cette section, nous comparons les performances de notre méthode avec quelques méthodes de la littérature. Nous construisons le treillis de Galois sur l'ensemble d'apprentissage. Le Tableau.3.2 compare les résultats expérimentaux de notre approche avec les meilleures performances en PASCAL VOC2007. Comme nous pouvons le voir, notre méthode présente de bons taux de classification sur la plupart des catégories d'objets. On peut constater que nous avons obtenu de bonnes performances sur la plupart des catégories, sauf pour les catégories "Dog" et "Bottle". Ces catégories sont difficiles à reconnaître en raison du faible nombre d'images en comparaison avec les autres catégories. Les résultats s'avèrent très prometteurs pour la catégorisation des images et permettent d'atteindre un bon taux de classification, par exemple "Person"(88,7%), "Cars"(86,2%) ou "Train"(84,7%), ce qui peut s'expliquer par le fait que ces catégories sont représentées par un nombre suffisant d'exemples (plus de 400 images en nombre moyen par catégorie), tan-

dis que certaines catégories (*Dog* par exemple) ont moins de 50 images et sont plus difficiles à reconnaître en raison du manque d'une description fine de ces images. Dans l'ensemble, les meilleurs résultats sont obtenus quand une catégorie a un nombre suffisant d'exemples et pour les catégories dont les caractéristiques résultant de l'étape de traitement sont assez discriminantes. Les résultats obtenus soulignent le succès du treillis de Galois associé au dictionnaire visuel dans le contexte de la classification d'images.

Categories	[Zhou et al.2010]	[Marszalek and Schmid2007]	[Galleguillos et al.2008]	[Wang et al.2009]	[Khan et al.2009]	[Yang et al.2009]	Notre approche
aeroplane	79.4%	77.5%	63.0%	65.0%	65.0%	79.4%	74.4%
bicycle	72.5%	63.6%	22.0%	44.3%	48.0%	62.4%	73.2%
bird	55.6%	56.1%	14.0%	48.6%	44.0%	58.5%	47.1%
boat	73.8%	71.9%	42.0%	58.4%	60.0%	70.2%	75.1%
bottle	34.0%	33.1%	43.0%	17.8%	20.0%	46.6%	30.7%
bus	72.4%	60.6%	50.0%	46.4%	49.0%	62.3%	66.8%
car	83.4%	78.0%	62.0%	63.2%	70.0%	75.6%	86.2%
cat	63.6%	58.8%	32.0%	46.8%	49.0%	54.9%	62.8%
chair	56.6%	53.5%	37.0%	42.2%	50.0%	63.8%	60.1%
moto	71.9%	64.0%	31.0%	50.3%	59.0%	68.1%	70.2%
person	85.1%	85.9%	43.0%	78.1%	81.0%	87.1%	88.7%
dog	49.5%	45.8%	29.0%	37.7%	40.0%	51.6%	39.1%
table	59.8%	50.6%	37.0%	26.8%	42.0%	56.4%	60.3%
train	83.3%	79.2%	29.0%	62.8%	68.0%	75.9%	84.7%
TV	58.9%	53.2%	62.0%	33.3%	49.0%	54.4%	62.1%
table	63.2%	54.9%	30.0%	20.8%	39.0%	58.3%	68.7%
plant	36.4%	36.3%	33.0%	27.2%	32.0%	49.5%	57.5%

TABLE 3.2 – Taux de reconnaissance du treillis de concepts et d'autres méthodes dans le cadre de la classification de quelques catégories de la base Pascal VOC 2007.

3.7 Évaluation des performances

Dans cette section, nous évaluons la performance du treillis de Galois_{image} utilisé dans le cadre de la catégorisation des images en fonction du nombre d'attributs (la taille du vocabulaire dans notre cas) et du nombre de données d'apprentissage.

3.7.1 Évaluation des performances du treillis en fonction du nombre d'attributs

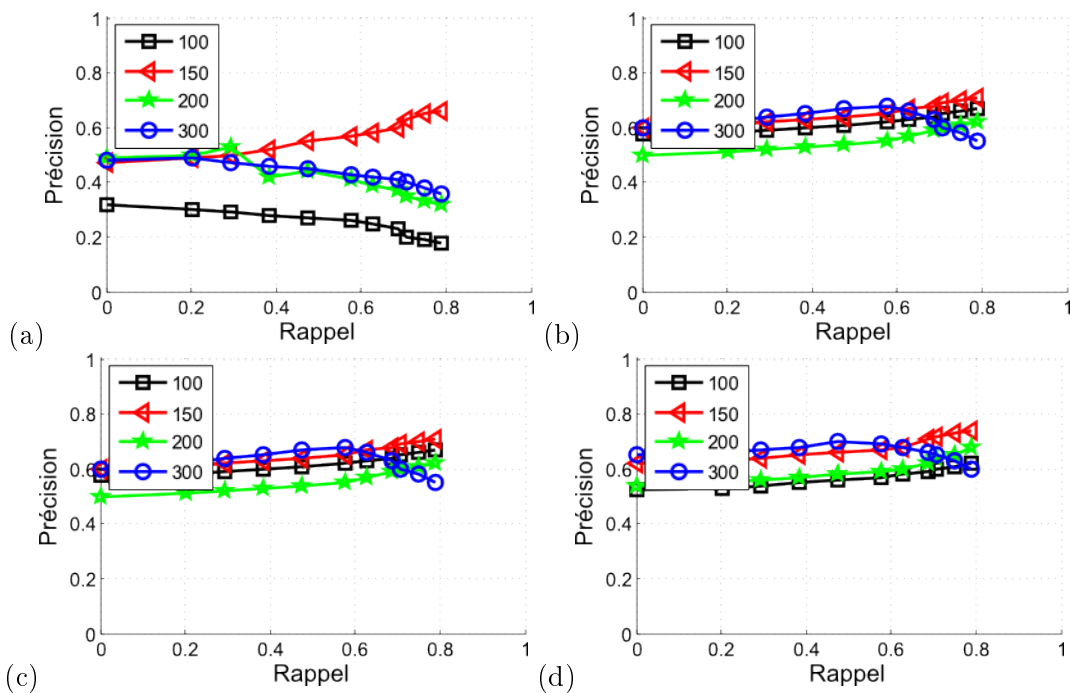


FIGURE 3.7 – Évaluation du taux de reconnaissance du treillis de concepts en fonction de la taille du vocabulaire visuel. Le nombre d'images d'apprentissage varie entre 0 et 30 images. Les courbes montrent les performances des classificateurs en utilisant un seul descripteur (le descripteur SIFT est utilisé dans cette expérimentation) pour la classification d'images provenant des bases Caltech 256 : (a) Avion, (b) Personne, (c) Voiture, (d) Bateau.

Le treillis de Galois_{image} que nous proposons est basé sur la modélisation des relations entre les images et leurs propriétés. Dans l'expérience suivante, nous allons varier le nombre de mots visuels utilisés pour construire notre classificateur. Nous avons choisi un nombre compris entre 100 et 300 mots visuels. Nos expériences présentées dans la Figure.3.7 vise à évaluer l'impact de la taille du dictionnaire visuel sur le taux de classification obtenu par le treillis de Galois_{image} . Dans cette expérience, nous cherchons à minimiser le taux d'erreur de classification sur un ensemble d'images test. Cela signifie que nous cherchons à minimiser le biais entre les taux de *précision/rappel*.

Nous avons choisi d'évaluer les catégories indiquées dans la Figure.3.7 parce qu'elles montrent de bons taux de reconnaissance au cours de notre processus ce qui nous aide à trouver des critères stables ou des paramètres qui seront utilisés lors des expériences. On remarque que le résultat obtenu à l'aide du treillis de Galois_{150} est le plus stable au cours de toutes les expériences en comparaison aux autres tailles du vocabulaire. Cela peut s'expliquer par

le nombre suffisant de descriptions discriminantes pour chaque image indépendamment de la catégorie d'images.

3.7.2 Évaluation du taux de reconnaissance en fonction de la taille de l'échantillon d'apprentissage

Notre seconde expérience vise à évaluer l'influence de la taille de l'ensemble d'apprentissage sur le taux de reconnaissance du classifieur. La Figure.3.8 présente les résultats de l'étape de classification pour quelques catégories de la base Caltech101. Ces expérimentations permettent de mettre en évidence la variation du taux de reconnaissance du treillis en fonction du nombre d'images d'apprentissage.

Nous constatons que le taux de reconnaissance augmente lorsque le nombre d'images d'apprentissage augmente jusqu'à atteindre un certain seuil de 50 images où les résultats commencent à prendre une autre tendance. À partir de ce seuil, nous pouvons voir que le taux de classification diminue et stagne ensuite, ce qui peut s'expliquer par le phénomène de sur-apprentissage. Nous concluons que lors de l'utilisation d'un certain nombre d'images d'apprentissage compris entre 15 et 50 images, nous obtenons les meilleurs résultats pour les taux de reconnaissance. Le treillis de Galois est sensible à la taille de l'ensemble d'apprentissage et lorsque ce nombre atteint un certain seuil, il sera plus difficile de différencier entre les images de différentes catégories.

En guise de conclusion, nous pourrions affirmer que le treillis de Galois_{image} dépend du nombre des images de l'ensemble d'apprentissage et pour pallier au problème de sur-apprentissage, il nous est paru nécessaire de procéder à un élagage du treillis.

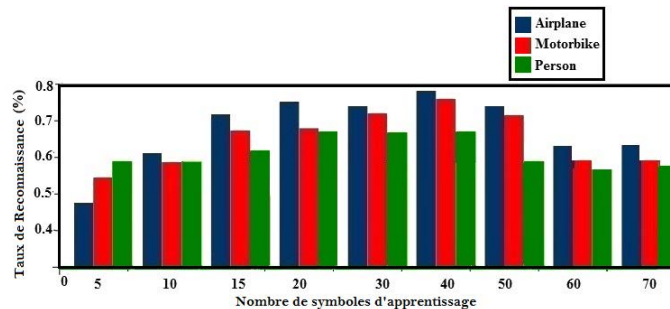


FIGURE 3.8 – Évaluation du taux de reconnaissance retourné par le treillis de Galois en fonction de la taille de l'ensemble d'apprentissage. Le nombre d'images test est 15 images. La taille du vocabulaire utilisée dans cette expérimentation $|\Omega|=150$.

3.8 Étude de la Complexité du treillis

Dans cette section, nous évaluons la complexité algorithmique du classifieur utilisé en terme du nombre de concepts générés ainsi qu'en terme du temps d'exécution nécessaire pour l'étape de reconnaissance. Les résultats reportés dans la Table.3.3 proviennent des expérimentations qui ont été effectuées en interrogeant les bases Caltech 101, Caltech 256 et Pascal VOC2007. La taille de l'ensemble d'apprentissage varie entre 5 et 30 images pour chaque ensemble et l'ensemble de test varie entre 20 et 50 images. Dans la Table.3.3, nous avons présenté les statistiques obtenus par notre structure en vue de tester sa robustesse dans le contexte de la catégorisation des images. Le temps_{Classification} englobe le temps de

recherche de concepts partageant les mêmes attributs qu'une image requête ainsi que les mis à jour qui pourront avoir lieu dans le treillis de concepts durant la phase de reconnaissance. Le nombre de concepts augmente avec l'augmentation de l'ensemble d'apprentissage mais reste modéré ce qui s'explique par l'efficacité de l'algorithme de construction utilisé ainsi que l'heuristique de génération des concepts nécessaires à la classification que nous avons utilisée. Un constat global que nous pouvons faire est que l'algorithme que nous avons utilisé pour la construction ainsi que l'ensemble des heuristiques adoptées s'avèrent très performantes et ont permis de réduire le coût en temps et en espace de la structure du treillis proposée. De plus, le temps de classification est nettement réduit et reste souvent modéré.

	NbSymb _{Apprentissage}	NbConcepts	Temps _{Classification} (s)
Caltech 101	5	48	2.1
	10	32	1.9
	15	89	3.1
	30	153	2.3
Caltech 256	5	62	2.6
	10	60	1.9
	15	104	2.7
	30	201	4.7
Pascal VOC 2007	5	34	2
	10	55	1.7
	15	77	1.2
	30	108	3.6

TABLE 3.3 – Évaluation des performances du treillis de Galois en interrogeant les bases : Caltech101, Caltech256 et Pascal VOC2007. Nous évaluons les performances du treillis de concepts en terme du nombre d'images d'apprentissage $\mathbf{NbSymb}_{Apprentissage}$. Nous mesurons également le nombre de concepts du treillis $\mathbf{Nb}_{concepts}$ et le temps $\mathbf{Temps}_{Classification}$ nécessaire pour la recherche des instances d'images similaires à la requête. La taille du vocabulaire utilisé dans cette expérimentation est $|\Omega|=50$.

3.9 Conclusion

Dans ce chapitre, nous avons présenté une application du treillis de concepts dans le cadre d'un système de classification d'images afin de tester sa robustesse. Notre travail fournit un cadre original et intéressant d'application du treillis de Galois qui ouvre la voie à mieux explorer cette structure dans divers domaines tout en intégrant les heuristiques capables de remédier à la complexité algorithmique de ce formalisme dans le pire des cas.

Notre contribution vise à tirer profit de la capacité d'un tel classifieur pour reconnaître une banque d'images. En outre, étant donné que la partie la plus coûteuse de l'approche se fait hors ligne, le système proposé est adapté pour les requêtes en ligne et les résultats obtenus en comparaison avec l'état de l'art sont compétitifs.

Une comparaison avec les classifieurs usuels de la littérature a été proposée. Sur ce point, nous pouvons dire que le treillis s'est avéré assez robuste et a fourni des résultats prometteurs et semblables aux résultats des autres approches. Le treillis de Galois présente de nombreux avantages : il présente un espace de recherche exhaustif et concis, en plus il permet de hiérarchiser un grand nombre de données tout en gardant l'aspect relationnel et sémantique entre $\{Objets/Propriétés\}$ ce qui permet de pallier au problème du biais sémantique de certains classifieurs.

Nous ne prétendons pas dans ce travail présenter un classifieur très robuste, pas plus que nous démontrons que le treillis de Galois s'il est bien intégré dans une architecture adaptée, il permet de fournir de bonnes performances. De plus, les règles d'association qui peuvent être extraites à partir de l'arborescence du treillis représentent une source d'information très fine permettant de traiter même la fouille de données en extrayant des motifs apparaissant dans des documents graphiques comme les logos par exemple. Cela représente un axe de recherche très prometteur et intéressant. Dans les chapitres suivants, nous allons présenter quelques travaux qui s'inscrivent dans le cadre de traitement des documents graphiques et plus précisément nous nous intéressons à la problématique de la localisation de symboles graphiques dans les dessins techniques.

Troisième partie

Localisation de symboles dans les documents graphiques

État de l'Art sur les Approches de Localisation de Symboles dans les Documents

"On peut déterminer, non seulement des probabilités à partir des observations issues d'une expérience, mais aussi les paramètres relatifs à ces probabilités."

Thomas Bayes

Sommaire

1.1	Introduction	139
1.2	Localisation de Symboles dans les Documents Graphiques	140
1.2.1	Approches Structurelles	141
1.2.2	Approches Pixelaires	148
1.2.3	Discussion	151
1.3	Conclusion	152

1.1 Introduction

Dans ce chapitre, nous abordons une autre thématique qui intéresse la communauté autant que la reconnaissance de symboles graphiques. Il s'agit de la localisation des symboles dans les documents graphiques (voir Figure.1.1). Ici, les symboles ne sont plus isolés de leur contexte comme les symboles étudiés dans les chapitres précédents. L'intérêt porté sur ce type d'approches revient au fait que, durant les dernières décennies, les contributions proposées dans la littérature tournaient autour du problème de la reconnaissance de symboles et laissaient de côté le problème de détection de symboles graphiques en contexte. En outre, de nombreux domaines ont montré un besoin croissant d'indexation automatique de documents et la récupération de l'information dans de grandes collections de données. Certains de ces domaines portent sur des bases de documents très volumineuses, par exemple les bases de livres dans les musées ou les bibliothèques, les catalogues de produits ou services dans le domaine du web ou encore les plans architecturaux dans le domaine de l'architecture. Un tel contexte est une problématique plus complexe que celle de la reconnaissance de



FIGURE 1.1 – Quelques exemples de documents graphiques où les symboles sont enfouis dans le document.

symboles isolés puisqu'elle a besoin en plus de la phase de reconnaissance, d'une phase de localisation. D'une manière générale, l'architecture d'un système de localisation de symboles se compose des mêmes étapes que les systèmes de reconnaissance décrits dans le chapitre précédent auxquelles s'ajoute une étape de localisation qui permet de détecter les régions du document graphique susceptibles de contenir les occurrences du symbole requête. Dans une première étape, les documents sont décomposés en un ensemble de composantes puis un ensemble de descripteurs de formes prédéfinis sont appliqués sur chacune de ces composantes. Les approches de localisation de symboles dans les documents graphiques ne sont pas trop abordées dans la littérature malgré les nombreux travaux présentés dans le cadre de l'analyse des documents graphiques.

Ce chapitre sera consacré à l'étude des approches de localisation de symboles tout en présentant un état de l'art sur les travaux qui ont été réalisés dans les dernières années dans ce contexte. Tout d'abord, nous présentons un panorama de quelques techniques utilisées dans la littérature et les contributions de la communauté dans le contexte de la résolution du problème de localisation de symboles dans les documents graphiques et décrivons ensuite nos techniques proposées dans le cadre de cette thèse et qui présentent un effort continu qui s'inscrit dans le cadre des systèmes de détection d'objets graphiques dans leur contexte.

1.2 Localisation de Symboles dans les Documents Graphiques

La plupart des approches de localisation de symboles nécessitent une étape préalable de segmentation afin d'assurer un taux de reconnaissance assez élevé pour le processus de localisation d'un symbole requête dans une base de documents graphiques. En se basant sur ce critère, dans la littérature il existe deux catégories d'approches de localisation : les approches structurales et les approches pixellaires. Les approches structurales sont basées souvent sur les graphes et nécessitent généralement une étape de segmentation. Une étape de vectorisation s'avère souvent nécessaire [Locteau *et al.*2007,

Wenyin *et al.*2007, Rusiñol and Lladós2007]. Certains travaux [Tabbone and Zuwala2007, Rusiñol and Lladós2007, Rusiñol and Lladós2005] proposent de ne considérer que les symboles satisfaisant certaines conditions comme par exemple la convexité, la connectivité ou la fermeture du symbole. L'image est donc décomposée en primitives comme des lignes, des régions, des courbes ou encore des formes géométriques comme les cercles, les rectangles... Les relations entre primitives sont dans la plupart des cas représentées par des graphes et le processus de localisation d'un symbole requête revient à la recherche d'un isomorphisme de (sous-) graphes. Contrairement aux approches pixelaires, où l'étape de localisation est effectuée directement sur les documents sans étape préalable de segmentation lourde. Dans ce qui suit, nous évoquons quelques approches qui nous paraissent les plus communes ou les plus adaptées au contexte de nos travaux.

1.2.1 Approches Structurelles

Les systèmes de localisation sont destinés à assurer à la fois la reconnaissance et la segmentation, par la suite ils ont besoin d'une étape préalable de segmentation des documents afin d'en extraire des primitives. Ensuite, pendant la phase de localisation, le système est invité à produire une liste des régions d'intérêt susceptibles de contenir les occurrences du symbole interrogé. Dans ce qui suit, nous allons présenter quelques approches de localisation existantes dans la littérature. Muller and Rigoll [Müller and Rigoll2000] ont proposé une

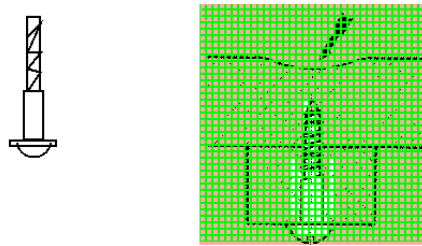


FIGURE 1.2 – Symbole requête et la segmentation correspondante du document graphique (extrait de [Müller and Rigoll2000]).

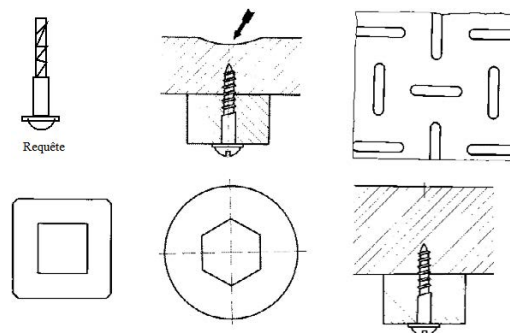


FIGURE 1.3 – Symbole requête et les résultats du processus de localisation (extrait de [Müller and Rigoll2000]).

parmi les premières approches dans la littérature. Les dessins techniques sont segmentés en utilisant une grille de taille fixe comme c'est montré dans la Figure.1.2. Chaque petite cellule agit alors comme une entrée dans une chaîne de markov HMM à deux dimensions

apprise déjà afin d'identifier les endroits du dessin technique où un symbole requête est susceptible d'être trouvé. Le principal avantage de leur système est que les symboles interrogés peuvent être repérés, même s'ils apparaissent dans un environnement encombré (voir Figure.1.3). Cependant, l'étape d'apprentissage nécessaire pour apprendre le HMM utilisé, entraîne une perte de flexibilité de la méthode présentée.

Afin de localiser les zones de l'image qui pourrait potentiellement contenir le symbole demandé, [Dosch and Lladós2003] propose d'utiliser la signature vectorielle afin de représenter le symbole interrogé ainsi que les régions dans l'image. Le symbole est défini par un ensemble de relations entre les segments. Cinq types de relations sont considérées : le recouvrement, la colinéarité, le parallélisme, la jonction "T" et la jonction "V" (voir Figure.1.4). Le processus de localisation consiste à découper l'image en plusieurs régions

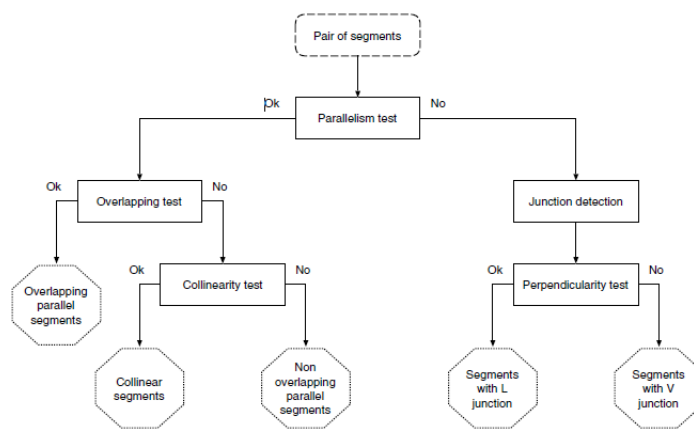


FIGURE 1.4 – Les différents types de relations existantes entre deux segments pour la construction de la signature vectorielle : la colinéarité (C), le recouvrement (R), le parallélisme (P), la jonction "T" (T) et la jonction "V" (V) (extrait de [Dosch and Lladós2003]).

disjointes (appelées "bucket") dont la taille dépend de la taille du plus grand symbole de référence. Un test d'inclusion des signatures entre le symbole requête et ces "buckets" est effectué afin de trouver l'ensemble de régions du document contenant les occurrences du symbole cherché. Cette approche qui est basée sur une étape de pré-traitement ne détermine pas l'emplacement exact des symboles, mais permet de détecter rapidement le type de symbole contenu dans chaque région. Les résultats fournis par cette méthode permettent de déduire que l'approche est robuste au niveau de la détection du type du symbole mais elle reste limitée au niveau de la détermination des localisations exactes du symbole (voir Figure.1.5). Dans [Zuwala and Tabbone2006, Tabbone and Zuwala2007], les auteurs proposent un système de localisation basé sur une méthode de description structurelle fournissant les régions du document pouvant contenir un symbole requête. À partir d'un découpage du document graphique en chaînes de points clés connectés, ils proposent de fusionner successivement les régions entre elles en fonction d'un critère de densité, ce qui permettra ensuite de redéfinir de nouveau les symboles et effectuer une reconnaissance à partir du symbole requête grâce à l'utilisation de la structure du dendrogramme (voir Figure.1.6). Après la phase de décomposition du document en chaînes de points connectés (voir Figure.1.7(c)), un graphe de jonctions est construit dont les nœuds correspondent aux chaînes de points et les arcs représentent la connexion entre deux chaînes de points (voir

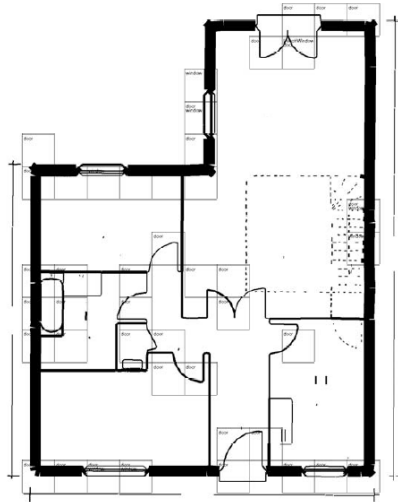


FIGURE 1.5 – Quelques résultats de localisation de symbole dans des dessins techniques obtenus par la méthode du calcul des signatures vectorielles. Les régions d'intérêt susceptibles de contenir le symbole requête sont présentées par un rectangle contenant le nom du symbole(s) détecté (extrait de [Dosch and Lladós2003]).

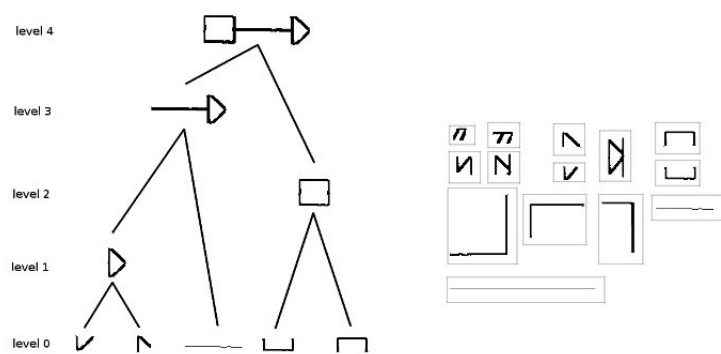


FIGURE 1.6 – (a)Un dendrogramme, (b)Les partitions trouvées (extrait de [Zuwala and Tabbone2006]).

Figure.1.8). Le processus de localisation consiste à trouver les sous-ensembles de chaînes les plus proches parmi ceux construits par les dendrogrammes. Les résultats s'avèrent pro-

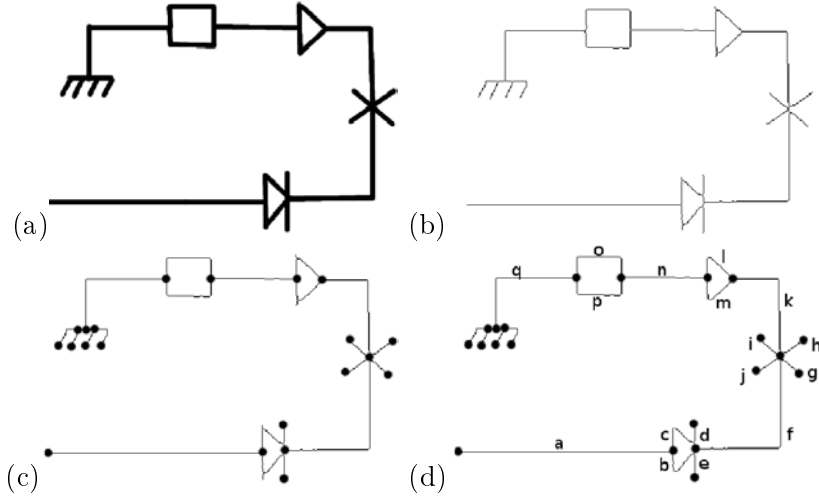


FIGURE 1.7 – Pré-traitements du document (a) 1^{ère} étape : binarisation du document, (b) 2^{ème} étape : squelettisation du document, (c) 3^{ème} étape : localisation des points de jonctions, (d) 4^{ème} étape : décomposition en chaînes de points (extrait de [Zuwala and Tabbone2006]).

metteurs sauf qu'un ensemble de contraintes ont été introduites par les auteurs comme par exemple le fait que les symboles sont considérés comme un ensemble de chaînes de points compactes et plutôt convexes. Malgré que les auteurs aient essayé d'améliorer les résultats en utilisant un descripteur photométrique, le système ne peut pas détecter les symboles dont les chaînes ne sont pas connectées, les symboles qui ne possèdent pas de points de jonction dans le document et aussi les symboles partageant une même chaîne de points (voir Figure.1.9). D'autres techniques, comme dans [Barbu *et al.*2005], [Lladós *et al.*2001],

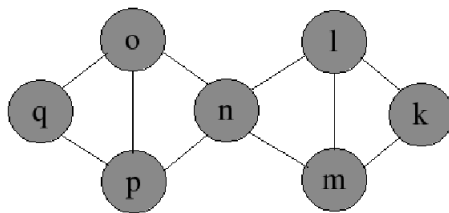


FIGURE 1.8 – Une partie du graphe de jonction définie sur la Figure.1.7 (extrait de [Zuwala and Tabbone2006]).

[Locteau *et al.*2007], [Melo *et al.*2011a], [Qureshi *et al.*2008] s'appuient sur une représentation du document graphique basée sur les graphes. Ces méthodes sont axées sur une description structurelle des symboles graphiques. L'isomorphisme des (sous-) graphes est alors utilisé afin de localiser et reconnaître les symboles requêtes dans un document graphique. Cependant, ces approches ne sont pas adaptées à de grandes collections de documents vu que le processus d'appariement de graphes est coûteux.

[Liu *et al.*2009] ont utilisé des lignes et des courbes comme primitives afin de caractériser des images. Pour chaque primitive, la structure locale est définie par les voisins les plus

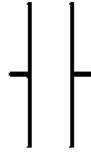


FIGURE 1.9 – Exemple de symboles non traités par la méthode proposée par [Zuwala and Tabbone2006].

proches. Elle est caractérisée par les distances et les angles relatifs entre cette primitive et les primitives dans le voisinage. La détection de symboles est réalisée par un processus de vérification d'hypothèses. Tout d'abord, les premiers k plus proches primitives locales dans le document sont obtenues pour chaque structure locale du symbole modèle, et les paramètres de transformation correspondant sont estimés. En considérant chaque résultat de l'estimation comme un point dans l'espace des paramètres, une région dense dans cet espace indique qu'il existe une occurrence du modèle dans l'image. Enfin, une étape de vérification des régions déjà détectées pour localiser les occurrences du modèle dans le document, est mise en place. [Rusiñol and Lladós2005] propose un système de localisation

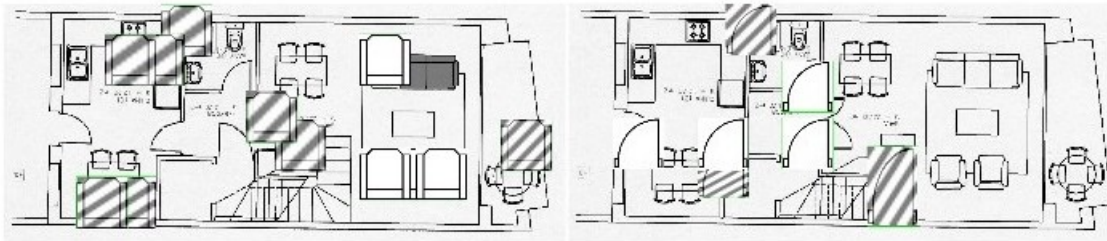


FIGURE 1.10 – Quelques exemples de résultats de localisation obtenus par la méthode de [Rusiñol and Lladós2005].

de symboles (voir Figure.1.10) où des signatures sont formulées en terme de contraintes géométriques et structurelles entre les segments, comme les parallélismes, les angles droits, ... Après avoir représenté des dessins de trait vectorisés avec des graphes attribués, l'approche fonctionne avec une représentation multi-échelle de ces graphiques. Un processus de vote est ensuite appliqué afin de produire les régions d'intérêt retenues à partir des documents où le symbole requête est susceptible d'être trouvé. Afin d'améliorer cette approche, [Rusiñol and Lladós2007] propose de localiser des régions dans le document sans étape préalable de segmentation. Une analyse en composantes connexes est effectuée sur un document afin de fournir les régions fermées qui seront par la suite représentées par des polygones d'approximation comme montré dans la Figure.1.11. Une fonction de hachage est ensuite construite en utilisant l'ovale de Cassini qui possède deux foyers représentant les deux points les plus éloignés du polygone et dont l'intersection passe par le centre de gravité du polygone. Un polygone est représenté par un couple (a, b) où a et b sont les deux paramètres géométriques de l'ovale de Cassini qui couvre le polygone. Ces couples sont ensuite utilisés comme entrées dans la table de hachage. Le système de localisation

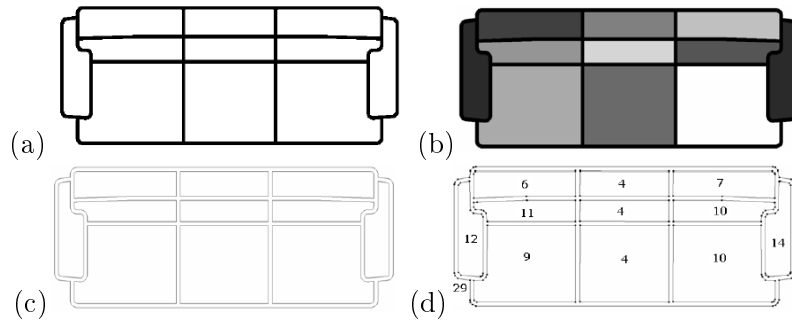


FIGURE 1.11 – Les étapes de représentation des symboles. (a) Symbole original (b) Les composantes connectées (c) Les contours de chaque région (d) Approximation polygonale des régions fermées avec le nombre de segments composant chaque polygone (extrait de [Rusiñol and Lladós2007]).

consiste à trouver les régions dans le document pouvant contenir des occurrences du symbole requête. Un processus de vote permet d'identifier les polygones dans le document qui ressemblent à ceux du symbole requête. Une limite de cette approche réside dans le fait qu'elle ne s'applique que sur les symboles possédant des contours fermés et elle favorise les faux positifs comme illustré dans la Figure.1.12.

Dans le cadre de l'évaluation des approches de localisation, il est de coutûme de se situer et

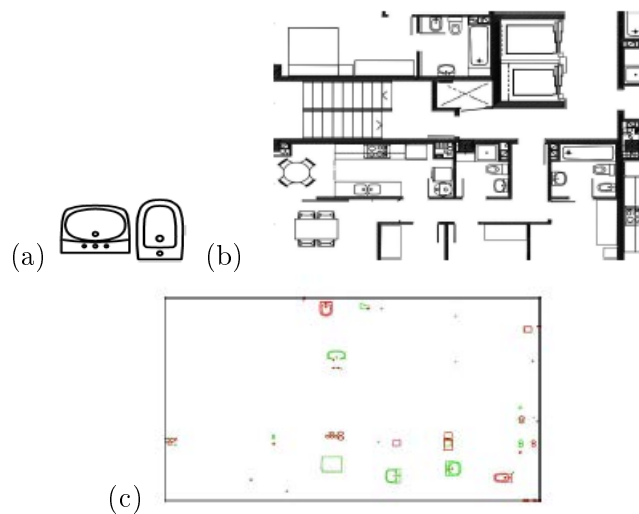


FIGURE 1.12 – Résultats du processus de localisation de symboles dans les documents graphiques [Rusiñol and Lladós2007] : (a) Symboles requêtes : un lavabot et un bidet (b) Image originale, (c) Les régions correspondantes au lavabot sont marquées en vert et les régions correspondantes au bidet sont marquées en rouge.

de se comparer aux approches existantes, pour cela et dans ce qui suit, nous allons détailler une approche qui a été proposée pour la localisation des expressions mathématiques dans un document graphique en utilisant l'arbre de décision. Nous mettons cette approche en relief car nous utiliserons une structure similaire qui est la structure du treillis de concepts afin de satisfaire le processus de localisation de symboles. [Zanibbi and Yu2011] proposent

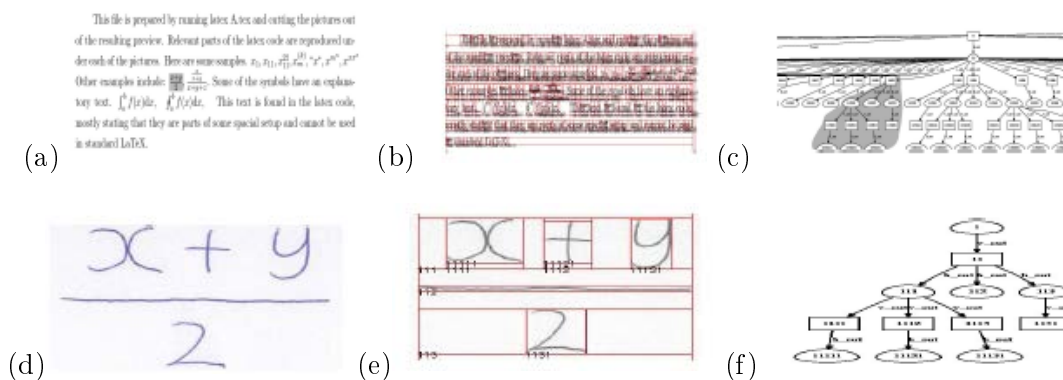


FIGURE 1.13 – (a) Un document, (b) Décomposition du document en arbre X-Y, (c) Arbre X-Y du document, (d) Image requête, (e) Décomposition en arbre X-Y, (f) Arbre X-Y de l'image requête (extrait de [Zanibbi and Yu2011]).

un système de localisation des expressions mathématiques dans des documents techniques sans l'utilisation de la reconnaissance de caractères. En se basant sur la projection des pixels, les arbres X-Y peuvent être obtenus par un découpage X-Y qui découpe la page respectivement selon les directions horizontales et verticales (voir Figure.1.13(b)). Un index des régions du document est produit à partir d'arbres X-Y définis pour chaque page dans le corpus. Les requêtes sont fournies sous forme d'images d'expressions écrites à la main, pour lesquelles un arbre du type X-Y est calculé (voir Figure.1.13). Après avoir obtenu les arbres X-Y, le problème de localisation mathématique se réduit donc à un problème d'appariement d'arbres. La Figure.1.13(f) représente un arbre X-Y pour l'image requête, et si on peut trouver un sous-arbre dans la Figure.1.13(c), similaire à l'arbre de requête X-Y, signifie que la requête est incluse dans la page. Les trois étapes décrites dans [Zanibbi and Yu2011] sont les suivantes :

- Étant donné deux arbres X-Y nommés T1 et T2, un algorithme en temps linéaire qui balaye chaque nœud une seule fois doit être mis en place afin de trouver un sous arbre commun entre T1 et T2.
- La constitution d'un appariement de T1 à T2 en se basant sur les résultats de l'étape précédente.
- Évaluer le coût de l'appariement (distance d'édition) entre T1 et T2 (par exemple sur la Figure.1.14(a) le coût de l'appariement entre les deux arbres T_a et T_b est égale à la somme de : (1) coût de suppression de a_1, a_2, a_3, a_4 de T_a , (2) coût de l'ajout de b_1, b_4 dans T_b , (3) coût de conversion (a_5 en b_2), (a_6 en b_3)..., (a_{10} en b_8). Dans la Figure.1.14(b) le coût est égale à zéro, donc il s'agit d'un appariement exacte.

Cette approche se base sur l'appariement des arbres afin de trouver les régions d'intérêt susceptibles de contenir l'image requête. Une étude expérimentale montre qu'en utilisant un algorithme linéaire pour l'appariement entre deux arbres, cette approche peut produire des résultats satisfaisants pour le processus de localisation.

Discussion

En guise de conclusion de cette section, nous pouvons affirmer que les approches structurales présentées ci-dessus peuvent localiser efficacement les symboles, même dans les cas complexes, cependant, elles nécessitent généralement quelques étapes préalables de trai-

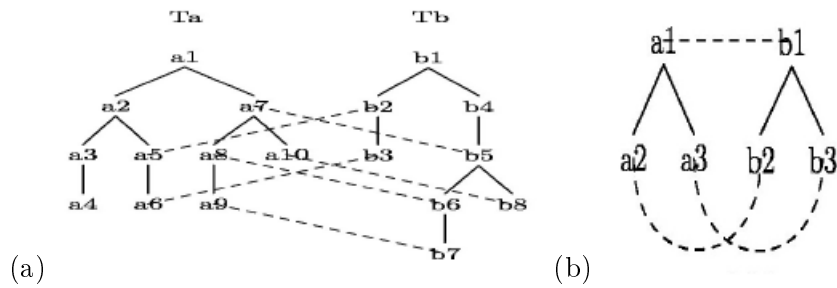


FIGURE 1.14 – (a) Appariement entre T_a et T_b , (b) Un appariement exacte (extrait de [Zanibbi and Yu2011]).

tement telles que la vectorisation et la segmentation d'images afin de les décomposer en primitives. Ces primitives seront plongées par la suite dans une structure de graphe dans la plupart des cas. La recherche d'un isomorphisme de (sous-) graphes aboutit à la localisation de l'objet dans le document. La complexité du processus de mise en correspondance entre les objets correspond à la complexité de la mise en correspondance entre les (sous-) graphes qui est un problème *NP-complet*. En outre, elles imposent des contraintes et des hypothèses sur des symboles tels que la connectivité, la convexité ou la fermeture de symbole. Dans la prochaine section, nous présentons un autre type d'approches qui ne nécessitent aucune étape préalable de segmentation avant la phase de localisation.

1.2.2 Approches Pixelaires

Afin de remédier aux problèmes des approches structurelles évoqués ci-dessus, un ensemble d'approches pixelaires ne nécessitant ni d'étape préalable de segmentation lourde ni d'hypothèses sur les symboles, ont été proposées dans la littérature. Parmi ces travaux, citons [Rusiñol and Lladós2008] qui propose un processus de vote sur toute l'image en validant l'organisation spatiale des points d'intérêt appariés. Chaque paire de points appariés est utilisée pour définir, à travers un système de vote une liste de zones du document susceptibles de contenir le symbole requête. Pour représenter l'organisation spatiale des descripteurs locaux, un graphe de proximité est construit. Chaque point d'intérêt est relié à ses k plus proches voisins par une arête du graphe [Rusiñol and Lladós2008]. Un exemple de ce graphe est illustré dans la Figure.1.15. Cette approche montre de bons résultats, cependant, les symboles possédant peu de points d'intérêt discriminants (en particulier dans le cas de connexions avec d'autres parties du document) sont difficiles à localiser. Un appariement entre chaque paire de points respectivement du modèle et du document permet de définir, au travers un système de vote, le centre hypothétique du symbole dans le document. Ce centre est déterminé de sorte que la relation spatiale existante entre ce centre et le paire de points dans le document soit similaires à celle existante entre le centre et les deux points d'intérêt appariés du modèle requête.

MacLean et Tsotsos [MacLean and Tsotsos2008] proposent une méthode de localisation sans étape préalable de pré-traitement ou de décomposition. Une représentation pyramidale du modèle requête et une autre pour l'image sont construites (Figure.1.16(a) et (b)). Une recherche des maxima sur la surface de corrélation normalisée et calculée à partir des sommets de deux pyramides (Figure.1.16(d)) permettant de trouver les positions potentielles des occurrences du modèle requête. Les positions récupérées sont propagées, niveau par niveau, vers la base de la pyramide. Le niveau le plus bas de la pyramide contient les

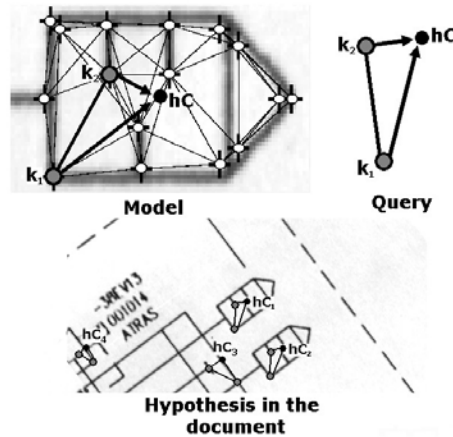


FIGURE 1.15 – Graphe de proximité ($k=5$) calculé à partir des points d'intérêt extraits. Chaque point d'intérêt possède un index dans la table de hachage. En se basant sur le graphe et ces indexes, une information sur l'organisation spatiale des descripteurs locaux similaires peut être déduite (extrait de [Rusiñol and Lladós2008]).

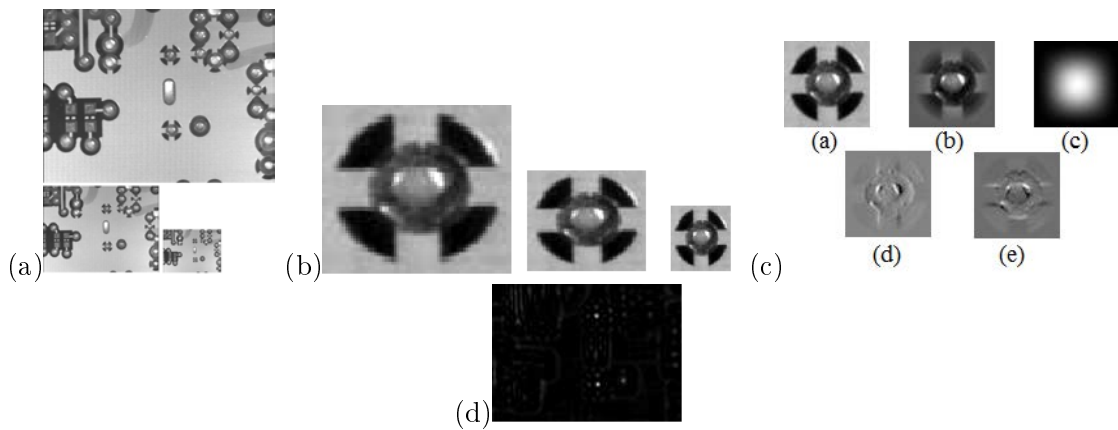


FIGURE 1.16 – (a) Représentation pyramidale de l'image, (b) Représentation pyramidale du modèle, (c) cette figure montre les différentes représentations nécessaires à chaque niveau de la pyramide pour la recherche de la corrélation de gradient. Le modèle est donné en (a). La version du modèle de fenêtres est représentée en (b), avec la fonction de fenêtrage représentée en (c), (d) Surface de corrélation au niveau le plus haut de l'image, les deux extrema les plus forts (blancs) correspondent aux positions de deux occurrences du modèle (extrait à partir de [MacLean and Tsotsos2008]).

propositions exactes des occurrences. Les auteurs proposent une analyse du pire des cas afin de trouver le nombre maximal de niveaux de la pyramide correspondant à chaque modèle. Les résultats de localisation sont prometteurs mais cette méthode est sensible au changement de densité du fond de l'image, à l'invariance à la rotation et au changement d'échelle. Dans ce même contexte, [Escalera *et al.*2009] propose une approche pixelaire basée sur le

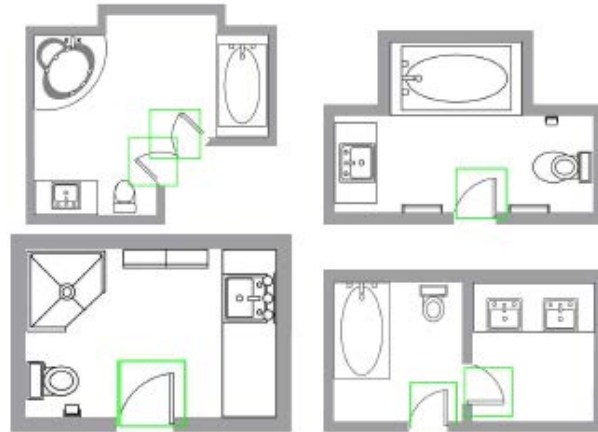


FIGURE 1.17 – Résultats de localisation de portes dans les documents de la méthode proposée par [Escalera *et al.*2009].

descripteur CBSM (*Circular Blurred Shape Model*). La méthode se base sur un parcours du document par une fenêtre glissante (le pas de déplacement est de 5 pixels). Un problème de cette approche concerne le choix des paramètres liés à la taille de la fenêtre et au pas de déplacement. Quelques résultats sont montrés dans la Figure.1.17.

Dans [Weber and Tabbone2012], les auteurs ont proposé une approche de localisation de symboles dans les documents graphiques en se basant sur la technique du *template matching*. Les auteurs ont proposé une adaptation de l'opérateur *hit-or-miss transform*. Ce dernier reste robuste à la translation et à la rotation. Un exemple des résultats de localisation des symboles dans les documents graphiques est montré dans la Figure.1.18.

Récemment, [Nguyen *et al.*2009] propose une approche pixelaire où les documents graphiques sont représentés par l'information extraite à partir des points d'intérêt, à des résolutions différentes (voir Figure.1.19). Un ensemble de mots visuels est construit par le regroupement des descripteurs similaires à partir des documents. Le modèle vectoriel et le fichier inverse qui sont des techniques de recherche d'information textuelle sont ensuite utilisés pour indexer et localiser les symboles. Une adaptation du *contexte de forme* de Belongie [Belongie *et al.*2002] en l'appliquant au niveau des points d'intérêt obtenus par le détecteur SIFT (*Scale-Invariant Feature Transform*) [Lowe1999] est proposée (CFPI). Les auteurs ont choisi ce détecteur malgré que d'autres méthodes s'avèrent plus robustes pour les symboles utilisés qui sont en niveau de gris car il est invariant au changement d'échelle. Une proposition a été faite dans ce travail en guise de garantir l'invariance du descripteur à la rotation et aux changements d'échelle en utilisant l'orientation θ_i du point d'intérêt comme l'axe des abscisses pour calculer les coordonnées relatives des points de contour. Une évaluation de cette méthode a montré que le descripteur CFPI retenu s'avère moins complexe au niveau de la représentation du symbole par rapport au *contexte de forme* (une réduction significative du nombre des points de contour et le nombre de points d'in-

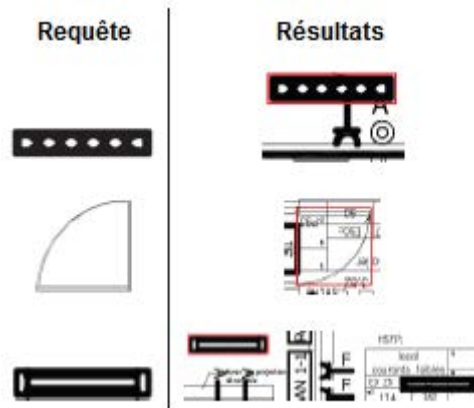


FIGURE 1.18 – Résultats de localisation de quelques symboles dans les documents graphiques de la méthode proposée par [Weber and Tabbone2012].

térêt retenus pour chaque symbole a été signalée en utilisant le CFPI). Bien que le descripteur proposé montre de bonnes performances en le comparant au *contexte de forme* [Belongie *et al.*2000], au GFD [Zhang and Lu2002] (*Generic Fourier Descriptor*) et à la \mathcal{R} -signature [Tabbone *et al.*2006], il présente quelques instabilités au sens de la répétition au niveau des courbes vu qu'il est basé sur les points d'intérêt SIFT. En fait, un symbole possédant peu de points d'intérêt stables, a une représentation moins discriminative par rapport à d'autres symboles.

1.2.3 Discussion

Après cet état de l'art des approches pixelaires de localisation de symboles dans les documents graphiques, nous pouvons constater que ces approches et malgré l'absence de l'aspect structurel, ne nécessitent aucune étape préalable de pré-traitement des documents graphiques. Un constat global que nous pouvons faire après l'évaluation de ces travaux de

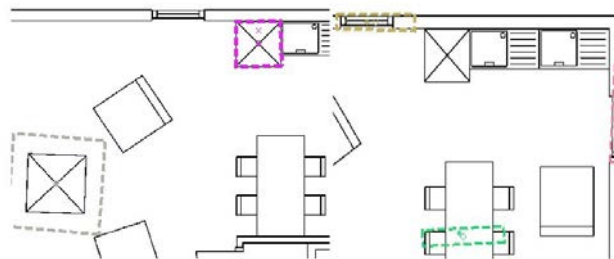


FIGURE 1.19 – Quelques Résultats de localisation de symboles obtenus avec la méthode [Nguyen *et al.*2009].

localisation et plus particulièrement l'approche de [Nguyen *et al.*2009], est que bien que ces approches fournissent de bonnes performances de localisation (voir Figure.1.19), certaines limites ont été détectées comme par exemple la complexité computationnelle du calcul, de la représentation et de l'appariement ainsi que la redondance des informations

définies par les descripteurs utilisés. De plus dans [Nguyen *et al.*2009], les expérimentations ont montré que seuls les symboles connectés ont la possibilité d'être localisés.

1.3 Conclusion

Dans ce chapitre, nous nous sommes efforcés de présenter un panorama des approches de localisation de symboles dans les documents graphiques qui ont été proposées dans la littérature. Ces travaux peuvent être classés en deux familles :

- Dans la première famille de méthodes, nous distinguons *les approches structurelles* qui nécessitent une étape préalable de segmentation de document en un ensemble de primitives (vectorisation, approximation polygonale, décomposition en segments...).

Une structure de graphes est généralement utilisée pour structurer ces différentes primitives de façon à bénéficier de leurs propriétés géométriques ainsi que de leurs relations (parallélisme, perpendicularité, lignes-arcs/cercles...). Autrement dit, si les primitives sont bien définies, l'usage des graphes permet de respecter des règles de combinaison ou de connexions entre les primitives, ce qui permet d'obtenir des résultats de localisation assez correctes. Une recherche d'un isomorphisme entre les graphes revient à retrouver les régions du document susceptibles de contenir les occurrences du symbole requête. Cet appariement est un problème *NP-complet* ce qui rend ces approches malgré la précision qu'elles pourront fournir lors de la localisation des positions de symboles dans le document, difficiles à appliquer dans le contexte de grandes bases de documents.

- Le deuxième type des approches de localisation proposées dans la littérature sont *les approches pixelaires* qui ne nécessitent pas d'étape préalable de vectorisation ou de segmentation. La localisation est effectuée directement sur l'image entière ce qui permet d'éviter le problème de complexité lié à la phase d'appariement de graphes comme est le cas des *approches structurelles*.

Ces approches montrent des résultats prometteurs dans le cadre de la localisation d'objets graphiques non isolés de leur contexte. Le constat que nous pouvons faire est que malgré la prolifération des systèmes de localisation, la détection d'objets graphiques non isolés de leur contexte demeure un problème ouvert vu la complexité des documents où les symboles apparaissent généralement connectés avec d'autres objets. En fait, toutes ces approches se heurtent non seulement au problème de la reconnaissance mais aussi au problème de la localisation. Ces méthodes doivent se focaliser sur des zones d'intérêt dans le document une fois que le symbole requête est recherché.

Notre première contribution dans le cadre du problème de localisation de symboles dans les documents graphiques sera d'évaluer quelques approches pixelaires. L'objectif de cette évaluation consiste à tester l'impact de l'étape d'extraction des primitives des documents graphiques sur les résultats de localisation. En effet, notre thèse s'inscrit dans le cadre du projet Européen EUREKA SCANPLAN où nous avons essayé d'apporter des réponses au problème de traitement des documents graphiques et plus précisément les plans architecturaux. Ce type de documents est assez complexe d'où la nécessité d'une méthode de segmentation robuste. Dans ce contexte, nous avons choisi d'évaluer quelques techniques de détection des points d'intérêt comme par exemple le détecteur Harris [Harris and Stephens1988], SURF [Bay *et al.*2006] et DoG [Lowe1999] utilisé dans l'approche de [Nguyen *et al.*2009]. Nous avons essayé d'évaluer l'impact de l'étape de représentation des documents sur les résultats de détection des symboles en contexte. Les approches utilisées se basent sur différents détecteurs de points d'intérêt mais nous avons choisi des descripteurs similaires pour

évaluer la totalité du système de localisation et voir l'influence de l'étape d'extraction de primitives sur toute la chaîne. Nous avons choisi les approches pixelaires de localisation car malgré l'absence de l'aspect structurel, il n'y a pas d'étape de pré-traitement ce qui permet de réduire la complexité de la chaîne de traitement de localisation. En vue de l'obtention d'une évaluation réaliste de l'étape d'extraction des primitives à partir des documents graphiques et son impact sur les performances de la localisation, nous avons opté pour la même stratégie de localisation adoptée dans [Nguyen *et al.*2009] car cette méthode était disponible dans l'équipe ainsi que la base d'expérimentations.

Dans le dernier chapitre de cette partie du manuscrit, nous avons proposé de réintégrer la structure du treillis de concepts déjà utilisée dans nos travaux présentés aux chapitres précédents. Nous avons opté pour une approche originale tirant bénéfice des capacités de la structure du treillis en l'intégrant dans une architecture d'un système de détection de symboles dans leur contexte. Cette approche propose une variante assez intéressante par rapport aux approches déjà proposées. Nous évaluons nos résultats dans un contexte de localisation en ayant recours aux bases de données utilisées dans la littérature afin de positionner nos travaux par rapport à l'existant.

Étude Comparative des Approches Pixelaires pour la Représentation et la Localisation de Symboles dans les Documents Graphiques

"La thématique Fouille de données (et connaissances) se focalise sur des méthodes d'apprentissage automatique et d'extraction de connaissances à partir de données complexes (ex : images, bases de données, etc.)."

Pierre Gançarski

Sommaire

2.1	Introduction	156
2.2	Formulation du problème	157
2.2.1	Le détecteur de Harris (1988)	158
2.2.2	Le Détecteur Harris-Laplace (2004)[Mikolajczyk and Schmid2001]	159
2.2.3	Le Détecteur Hessian-Laplace	160
2.2.4	Le Détecteur SIFT (DoG) [Lowe1999]	160
2.2.5	Le Détecteur <i>MSER</i> [Matas <i>et al.</i> 2004]	161
2.2.6	Discussion	162
2.2.7	Le Détecteur <i>pseudo-aléatoire</i>	163
2.3	Indexation du document graphique avec un vocabulaire visuel	163
2.3.1	Construction du Vocabulaire Visuel	164
2.3.2	Correspondance entre le Vocabulaire Visuel	164
2.3.3	Indexation du Document Graphique	165
2.3.4	Localisation de symboles dans les documents graphiques	166
2.4	Résultats Expérimentaux	168
2.4.1	Comparaison des Détecteurs de Points d'Intérêt	169
2.5	Discussion	171
2.6	Conclusion	171

2.7 Impact de l'étape de représentation sur les résultats de la localisation	172
2.7.1 Méthode a	172
2.7.2 Méthode b	173
2.7.3 Méthode c	173
2.7.4 Méthode d	173
2.8 Conclusion	182

2.1 Introduction

Dans le chapitre précédent, nous avons passé en revue un panorama des travaux relatifs à la détection des objets dans leur contexte. À la différence de la reconnaissance de symboles, outre l'étape de reconnaissance, ces approches introduisent l'étape de localisation dont l'objectif n'est plus de reconnaître simplement un objet mais aussi de localiser les zones du document susceptibles de contenir l'objet requête. Un constat que nous pouvons faire est que malgré la quantité importante de travaux dans ce contexte, l'étape d'extraction des caractéristiques des objets graphiques demeure un problème ouvert. Cette étape est primordiale et constitue une part très importante et décisive de la chaîne de traitement des systèmes de reconnaissance ou de localisation. Les descripteurs utilisés ont besoin d'être simples et compacts ayant un pouvoir discriminant et modélisant assez élevé pour satisfaire les objectifs du système proposé. Nous avons remarqué que quelques descripteurs sont jugés comme dépendants et plus adaptés aux images photométriques en niveaux de gris. Une question découle de cette interprétation est la suivante : *Est ce que ces descripteurs sont également adaptés aux documents graphiques ?*, *Y-aura-elle vraiment une différence significative entre les résultats de détection fournis en ayant recours à tel ou tel détecteur ?*. Nous nous sommes alors orientés vers l'évaluation de quelques approches pixelaires de détection des symboles dans leur contexte en vue d'évaluer l'étape d'extraction des primitives à partir des documents graphiques. Plus précisément, l'étude de l'approche de [Nguyen *et al.*2009] nous a permis de cerner les étapes de la chaîne de traitement. Nous avons opté pour cette approche car les bases de documents utilisés dans le protocole expérimental sont disponibles permettant ainsi, l'obtention d'une évaluation assez réaliste et objective.

Ainsi, nous avons choisi d'évaluer ces descripteurs et leurs détecteurs correspondants dans le cadre d'une approche de localisation en les mettant en comparaison par rapport à un détecteur qui ne se base sur aucun modèle (photométrie ou géométrie) et qui concerne la détection aléatoire des points. Ce détecteur sera détaillé par la suite. En ce qui concerne le processus de localisation, nous avons eu recours à la même stratégie adoptée dans [Nguyen *et al.*2009] où les auteurs proposent d'utiliser le modèle vectoriel et le fichier inverse appartenant au domaine de la recherche d'information textuelle. Le document graphique est textualisé en utilisant le vocabulaire visuel de façon à construire un index de mots visuels. Le document graphique est alors traité comme un ensemble de mots à appairer afin de trouver des correspondances entre les mots représentant le symbole requête et les mots représentant le document. Donc dans ce chapitre, notre contribution consiste à évaluer l'impact de l'étape d'extraction des primitives des documents graphiques sur les résultats de localisation. Nous avons comparé plusieurs détecteurs usuels de points d'intérêt, soit d'une façon indépendante en terme du nombre des points détectés et du temps nécessaire à leur extraction, soit dans le cadre d'une chaîne de localisation.

La suite de ce chapitre est organisée comme suit. Dans une première section, une formulation du problème est posée suivie d'un panorama des détecteurs d'indices visuels des images. Dans une deuxième section, nous décrivons nos choix et détaillons les solutions proposées à la problématique de localisation. Enfin, une évaluation quantitative et qualitative des méthodes de représentation des primitives est fournie. Cette évaluation vise à analyser l'impact de cette étape sur les résultats de localisation obtenus par les approches pixelaires retenues dans ce travail.

2.2 Formulation du problème

Dans la littérature, les contributions à la problématique d'extraction des primitives sont basées sur les détecteurs classiques des points d'intérêt où un descripteur local est calculé au voisinage de ces points. Des études récentes [Mikolajczyk and Schmid2005] montrent la robustesse de ces méthodes dans les systèmes de reconnaissance de formes. Paradoxalement à ce nombre considérable de méthodes, nous ne pouvons pas privilégier telle ou telle méthode car le contexte traité nécessite une méthode assez robuste pour bien extraire les informations incluses dans le document.

Dans ce travail, nous nous sommes orientés vers l'étude de l'impact de l'étape de représentation des documents sur les résultats de localisation. Cette étape est déterminante pour l'obtention de bons taux de détections des symboles dans leur contexte. Nous avons comparé quelques détecteurs usuels de points d'intérêt à savoir le détecteur de Harris [Harris and Stephens1988], SURF [Bay *et al.*2006] et DoG [Lowe1999] utilisé dans [Nguyen *et al.*2009]. Cette idée d'évaluer l'impact de l'étape de représentation des documents sur les performances de détection a été émergée suite à l'évaluation de l'approche de [Nguyen *et al.*2009]. Bien que cette approche ait montré de bonnes performances de localisation, certains symboles sont mal localisés, nous avons donc décidé d'étudier les causes de cette dégradation. Ainsi, notre évaluation comparative sera en fonction d'un détecteur de points aléatoires qui servira d'oracle et qui ne se repose sur aucun modèle (géométrie ou photométrie). Plus précisément, il s'agit d'un algorithme *semi-aléatoire* où nous proposons d'extraire des points à partir du document d'une façon aléatoire suivie d'un processus de contrôle basé sur l'extraction des régions stables au voisinage des points détectés. La stabilité est définie par les critères du détecteur *MSER* [Matas *et al.*2004]. Cette technique est exemptée de contraintes à poser sur les documents graphiques utilisés. L'idée est de voir comment se comporte nos détecteurs par rapport à l'oracle.

Dans ce qui suit, nous allons commencer par une présentation de quelques descripteurs usuels. De nombreuses méthodes ont été proposées pour détecter des points d'intérêt. Elles peuvent être classées grossièrement suivant trois catégories :

- *Approches contours* : l'idée est de détecter les contours dans une image dans un premier temps. Les points d'intérêt sont ensuite extraits le long des contours en considérant les points de courbures maximales ainsi que les intersections de contours (voir Figure.2.1).
- *Approches intensité* : l'idée est cette fois-ci de regarder directement la fonction d'intensité dans les images pour en extraire directement les points de discontinuités (voir Figure.2.2).
- *Approches à base de modèles* : les points d'intérêt sont identifiés dans l'image par mise en correspondance de la fonction d'intensité avec un modèle théorique de la fonction des points d'intérêt considérés (voir Figure.2.3).

Les approches de la deuxième catégorie sont celles utilisées généralement. Les raisons de leurs proliférations sont : (a) leur indépendance vis à vis de la détection de contours (sta-

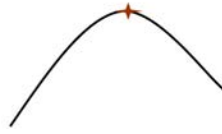


FIGURE 2.1 – Approche contour : un point d'intérêt peut être un point d'une courbure.

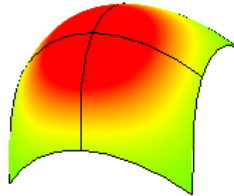


FIGURE 2.2 – Approche photométrique.

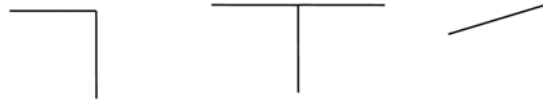


FIGURE 2.3 – Approche modèle.

bilité) et, (b) leur indépendance vis à vis du type de points d'intérêt (méthodes plus générales). Le principe consiste à faire émerger des zones pouvant caractériser le plus fidèlement possible la structure de l'image. Dans cette partie, nous allons évoquer quelques détecteurs qui nous paraissent les plus communs dans le cadre de nos travaux.

2.2.1 Le détecteur de Harris (1988)

Le détecteur de Moravec [Moravec1977] fonctionne dans un contexte limité. Il souffre en effet de nombreuses limitations. Harris et Stephen ont identifié certaines limitations et, en les corrigeant, en ont déduit un détecteur de coins très populaire : le détecteur de Harris [Harris and Stephens1988] (voir Figure.2.4).

Les limitations du détecteur de Moravec [Moravec1977] prises en compte sont les suivantes :

1. La réponse du détecteur est anisotrope en raison du caractère discret de changement des directions que l'on peut effectuer (des pas de 10 degrés). Pour améliorer cet aspect, il suffit de considérer le développement de Taylor de la fonction d'intensité \mathcal{I} au voisinage du pixel (u, v) :

$$\mathcal{I}(x + u, y + v) = \mathcal{I}(u, v) + x \frac{\delta \mathcal{I}}{\delta x} + y \frac{\delta \mathcal{I}}{\delta y} + o(x^2, y^2) \quad (2.1)$$

D'où :

$$\mathcal{E}(x, y) = \sum_{u,v} w(u, v) [x \frac{\delta \mathcal{I}}{\delta x} + y \frac{\delta \mathcal{I}}{\delta y} + o(x^2, y^2)]^2, \quad (2.2)$$

le terme $o(x^2, y^2)$ est négligé (pour les petits déplacements, on a l'expression analytique suivante :

$$\mathcal{E}(x, y) = Ax^2 + 2Cxy + By^2, \quad (2.3)$$

avec :

$$\mathcal{A} = \frac{\delta \mathcal{I}^2}{\delta x} \otimes w \quad (2.4)$$

$$\mathcal{B} = \frac{\delta \mathcal{I}^2}{\delta y} \otimes w \quad (2.5)$$

$$\mathcal{C} = \frac{\delta \mathcal{I}}{\delta x} \frac{\delta \mathcal{I}}{\delta y} \otimes w \quad (2.6)$$

2. La réponse du détecteur de Moravec est bruitée en raison du voisinage considéré. Le filtre w utilisé est en effet binaire et est appliqué sur un voisinage rectangulaire. Pour améliorer cela, Harris et Stephen propose d'utiliser un filtre Gaussien.

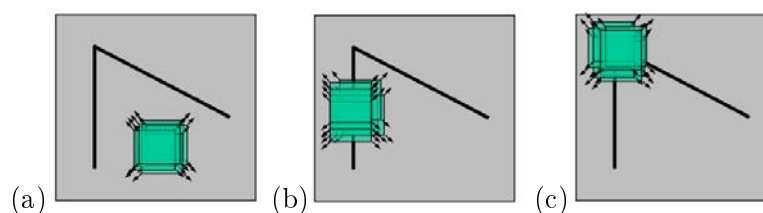


FIGURE 2.4 – L'idée principale du détecteur de Harris : (a) "flat" : aucun changement de directions, (b) contour : pas de changement tout au long du contour. (c) changements significatifs dans toutes les directions.

2.2.2 Le Détecteur Harris-Laplace (2004)[Mikolajczyk and Schmid2001]

Le principe de ce détecteur repose sur la recherche des maxima locaux de R (voir Figure.2.5). Tout d'abord, la détermination des points d'intérêt d'une image est obtenue en

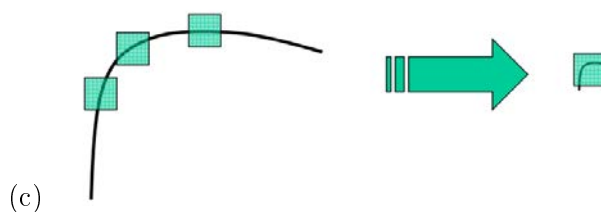


FIGURE 2.5 – Recherche des maxima locaux de R .

utilisant la formule de Harris suivante :

$$R = \text{Det}(M) - k \cdot [\text{Tr}(M)]^2 \quad (2.7)$$

La matrice des moments du 2d ordre (matrice d'autocorrélation) est :

$$M(x, y, \sigma_I, \sigma_D) = G_{\sigma_I} * \begin{pmatrix} L_x^2(x, y, \sigma_D) & L_x L_y(x, y, \sigma_D) \\ L_x L_y(x, y, \sigma_D) & L_y^2(x, y, \sigma_D) \end{pmatrix} \quad (2.8)$$

où L est l'image lissée avec une gaussienne G d'écart type σ .

$$L_x = \frac{L}{\partial x}, \dots \quad (2.9)$$

$$L(x, y, \sigma) = (G_\sigma * I)(x, y) \quad (2.10)$$

et la recherche des maxima locaux de $|\text{LoG}|$ (pour la sélection de l'échelle caractéristique) (voir Figure.2.6) :

$$|\text{LoG}(x, \sigma_D)| = \sigma_D^2 \cdot |L_{xx}(x, \sigma_D) + L_{yy}(x, \sigma_D)| \quad (2.11)$$

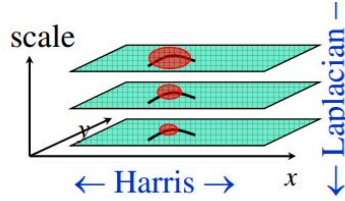


FIGURE 2.6 – Recherche des maxima locaux de Harris-Laplace pour la sélection de l'échelle caractéristique.

2.2.3 Le Détecteur Hessian-Laplace

Ce détecteur est une variante de Harris-Laplace utilisant la hessienne. Son principe consiste à chercher des maxima locaux de $\det(\mathcal{H})$:

$$\mathcal{H}(x, y, \sigma_D) = \begin{pmatrix} L_{xx}(x, y, \sigma_D) & L_{xy}(x, y, \sigma_D) \\ L_{xy}(x, y, \sigma_D) & L_{yy}(x, y, \sigma_D) \end{pmatrix} \quad (2.12)$$

$$L_{xy} = \frac{\partial^2 L}{\partial x \partial y}, \dots \quad (2.13)$$

$$L(x, y, \sigma) = (G_\sigma * I)(x, y) \quad (2.14)$$

et chercher des maxima locaux de $|\text{LoG}|$ (pour sélection de l'échelle caractéristique) :

$$|\text{LoG}(x, \sigma_D)| = \sigma_D^2 \cdot |L_{xx}(x, \sigma_D) + L_{yy}(x, \sigma_D)| \quad (2.15)$$

2.2.4 Le Détecteur SIFT (DoG) [Lowe1999]

Comme nous l'avons évoqué avant, nous abordons le problème de représentation des primitives afin d'améliorer le système de localisation proposé par [Nguyen *et al.*2009]. Dans ce travail, le détecteur DoG a été utilisé.

$$D(x, y, \sigma) = L_{k.\sigma_t}(x, y) - L_{\sigma_t}(x, y) = ((G_{k.\sigma_t} - G_{\sigma_t}) * I)(x, y) \quad (2.16)$$

Un point détecté correspond à un extrema à chaque niveau d'une pyramide d'échelles, obtenue suite à la convolution d'une image \mathcal{I} avec des filtres de DoG de tailles différentes. Nous remarquons d'après la Figure.2.7(a) que chaque niveau de la pyramide contient s images DoG. Chaque image est lissée par des filtres gaussiens qui diffèrent par un facteur d'échelle k . L'image DoG est obtenue en calculant la différence entre deux images. Le facteur k est fixé à $2^{\frac{1}{s}}$ afin d'obtenir k images DoG à chaque niveau de la pyramide. L'image initiale d'un niveau m (niveau suivant) est obtenue par un rééchantillonnage d'un facteur 2 de la

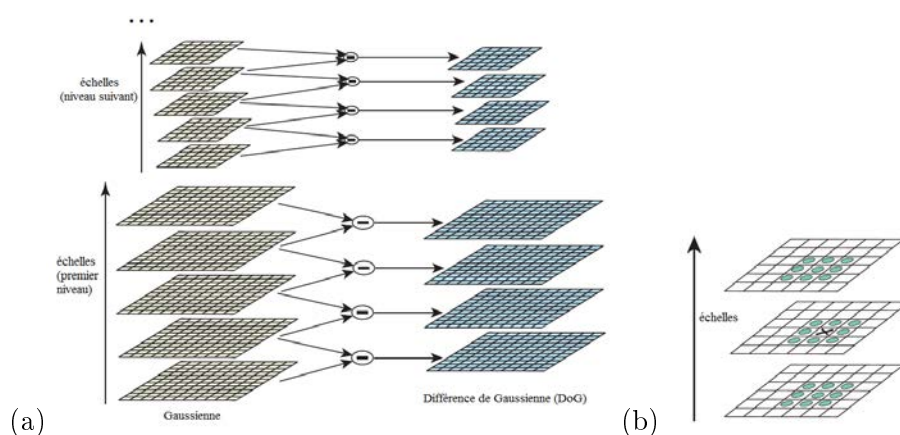


FIGURE 2.7 – Le détecteur de points d’intérêt utilisé dans [Lowe1999] : (a) Construction de la pyramide d’échelles. (b) Détection des extrema des images de DoG : les maxima et minima du DoG sont détectés en comparant un point (marqué par X) avec les 26 points dans la région de voisinage 3x3x3 (marqués par des cercles) de l’échelle courante et de deux échelles adjacentes (extrait de [Lowe1999]).

dernière image gaussienne du niveau $m-1$ (niveau précédent). Ces opérations sont itérées afin d’obtenir une pyramide complète.

$$\mathcal{D}(x, y, \theta) = \mathcal{G}(x, y, k\theta) - \mathcal{G}(x, y, \theta) * \mathcal{I}(x, y) \quad (2.17)$$

$$\mathcal{G}(x, y, \theta) = \frac{1}{2\pi\theta^2} e^{-\frac{(x^2+y^2)}{2\pi\theta^2}} \quad (2.18)$$

Les points d’intérêt se trouvent aux maxima et minima de $\mathcal{D}(x, y, \theta)$. Chaque point est comparé avec ses 8 voisins de l’image actuelle et aux 18 voisins de deux images adjacentes (voir Figure.2.7(b)). Un point est maxima, respectivement minima s’il est plus grand, respectivement plus petit que tous ses voisins. Les points trouvés sont cédés à un processus de filtrage pour les rendre plus stables en utilisant la matrice de Hessian (plus précisément en utilisant son déterminant et sa trace) calculée à la position et l’échelle du point. Enfin, un point d’intérêt p_i est caractérisé par un quadruplet $(x_i, y_i, \sigma_i, \theta_i)$ où (x_i, y_i) sont les coordonnées de p_i , σ_i la résolution ($\sigma_i = \theta$ dans l’équation.2.17) et θ_i désigne l’orientation du point déterminée par le gradient local dominant dans la région entourant le point considéré. Ce détecteur possède les propriétés suivantes :

- Invariant aux changements d’échelle
- Invariant aux changements de point de vue
- Robuste à l’addition de bruit
- Robuste aux changements d’image de fond
- Robuste aux occlusions
- Partiellement invariant aux changements d’illumination

2.2.5 Le Détecteur *MSER* [Matas *et al.*2004]

Matas [Matas *et al.*2004] ont proposé une approche originale permettant de définir une région d’intérêt robuste aux transformations affines (voir Figure.2.8). Premièrement, une étape de classification permet de regrouper en classe chaque pixel de l’image suivant sa valeur d’intensité. L’histogramme ainsi créé permet de déterminer une fonction d’intensité

en se basant sur l'aire de chaque classe. En appliquant différents seuillages à cette fonction d'intensité, certaines régions de l'image classifiée vont varier et d'autres non. Les régions d'intérêt R_i correspondent donc à celles qui restent robustes aux différents seuillages. Matas et al. démontrent que les régions d'intérêt ainsi extraites, sont invariantes aux transformations affines, aussi bien photométriques que géométriques.

$$\forall p \in \mathcal{R}_i, \forall q \in \text{boundary}(\mathcal{R}_i) \leftarrow I_{in}(p) \geq I_{in}(q) \quad (2.19)$$

$$I_{bin}^g = \begin{cases} 1 & \text{si } I_{in} \geq g \\ 0 & \end{cases}$$



FIGURE 2.8 – Application du détecteur *MSER* sur quelques images (<http://www-master.ufr-info-p6.jussieu.fr/specialite/ima/bima/cours7.2p.pdf>).

2.2.6 Discussion

Notre objectif est d'évaluer l'impact de l'étape de représentation des documents sur les résultats de localisation et de voir vraiment s'il est possible de trancher entre les différentes méthodes et voir si une méthode peut se démarquer du lot. Récemment, beaucoup d'auteurs ont commencé à utiliser des détecteurs multi-échelle comme par exemple le détecteur LoG (Laplacian of Gaussian) ou le détecteur de Harris [Harris and Stephens1988] mais bien que de tels détecteurs aient prouvé leur robustesse pour l'appariement des images, ils n'ont pas été conçus pour trouver les primitives contenant le plus d'information pour la localisation de symboles dans les documents graphiques. Une première interprétation qui découle des travaux de la littérature, est qu'il est difficile de se prononcer pour telle ou telle méthode car elles ont toutes des avantages ainsi que des inconvénients :

- Les détecteurs basés sur le Hessien obtiennent de meilleurs résultats que ceux basés sur les régions de Harris.
- Le détecteur Hessien Affine fournit de meilleurs résultats pour les changements d'échelle.
- Le détecteur SIFT obtient des résultats pour les changements de point de vue meilleurs que les autres détecteurs.
- Le détecteur *MSER* obtient les meilleurs résultats et a montré son invariance aux changements d'échelle, aux rotations ainsi qu'aux occlusions en le comparant aux autres détecteurs.

En vue d'évaluer l'impact de l'étape d'extraction et de description de primitives sur les performances du système de localisation, nous l'avons évalué par rapport aux résultats

fournis par quelques approches pixelaires de localisation. En plus, nous pouvons tester si les descripteurs photométriques sont viables aux documents graphiques traités dans nos travaux.

2.2.7 Le Détecteur *pseudo-aléatoire*

En analysant quelques approches de détection de symboles dans leur contexte, nous nous sommes rendus-compte que le choix d'une bonne méthode d'extraction des primitives est un problème ouvert. Notre problématique se situe dans le cadre de l'évaluation de l'impact de l'étape d'extraction des primitives sur les résultats de détection. La méthode retenue pour l'extraction des indices visuels à partir des documents graphiques peut être caractérisée de *pseudo-aléatoire* car la détection aléatoire des points est contrôlée par la détermination d'un voisinages stable autour de ces derniers en se basant sur le détecteur *MSER* [Matas *et al.*2004] dans le but d'obtenir l'information la plus fine possible. Cette alliance intéressante entre l'extraction des points de l'image d'une façon aléatoire et la détermination de leurs voisinages stables permet de bénéficier d'une part de la richesse de l'information que peut apporter la bonne répartition des points dans toute l'image et, d'autre part de la stabilité des régions d'intérêt où des descripteurs robustes peuvent être calculés. Malgré la simplicité et le manque de géométrie globale de la technique aléatoire, elle s'est avérée être étonnement discriminante, et présentait un outil de référence (oracle) efficace dans notre évaluation. Dans la suite, nous allons décrire les étapes de la chaîne de traitement de localisation où nous avons intégré le détecteur aléatoire afin d'évaluer son impact sur les performances du système.

2.3 Indexation du document graphique avec un vocabulaire visuel

Nous rappelons dans cette section l'approche de localisation de symboles utilisée dans [Nguyen *et al.*2009] qui repose sur un vocabulaire visuel. Après l'application du détecteur *pseudo-aléatoire* et le calcul des descripteurs au niveau du voisinage "stable" des points, le document graphique est indexé en utilisant un vocabulaire visuel. L'indexation d'un symbole consiste alors à indexer séparément chacune de ses primitives. Cette approche ne prend pas en compte la cohérence spatiale des primitives extraites. Récemment, une méthode pour localiser les objets dans des images en utilisant la représentation par sac-mots a été proposée dans [Sivic and Zisserman2009].

La représentation de l'image à l'aide de l'approche sac mots est couramment utilisée dans la classification des images et nous allons l'étendre à l'approche de localisation. Tout d'abord, la représentation des images est effectuée en deux phases : la création d'un vocabulaire visuel et la représentation de l'image en utilisant ce vocabulaire. Le vocabulaire \mathcal{V} est obtenu en quantifiant l'ensemble des descripteurs issus de l'étape précédente à l'aide de l'algorithme *k-means*.

Le processus comporte trois étapes : 1) la sélection des régions stables au voisinage des points d'intérêt détectés aléatoirement, 2) le calcul des descripteurs SIFTs au niveau des régions déjà extraites et, 3) la quantification des descripteurs SIFTs. Dans la littérature et dans ce contexte, [Agarwal *et al.*2004], [Jurie and Triggs2005], [Sivic and Zisserman2009] ont proposé différentes techniques pour la quantification des descripteurs en vue de construire un dictionnaire visuel. Nous avons choisi d'appliquer l'algorithme *k-means* afin de construire

le vocabulaire visuel qui décrit le document graphique.

Une fois que ce dernier est décrit par des mots visuels, le modèle vectoriel est utilisé pour indexer l'ensemble du document, par analogie au document textuel. Ensuite, une structure de fichier inverse est utilisé pour indexer le contenu afin de trouver les correspondances avec des régions candidates dans chaque document.

2.3.1 Construction du Vocabulaire Visuel

Pour notre approche, la caractérisation des images se fait par les points d'intérêt détectés de façon aléatoire et représentés par leurs ellipses englobantes. Ces régions sont alors décrites par le descripteur SIFT. Le regroupement des classes se fait en appliquant l'algorithme *k-means* sur l'ensemble des descripteurs pour obtenir k classes ou clusters de descripteurs. Chaque centre de cluster représente un mot visuel (voir Figure.2.9).

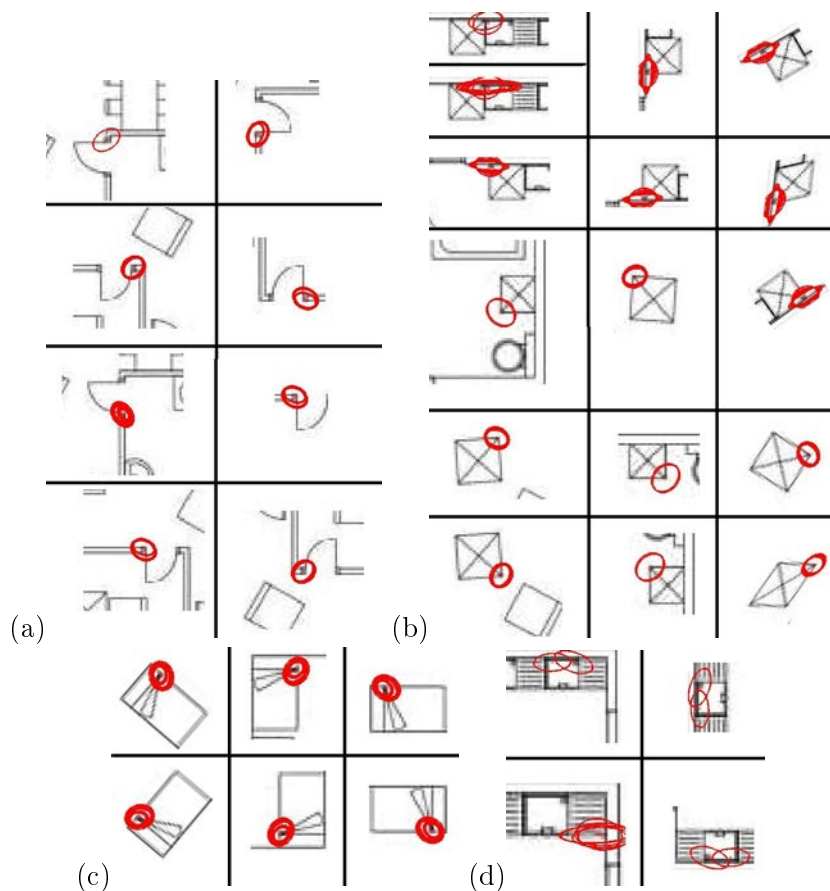


FIGURE 2.9 – Exemples de quelques indices visuels délimités par des ellipses rouges présentant 4 classes de mots visuels.

2.3.2 Correspondance entre le Vocabulaire Visuel

L'approche proposée par [Nguyen *et al.*2009] repose sur la correspondance de chaque point d'intérêt à partir d'un symbole avec un mot visuel. Il n'est pas sûr que n'importe quelle méthode de quantification puisse fournir une classification parfaite, c'est pourquoi, afin de minimiser la confusion du regroupement, l'approche se base sur l'association d'un point

d'intérêt avec un ou plusieurs mots visuels en fonction de la similarité entre ce point et tous les mots visuels. Un degré de confiance est attribué à chaque appariement entre un point d'intérêt et les mots qui lui sont associés. Le degré de confiance tel que décrit dans [Nguyen *et al.*2009] est défini comme suit :

$$dConf_{p_i, w_j} = \frac{S_{p_i, w_j}}{\sum_{w_k \in V_i} S_{p_i, w_k}} \quad (2.20)$$

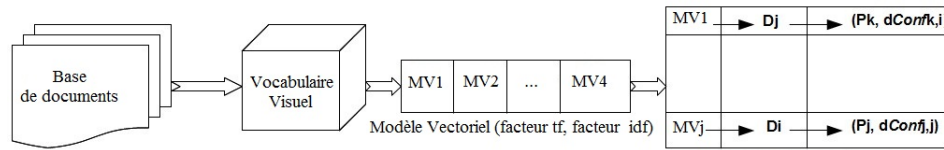
où V_i est l'ensemble des mots visuels correspondant au point p_i et S_{p_i, w_j} est la similarité entre le point p_i et le mot w_j .

$$V_i = \{w_j \in Voc \mid \frac{S_{p_i, w_j}}{S_{p_i, w_0}} > \epsilon\} \quad (2.21)$$

avec Voc est le vocabulaire visuel, w_0 est le plus proche mot de p_i et ϵ est un certain seuil (fixé à 0.96 dans les résultats de [Nguyen *et al.*2009]).

2.3.3 Indexation du Document Graphique

Comme nous l'avons largement détaillé, le système de localisation adopté se base sur des techniques utilisées généralement dans le cadre de la recherche textuelle qui sont respectivement le modèle vectoriel et le fichier inverse (voir Figure.2.10).



Structure de fichier inverse d'indexation du vocabulaire visuel

FIGURE 2.10 – Les étapes de la chaîne de traitement pour la localisation de symboles dans les documents graphiques.

Modèle Vectoriel Une fois que le document est décrit par des mots visuels, par analogie au document textuel, la structure du modèle vectoriel est utilisé. Avec ce modèle, un document est alors représenté par un vecteur de fréquences d'apparition de mots (voir Figure.2.10). Ce vecteur est pondéré par le degré d'importance relatif aux mots, deux facteurs sont alors définis :

- le facteur *tf* : désigne la fréquence d'apparition des mots (termes) dans le document.
- le facteur *idf* : désigne l'importance du mot dans le document (selon ce facteur, on distingue un document *pertinent* ou *non pertinent*).

Dans ce modèle (voir Figure.2.11), un document est représenté comme un vecteur de fréquences de mots. Il est généralement décrit par le vecteur des poids pondérés des fréquences dont la composante fournit la valeur de deux facteurs : le *terme de fréquence* (facteur *tf*) et le *terme inverse de fréquence* (facteur *idf*) [Baeza-Yates and Ribeiro-Neto2011]. Un document est alors représenté par un vecteur de *tf-idf* :

$$\vec{doc_j} = \{f_{1,j}, f_{2,j}, \dots, f_{\kappa,j}\} \quad (2.22)$$

où \mathcal{K} est la taille du vocabulaire Voc et $f_{i,j}$ la fréquence pondérée du mot i dans le document j .

$$f_{i,j} = tf_{i,j} * idf_{i,j}, i = 1, \mathcal{K} \quad (2.23)$$

La stratégie adoptée pour la détermination des documents similaires à la requête est celle proposée dans [Nguyen *et al.*2009] en améliorant l'étape de représentation des primitives comme nous l'avons décrit dans les sections précédentes :

- Calculer les descripteurs SIFTs de la requête.
- Apparier les mots visuels avec les descripteurs.
- Calculer les fréquences pondérées des mots existants dans la requête [Nguyen *et al.*2009].

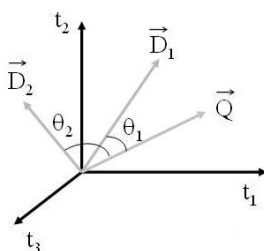


FIGURE 2.11 – Une représentation de deux documents (D_1, D_2) et une requête (Q) dans un espace de 3 termes (t_1, t_2, t_3).

Fichier Inverse L'idée retenue dans [Nguyen *et al.*2009] repose en plus du modèle vectoriel qui s'avère non satisfaisant pour répondre aux besoins de l'approche de localisation, sur la structure du fichier inverse.

Le fichier inverse est composé de deux éléments : le vocabulaire et les occurrences des mots (voir Figure.2.10). Le vocabulaire est l'ensemble des mots visuels. Pour chaque mot visuel, une liste de toutes les positions de ses occurrences est sauvegardée. Chaque élément de ces listes correspond à un document et contient les positions de tous les points correspondants à ce mot avec leur degré de confiance [Nguyen *et al.*2009].

2.3.4 Localisation de symboles dans les documents graphiques

Comme nous l'avons développé précédemment, l'objectif du système proposé n'est plus de reconnaître un symbole requête mais de localiser les endroits du document graphique les plus probables de contenir les occurrences les plus vraisemblables à la requête. Pour cela, les étapes de la chaîne de traitement du système de détection d'un symbole requête dans son contexte sont les suivantes : (1) le calcul des descripteurs SIFTs au niveau des régions stables entourant un point choisi aléatoirement, (2) la mise en correspondance des mots visuels construits sur la base de documents avec les descripteurs SIFTs du symbole, (3) la recherche des documents contenant le symbole requête en utilisant le fichier inverse. Cette recherche consiste à extraire les régions du document susceptibles de contenir les symboles similaires à la requête. Dans ce qui suit, nous allons présenter la technique adoptée pour extraire ces régions.

Régions d'intérêt

Étant donné un symbole requête, les points d'intérêt et les mots visuels qui y correspondent sont déterminés comme décrit dans la section précédente.

Nous avons défini les centres des objets $C(x_C, y_C)$ dans la requête et dans le rectangle englobant $rect=(x_C, y_C, w, h)$ avec w, h représentent respectivement la largeur et la hauteur du rectangle. Maintenant, nous essayons de localiser les régions du document susceptibles de contenir le symbole requête. Ces dernières sont déterminées en se basant sur la relation entre le point d'intérêt et $rect$ [Nguyen *et al.*2009].

Si le point $p_i = (x_i, y_i, \gamma_i, \theta_i)$ (dans la requête) et $p_j^d = (x_j^d, y_j^d, \gamma_j^d, \theta_j^d)$ (dans le document) sont associés au même mot visuel, une région $rect_j^d = (x_{C_j^d}, y_{C_j^d}, w_j^d, h_j^d, \psi_j^d)$ dans le document est déterminée en fonction de $\{rect, p_i, p_j^d\}$, à travers quelques opérations de rotation et de translation. p_i, p_j^d sont considérés comme des points de contrôle des deux rectangles respectivement $rect, rect_j^d$.

Nous avons constaté que dans le travail de [Nguyen *et al.*2009], chaque région du document est représentée par un ensemble de mots visuels, sauf que ces derniers ne sont pas liés par leurs relations spatiales ce qui a entraîné un problème de faux positifs. En effet, il peut arriver qu'une région ne contenant pas des occurrences du symbole requête reçoit une grande valeur de vote. Cela est dû au fait que la région et le symbole partagent les mêmes mots visuels sauf que ces derniers sont liés d'une façon différente.

Stratégie de Vote

En se basant sur le modèle vectoriel [Baeza-Yates and Ribeiro-Neto2011], un vote est attribué pour le centre de chaque région d'intérêt basé sur la similarité entre la région et la requête [Nguyen *et al.*2009].

Supposons, que $W_r = \{w_{r_1}, w_{r_2}, \dots, w_{r_K}\}$ est l'ensemble des mots attribués à un point d'une région donnée r avec le degré de confiance $\{dConf_{r_1}, dConf_{r_2}, \dots, dConf_{r_K}\}$.

La fréquence d'apparition d'un mot w_j dans la région est définie par tf_j^r où :

$$tf_j^r = \frac{\sum_{w_{r_k} \in W_r, w_{r_k} \equiv w_j} dConf_{r_k}}{\sum_{k=1}^M dConf_{r_k}} \quad (2.24)$$

et le poids de la fréquence du mot w_j dans la région est défini comme suit :

$$wf_j^r = tf_j^r * idf_j \quad (2.25)$$

où idf_j est le terme inverse de la fréquence du mot w_j . La région est représentée par le vecteur de fréquences v^r .

$$v^r = (wf_1^r, wf_2^r, \dots, wf_{||Voc||}^r) \quad (2.26)$$

L'idée de base consiste à découper le document en régions distinctes encadrant chaque symbole. Une stratégie de vote majoritaire est appliquée au niveau de chaque région. Les régions où les votes atteignent leur maximum seront sélectionnées comme candidates susceptibles de contenir le symbole requête. Généralement, l'espace de vote est un espace à 3 dimensions (x, y, s) composé de coordonnées 2D, x, y et un rapport d'échelle s .

Étant donné une requête P_q , nous accumulons des votes au niveau des coordonnées du document graphique $v_i = (x_i, y_i)$. Le système de localisation a pour objectif de trouver les régions d'intérêt du document où un symbole interrogé est susceptible d'être trouvé en terme de votes accumulés. Dans l'espace de vote, les votes cohérents ont tendance à former

des pics saillants, le reste de votes seront dispersés à différents endroits. Un classement des votes permet de déterminer les endroits du document susceptibles de contenir le symbole requête.

Comme notre approche vise à évaluer l'impact de l'étape de représentation des documents graphiques sur les résultats de localisation, nous avons choisi d'évaluer les performances de notre détecteur avec les détecteurs de l'état de l'art les plus communs à notre travail. Cela nous emmène implicitement à évaluer la complexité de calcul des descripteurs. En effet, les méthodes de détection des points d'intérêt devraient idéalement identifier les régions les plus riches en information. En conséquence, nous avons étudié les méthodes d'extraction des primitives en fonction du nombre des points détectés ce qui permettra d'avoir un avis sur la complexité algorithmique de l'étape d'extraction des primitives qui représente la première étape de la chaîne de traitement mise en place.

2.4 Résultats Expérimentaux

Dans cette série d'expérimentations, nous avons analysé le rendement des détecteurs de l'état de l'art à savoir le détecteur de coins de Harris [Harris and Stephens1988], le détecteur SURF [Bay *et al.*2006], le détecteur DoG [Lowe1999] utilisé dans [Nguyen *et al.*2009] ainsi que le détecteur *pseudo-aléatoire*. Les détails de ces évaluations seront étalés dans la section suivante. Nous avons effectué les tests sur des bases issues de la compétition de reconnaissance de symboles qui a eu lieu durant le workshop GREC'03¹⁹.

Cette base est composée de 50 modèles de symboles segmentés ou isolés. Nous distinguons six catégories de bruits résultant de cette base que nous appelons **GREC** :

- Modèles de symboles obtenus par des transformations linéaires : changement d'échelle (voir Figure.2.12(a)). Nous appelons cette base **GREC_{scale}**²⁰.
- Modèles de symboles obtenus par des transformations linéaires : changement de rotation (voir Figure.2.12(b)). Nous appelons cette base **GREC_{rot}**²¹.
- Modèles obtenus par des transformations linéaires : changement d'échelle et de rotation (voir Figure.2.12(c)). Nous appelons cette base **GREC_{scale+rot}**²².
- Modèles dégradés (voir Figure.2.12(d)). Nous appelons cette base **GREC_{degr1}**²³.
- Modèles ayant subi une forte dégradation (niveau 1) (voir Figure.2.12(e)). Nous appelons cette base **GREC_{degr1}**²⁴.
- Modèles ayant subi une forte dégradation (niveau 2) (voir Figure.2.12(f)). Nous appelons cette base **GREC_{degr2}**²⁵.
- Modèles ayant subi une distorsion (niveau 1) (voir Figure.2.12(g)). Nous appelons cette

19. <http://www.cvc.uab.es/grec2003/SymRecContest/index.htm>

20. Cette base contient 250 symboles de la base **GREC** ayant subi des transformations d'échelle. Le facteur d'échelle varie entre 0.5 et 1.5. Le nombre d'occurrences de chaque modèle varie entre 0 et 13.

21. Cette base contient 250 symboles de la base **GREC** ayant subi des rotations avec un angle $\alpha \in [0, \pi]$. Le nombre d'instances de chaque modèle de symbole varie entre 0 et 11.

22. Cette base contient 250 symboles de la base **GREC** ayant subi à la fois des transformations d'échelle (le facteur d'échelle varie entre 0.5 et 1.5). et des rotation ($\alpha \in [0, \pi]$). Le nombre d'instances de chaque modèle varie entre 0 et 10.

23. Cette base contient 250 symboles de la base **GREC**. Le nombre d'occurrences de chaque modèle est 5 occurrences pour chaque modèle.

24. Cette base contient 75 symboles de la base **GREC** avec une forte dégradation et 20 modèles de **GREC**. Le nombre d'occurrences de chaque modèle varie entre 1 et 11.

25. Cette base contient 75 symboles de la base **GREC** avec une forte dégradation et 15 modèles de **GREC**. Le nombre d'occurrences de chaque modèle varie entre 1 et 11.

base **GREC**_{dist1}²⁶.

Afin d'évaluer les différents détecteurs de points d'intérêt, notre cadre de travail expérimental a consisté en deux scénarios. Tout d'abord, nous les avons testés en termes du nombre de points détectés ainsi qu'en terme du temps nécessaire pour leur extraction. Deuxièmement, nous les avons évalués en les intégrant dans le cadre d'une approche de détection de symboles dans leur contexte.

2.4.1 Comparaison des Détecteurs de Points d'Intérêt

La Table.2.1 donne les résultats comparatifs des différents détecteurs décrits ci-dessus. Nos expérimentations menées sur les données de la Table.2.1, nous ont permis d'étudier les détecteurs des points d'intérêt indépendamment des descripteurs. Notre expérimentation se focalise sur l'évaluation de la complexité des détecteurs, à la fois à l'égard du nombre de points détectés et du temps nécessaire pour l'extraction de ces points, ce qui a conduit à des conclusions triviales sur les performances des différentes méthodes. La plupart des détecteurs fournissent un nombre raisonnable de points d'intérêt qui varie avec la forme du symbole traité. Les différents détecteurs s'avèrent stables en termes du nombre de points d'intérêt. Le nombre de points détectés par [Lowe1999] varie de 14 à environ 170 points ce qui reste cohérent aux autres détecteurs à savoir le détecteur de [Harris and Stephens1988] qui extrait en moyenne 50 points, le détecteur SURF extrait en moyenne 39 points et le détecteur aléatoire extrait en moyenne 71 points d'intérêt. Le comportement des détecteurs semble proche malgré que nous constatons que le détecteur *pseudo-aléatoire* se démarque en terme du temps d'extraction des points qui s'avère assez rapide en comparaison aux autres détecteurs. Cette différence nous semble logique car la méthode aléatoire adoptée n'impose aucune contrainte sur les primitives extraites. Le détecteur de [Harris and Stephens1988] et SURF sont aussi raisonnablement stables, et montrent un temps d'extraction assez rapide par rapport au [Lowe1999]. Les détecteurs de points d'intérêt montrent des taux de

	Fig.2.14(a)	Fig.2.14(b)	Fig.2.14(c)	Fig.2.14(d)	Fig.2.14(e)	Fig.2.14(f)	Fig.2.14(g)	Moyenne
NB _{p_i} (<i>Harris</i> [Harris and Stephens1988])	76	31	41	63	121	29	36	56
Temps _{p_i} (<i>Harris</i> [Harris and Stephens1988])	0.87	0.3	0.18	0.64	1.1	0.1	0.43	0.51
NB _{p_i} (<i>DoG</i> [Lowe1999])	54	19	14	33	175	19	22	48
Temps _{p_i} (<i>DoG</i> [Lowe1999])	1.76	1.39	1.51	1.7	1.84	1.34	1.44	1.56
NB _{p_i} (<i>SURF</i> [Bay et al.2006])	34	17	14	37	78	46	51	39
Temps _{p_i} (<i>SURF</i> [Bay et al.2006])	1.05	1.01	1.34	1.1	1.03	0.9	1.32	1.10
NB _{p_i} (<i>pseudo-aléatoire</i>)	78	45	53	67	150	42	64	71
Temps _{p_i} (<i>pseudo-aléatoire</i>)	0.77	0.3	0.42	0.43	0.5	0.3	0.34	0.43

TABLE 2.1 – Évaluation de la complexité des détecteurs de points d'intérêt de l'état de l'art : *Harris* [Lowe1999], *DoG* [Lowe1999], *SURF* [Bay et al.2006] et le détecteur *pseudo-aléatoire*. Dans cette expérimentation nous avons choisi d'évaluer les descripteurs en termes du (a) : NB_{p_i} : nombre des points d'intérêt détectés, Temps_{p_i} : le temps nécessaire pour l'extraction des points p_i.

reconnaissance très semblables comme c'est illustré dans la Figure.2.13. Le taux de reconnaissance de chaque détecteur augmente avec le nombre de points détectés. Ils montrent des taux de reconnaissance assez élevés et assez proches. Ce comportement nous emmène à

26. Cette base contient 75 symboles de la base **GREC** avec une distorsion et 15 modèles de **GREC**. Le nombre d'occurrences de chaque modèle varie entre 1 et 10.

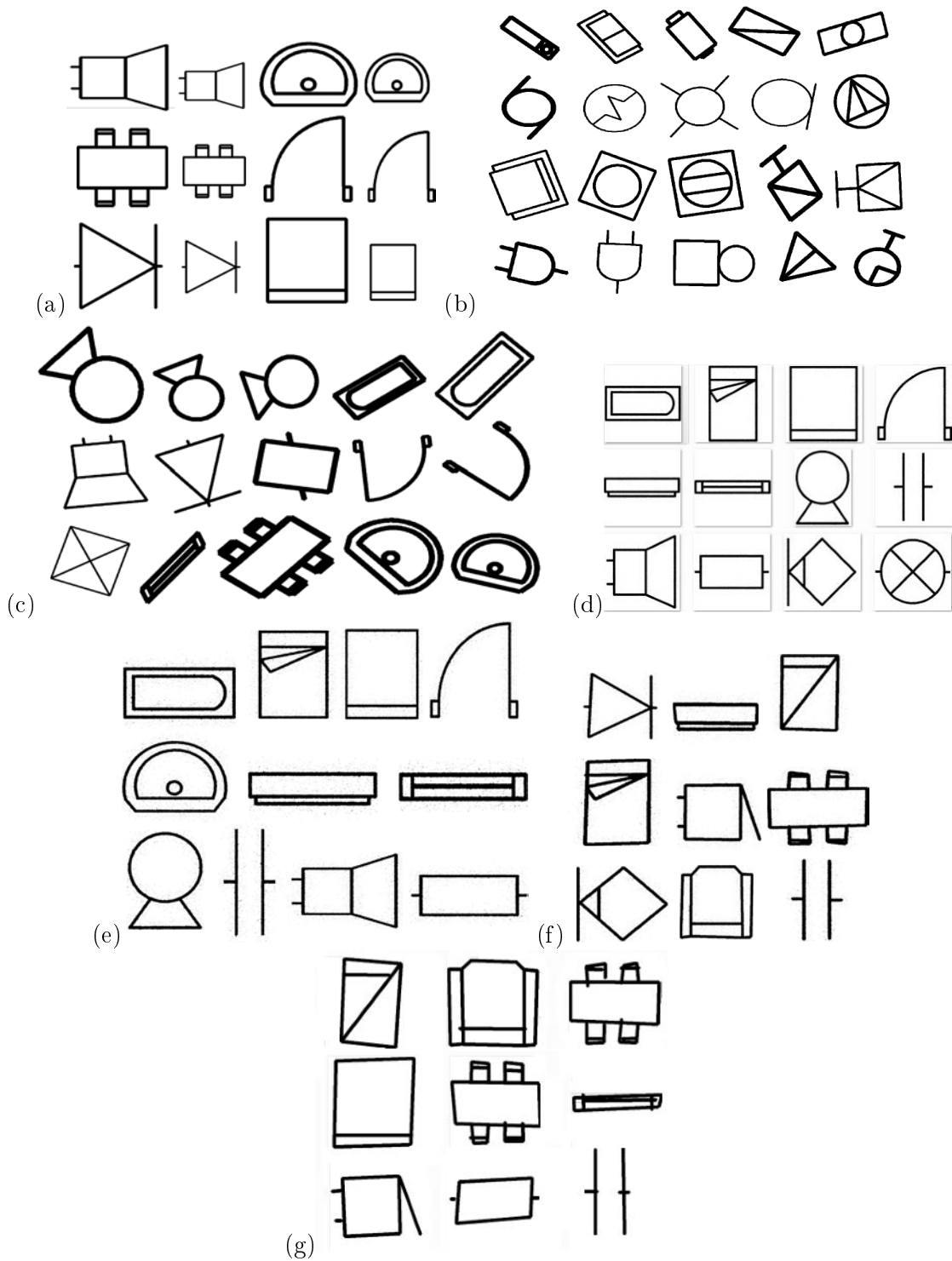


FIGURE 2.12 – Quelques exemples de symboles dans les bases de test (a) Symboles ayant subi des transformations d'échelle $GREC_{scale}$, (b) Symboles ayant subi des rotations $GREC_{rot}$, (c) Symboles ayant subi des transformations d'échelle et de rotation $GREC_{scale+rot}$, (d) Symboles dégradés niveau 1 de la base $GREC_{degr1}$, (e) Symboles dégradés niveau 2 de la base $GREC_{degr2}$, (f) Symboles détériorés niveau 1 de la base $GREC_{dist1}$, (g) Symboles détériorés niveau 2 de la base $GREC_{dist2}$.

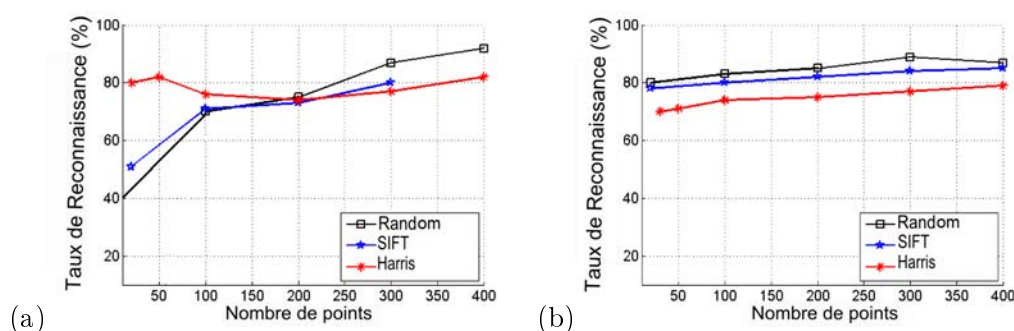


FIGURE 2.13 – Évaluation des détecteurs en fonction de l’impact du nombre de points d’intérêt sur le taux de reconnaissance de symboles. Dans cette expérimentation, nous avons utilisé le classifieur 7-ppv en vue de reconnaître les 7 occurrences semblables à la requête.

conclure que les résultats fournis par l’approche de [Harris and Stephens1988], [Lowe1999] et le détecteur aléatoire sont très semblables, ce qui est prévu, et approuvé par l’évaluation de la Table.2.1. Nous pouvons conclure que nous pouvons pas se pronocer pour telle ou telle méthode car les différences des performances, si elles existent, sont moindres. Il est important de noter que cette étude des détecteurs de points vise à évaluer l’impact de l’étape d’extraction de primitives avant son intégration dans la chaîne de traitement du système de détection de symboles dans leur contexte. Le succès d’un système de reconnaissance ou de localisation dépendra du descripteur choisi. Cette étude montre une claire ressemblance entre les performances de chaque détecteur ce qui rend difficile la prononciation pour telle ou telle méthode. La vraie évaluation d’un détecteur ne pourra être effective qu’après son intégration dans le cadre d’un système de localisation de symboles par exemple. D’après ces expérimentations, nous pouvons aussi conclure que ces détecteurs ayant un comportement similaire et proche de l’oracle, sont adaptés pour la segmentation des symboles.

2.5 Discussion

Un constat global que nous pouvons faire après cette évaluation est qu’il est difficile de se pronocer pour telle ou telle méthode d’extraction de primitives car la technique d’extraction et de représentation des primitives dépend du contexte de son application ou encore des configurations des symboles traités. Nous concluons alors qu’il n’y a pas une approche prédominante qui montre clairement sa robustesse car tout est relatif aux paramètres des expérimentations et aux types d’applications mises en œuvre.

2.6 Conclusion

Dans cette partie, nous avons essayé d’évaluer les détecteurs de points d’intérêt à savoir le détecteur aléatoire, le détecteur de Harris, le détecteur DoG adopté dans [Nguyen *et al.*2009] et SURF. Pour une évaluation objective, nous avons choisi de se référer à un même descripteur pour les trois premiers détecteur qui est le descripteur SIFT. Nous avons intégré ces descripteurs dans le cadre d’une chaîne de localisation en vue de répondre à la question suivante : *Ces descripteurs sont ils viables aux documents graphiques ?*

2.7 Impact de l'étape de représentation sur les résultats de la localisation

Comme nous l'avons largement détaillé auparavant, les approches de localisation de symboles traitent deux processus à savoir la reconnaissance et la segmentation en même temps. De telles capacités doivent être prises en compte par le processus d'évaluation. La méthode de segmentation a un impact direct sur les performances du système d'où nous avons choisi d'évaluer l'impact de la méthode de description de primitives sur les performances du système de localisation. Le problème majeur des systèmes de localisation est la proximité des symboles sur les plans architecturaux²⁷.

Pour mener ces expérimentations, nous avons utilisé des images synthétiques vu l'absence de bases de documents graphiques et leur vérité terrain ce qui peut introduire un biais de performances. Le contexte de notre travail pourra être caractérisé de supervisé car nous possédons une connaissance *a priori* des symboles à localiser. Nous avons choisi d'inté-

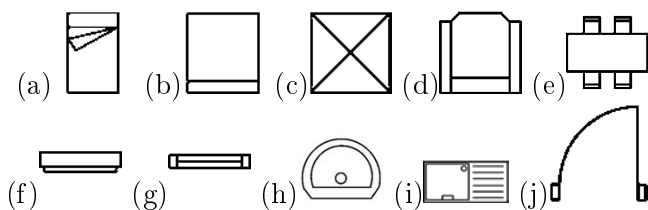


FIGURE 2.14 – Exemples de symboles requêtes interrogés lors du processus de localisation.

grer le détecteur aléatoire dans une chaîne de localisation et évaluer diverses approches de localisation pixelaires qui se basent sur le calcul des descripteurs photométriques au niveau des documents et des symboles graphiques. Pour mettre en place cette première série d'évaluations, nous avons utilisé les descripteurs SIFTs inspirés de l'approche de [Bagdanov *et al.*2007], SURF, le Contexte de Forme au niveau des Points d'Intérêt CFPI [Nguyen *et al.*2009] ainsi que notre approche basée sur le calcul des SIFTs au voisinage stable des points choisis aléatoirement. Dans ce qui suit, nous présentons une évaluation de quelques approches pixelaires de localisation de symboles dans leur contexte en vue de tester l'impact de l'étape de description des primitives sur les taux de détections.

2.7.1 Méthode a

L'approche de détection de symboles en se basant sur l'appariement des descripteurs SIFTs se base sur : (a) l'extraction de n_i points de Harris à partir d'un symbole \mathcal{S}_i , (b) Un vecteur SIFT \mathcal{P}_i est calculé pour chaque point d'intérêt n_i de \mathcal{S}_i , (c) Un symbole \mathcal{S}_i est alors décrit comme suit :

$$\mathcal{S}_i = \{x_k, y_k, \theta_k, \mathcal{P}_k\}, k \in \{1, n_i\} \quad (2.27)$$

avec x_k et y_k sont les coordonnées du point d'intérêt k , θ_k est l'échelle de détection du point d'intérêt k et \mathcal{P}_k est le descripteur SIFT associé au point k . Un appariement entre

27. En raison de l'absence de base de données de référence contenant des images de documents et la vérité terrain associée, nous avons recours à quelques images synthétiques proposées par [Delalandre *et al.*2007]. Certes, l'évaluation des systèmes de localisation en se basant uniquement sur ce type de documents engendre un biais aux performances de détection obtenues mais dans cette partie nous nous concentrons sur l'évaluation de l'étape de segmentation proposée dans la chaîne de traitements de reconnaissance ou de localisation.

les points du symbole requête \mathcal{S}_i et un document \mathcal{D}_i est effectué en utilisant 2-ppv :

$$N_1(\mathcal{S}_i, \mathcal{D}_j^q) = \min_k(\mathcal{P}_j - \mathcal{P}_k) \quad (2.28)$$

$$N_2(\mathcal{S}_i, \mathcal{D}_j^q) = \min_{k \neq N_1(\mathcal{S}_i, \mathcal{D}_j^q)}(\mathcal{P}_j - \mathcal{P}_k) \quad (2.29)$$

Un score d'appariement est ensuite obtenu en calculant le ratio entre les 2 voisins N_1 et N_2 :

$$\mathcal{M}(\mathcal{S}_i, \mathcal{D}_j^q) = \frac{N_1(\mathcal{S}_i, \mathcal{D}_j^q)}{N_2(\mathcal{S}_i, \mathcal{D}_j^q)} \quad (2.30)$$

Si le score d'appariement est inférieur à un certain seuil m alors le point pris en compte pourra être retenu. Dans nos expérimentations, ce seuil m est égale à 0.71 ce qui permet de garantir un nombre réduit de faux positifs. Dans la Figure.2.15, nous présentons des exemples d'appariement entre un symbole requête et un plan architectural assurant une détection de la requête dans le document.

2.7.2 Méthode b

L'approche de détection de symboles en se basant sur l'appariement des descripteurs SURFs se basent sur : (a) l'extraction de n_i points SURF à partir d'un symbole \mathcal{S}_i , (b) Un vecteur SURF \mathcal{X}_i est calculé pour chaque point d'intérêt n_i de \mathcal{S}_i , (c) les points SURFs d'un document graphique sont également calculés et chacun d'entre eux correspond au point SURF le plus similaire dans le symbole modèle. Donc une paire de points SURF correctement appariés permet de déterminer la position, l'échelle et l'orientation de la cible dans le document. Cependant, la plupart des paires ne sont pas correctes, un vote est utilisé pour retenir les paires de points qui sont correctement appariés. Dans les Figures.2.16 et .2.17, nous présentons des exemples d'appariements entre un symbole requête, un plan architectural et un document électronique assurant une détection de la requête dans le document.

2.7.3 Méthode c

Dans [Nguyen *et al.*2009], la localisation des symboles dans des documents graphiques se basent sur un appariement d'un ensemble de descripteurs CFPIs d'un symbole requête (Contexte de Forme au niveau des Points d'Intérêt) avec les mots visuels calculés sur une base de documents graphiques. Un CFPI est représenté par un ensemble de mots avec différents degrés de confiance. La structure du fichier inverse permet de déterminer l'ensemble des documents contenant les mêmes mots visuels du symbole requête. Une fois le document est retourné, des régions candidates sont sélectionnées et une structure du modèle vectoriel permet de retenir uniquement les régions susceptibles de contenir le symbole requête. Un ensemble de résultats est présenté dans la Figure.2.18.

2.7.4 Méthode d

Dans cette approche, un ensemble des vecteurs SIFTs sont calculés au niveau des voisinages stables des points \mathcal{P}_i du document détectés aléatoirement. Il s'agit de l'approche *semi-aléatoire* décrite dans les sections précédentes. Nous présentons quelques résultats de localisation obtenus par notre approche dans la Figure.2.19.

Dans ce qui suit, nous présentons un ensemble de mesures pour évaluer les performances

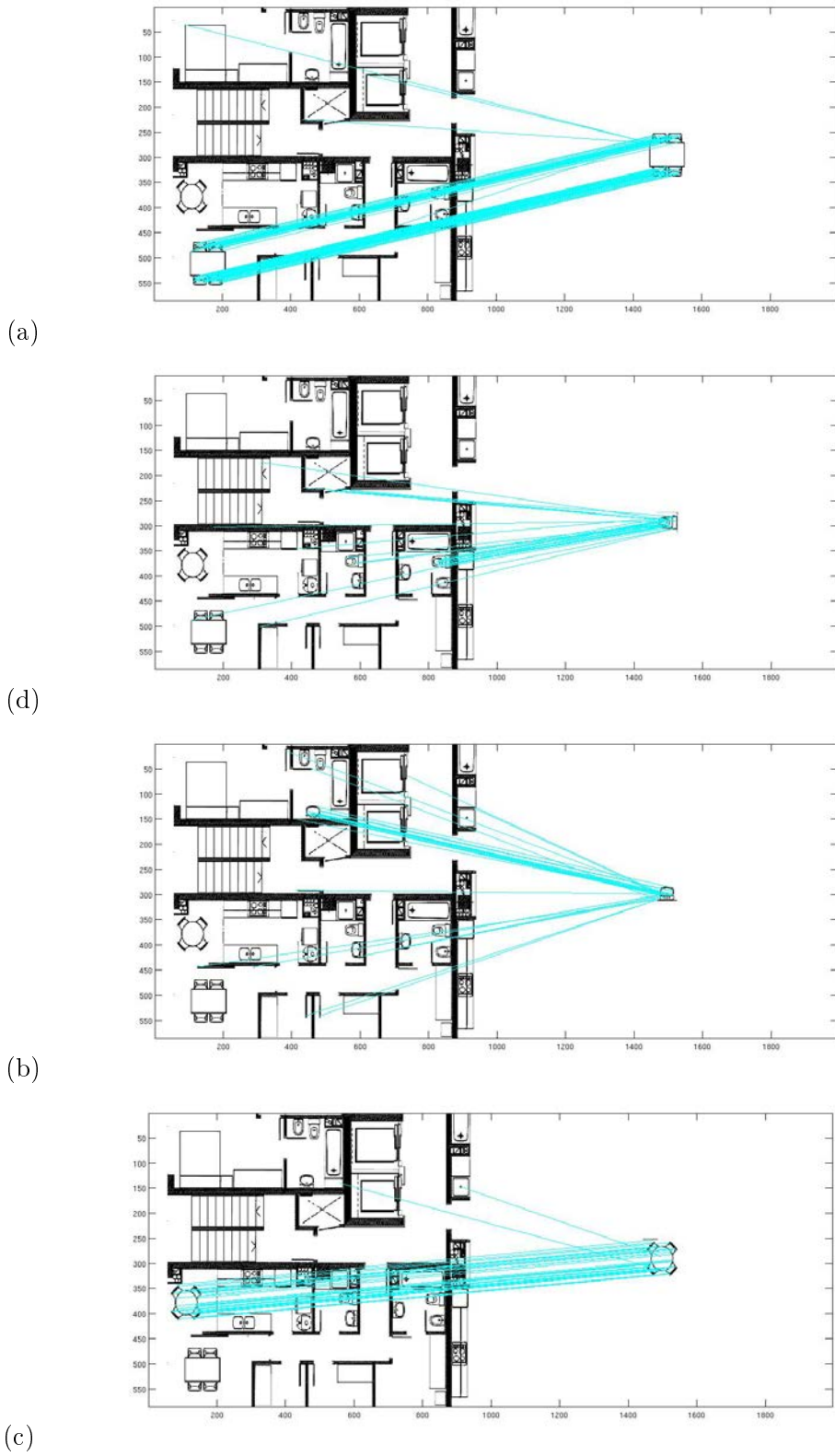


FIGURE 2.15 – Localisation des symboles requêtes dans des plans architecturaux en se basant sur le descripteur SIFT. Un appariement est réalisé entre le symbole modèle et le plan architectural.

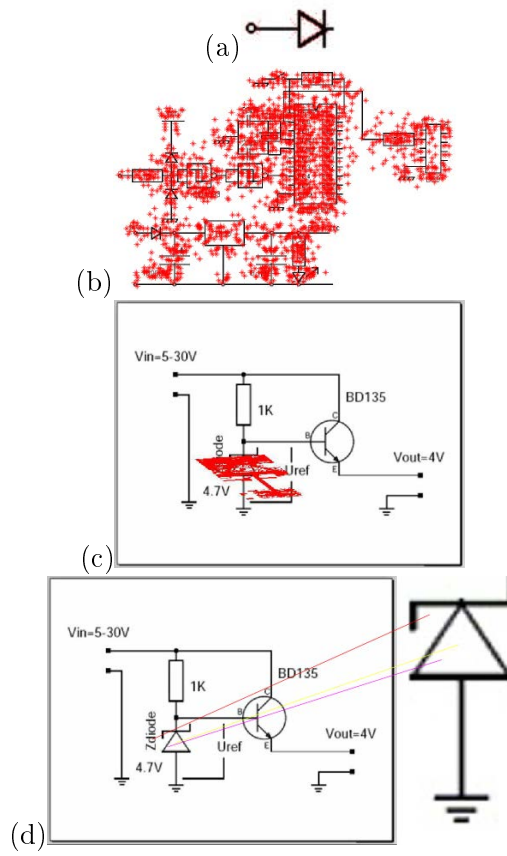


FIGURE 2.16 – Localisation des symboles requêtes dans des plans architecturaux en se basant sur le descripteur SURF. Un appariement est réalisé entre le symbole modèle et le dessin électronique.

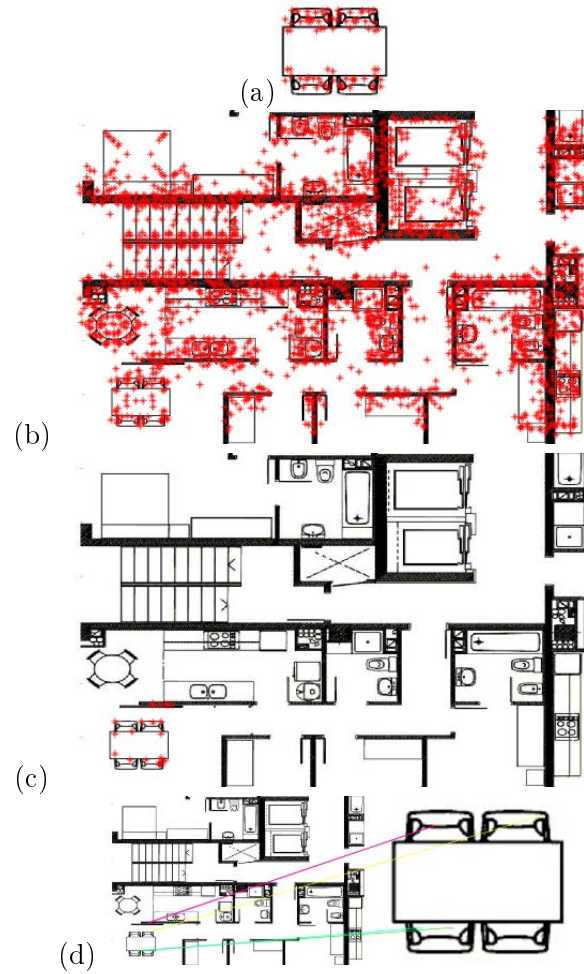


FIGURE 2.17 – Localisation de symboles requêtes dans des plans architecturaux en se basant sur le descripteur SURF. Un appariement est trouvé entre le symbole modèle et le plan architectural.

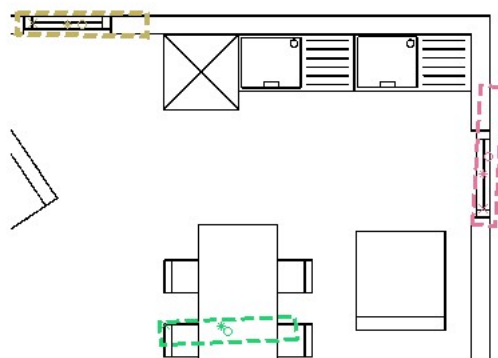


FIGURE 2.18 – Résultats de localisation de la requête.2.14(j) dans un plan architectural de la base [Delalandre *et al.*2007] (résultat extrait de [Nguyen *et al.*2009]).

des systèmes de localisation de symboles en terme de capacités de précision de localisation. Bien qu'il existe de nombreuses études sur le problème de détection de symboles dans leur contexte, il n'y a à notre connaissance aucune évaluation globale pour laquelle la même base de données est utilisée par toutes les approches existantes. Dans la littérature, les seules bases de données utilisées comme vérité de terrain sont les bases du projet SE-SYD²⁸ [Delalandre *et al.*2010]. Dans ces bases de données, chaque symbole est incorporé dans une région rectangulaire. En vue de se positionner par rapport aux travaux de l'état de l'art, notre première expérience consiste à comparer notre segmentation aléatoire avec les trois approches décrites ci-dessus.

Les expérimentations ont été réalisées en utilisant 82 documents (63 plans architecturaux et 19 documents électroniques). Les performances des différents systèmes sont mesurées en terme des Taux de Faux Positifs TFP²⁹ et des Taux de Vrais Positifs TVP³⁰ obtenus lors d'une interrogation d'une base de documents en vue de détecter un symbole modèle. Précisons que pour ces évaluations, la taille du vocabulaire que nous avons choisi est $|\mathcal{VOC}|=120$.

Analysons tout d'abord quelques résultats qualitatifs. Étant donné un symbole requête, le processus de localisation consiste à retrouver une liste de régions du document graphique susceptibles de contenir des occurrences du symbole requête. Les résultats obtenus permettent d'affirmer que chaque descripteur possède des avantages ainsi que des inconvénients. Les expérimentations ont montré que tous les symboles se trouvent dans les documents et, évidemment, certaines régions de faux positifs apparaissent. Le descripteur SIFT montre de bons résultats pour le Taux de Vrais Positifs meilleurs que le descripteur SURF et l'approche de [Nguyen *et al.*2009] qui se base sur le Contexte de Forme calculé au niveau des Points d'Intérêt. La méthode 4 montre de bons Taux de Vrais Positifs ce qui explique la bonne localisation des symboles dans les documents. Le descripteur

28. <http://mathieu.delalandre.free.fr/projects/sesyd/index.html>

29. Le TFP mesure la probabilité pour qu'un symbole n'apparaît pas dans un document graphique.

30. Le TVP mesure la probabilité de détecter le symbole requête dans le document.



FIGURE 2.19 – Localisation de quelques symboles requêtes dans des documents graphiques : (a) interrogation de la requête de la Figure.2.14(i), (b) interrogation de la requête de la Figure.2.14(b), (c) interrogation de la requête de la Figure.2.14(g), (d) interrogation de la requête de la Figure.2.14(j) et (e) l'interface de "Symbol Spotting" mise en place.

SIFT s'avère très performant dans le cadre d'un système de localisation dont l'objectif principal consiste à détecter correctement des symboles dans les documents. Le TFP présenté par le descripteur SIFT et SURF est un peu élevé en comparaison à l'approche de [Nguyen *et al.*2009] et l'approche basée sur le détecteur aléatoire ce qui explique l'apparition de quelques fausses détections lors de l'interrogation des documents ne contenant pas le symbole requête. La réponse du système dans ce cas devrait être le rejet. Notre approche et l'approche de [Nguyen *et al.*2009] sont plus adaptées au système dont l'objectif est de minimiser le nombre des faux positifs lors d'un processus de localisation. Bien que ces résultats sont prometteurs, il y aura toujours l'apparition de quelques faux positifs lors du processus de localisation. Les temps de calcul pour les descripteurs SIFT et SURF sont plus élevés en comparaison à l'approche de [Nguyen *et al.*2009], ce qui pourra se justifier par le nombre élevé des composantes des vecteurs caractéristiques calculés. Cette évaluation des détecteurs les plus utilisés dans la communauté de vision par ordinateur montre clairement qu'il n'y a pas d'approche prédominante mais qu'en fonction du contexte abordé et du type de l'application proposée, certains types de détecteurs sont plus pertinents que d'autres. Dans la Table.2.3, nous présentons un ensemble détaillé de mesures pour évaluer

TABLE 2.2 – Mesures de qualité des méthodes de détection de symboles dans les documents graphiques. Les critères d'évaluation sont : Taux de Vrais Positifs (TVP), Taux de Faux Positifs (TFP) et Temps (secs) désignant le temps nécessaire pour la localisation du symbole requête dans le document graphique.

	TVP	TFP	Temps (secs)
Méthode a	0.47	0.42	1.65
	0.86	0.35	1.96
	0.9	0.6	2.03
Méthode b	0.5	0.16	2.01
	0.72	0.3	2.43
	0.8	0.32	3.02
Méthode c	0.52	0.23	1.98
	0.72	0.27	1.84
	0.8	0.41	2.01
Méthode d	0.55	0.19	1.67
	0.72	0.25	2.5
	0.89	0.32	2.67

les performances des systèmes de localisation décrits auparavant à savoir l'approche basée sur le détecteur aléatoire (colonne (a)), l'approche de [Nguyen *et al.*2009] (colonne (b)), la méthode 1 (colonne (c)) et la méthode 2 (colonne (d)). Comme nous pouvons le voir, il n'y a pas des différences significatives entre les résultats sauf pour quelques symboles. Les taux de précision et de rappel sont bons pour les deux premières approches. Pour le deuxième symbole (Figure.2.14(f)), dans [Nguyen *et al.*2009] ce symbole est mal localisé. Ces problèmes de localisation sont expliqués par le fait que ces symboles ont peu de points discriminants. Ces symboles sont donc plus sensibles à des connexions avec des composants externes, et donc ils sont plus difficiles à localiser. Dans le cas de symboles de la Figure.2.14(f) et de la Figure.2.14(j), la majorité des points d'intérêt sont situés près des connexions externes. Les informations représentées par ces points sont moins robustes, car ils contiennent des informations externes. La méthode *a* montre de très bonnes valeurs de vrai positifs ce qui renforce l'hypothèse de l'amélioration du pouvoir discriminant du descripteur choisi. Le système de localisation s'avère performant lors de la récupération de la plupart des

symboles requêtes dans le document, en contre partie il y a une possibilité d'apparition de faux positifs ce qui pourra se justifier par les structures d'indexation utilisées qui sont respectivement le modèle vectoriel et le fichier inverse. Pour les méthodes (c) et (d) il est important de noter que le choix du seuil σ , utilisé pour l'élimination des points instables, a une conséquence sur le nombre de points d'intérêt ce qui pourra justifier les nombreux faux positifs détectés lors du processus de localisation. En effet, si le seuil s'éloigne de 1, la mise en correspondance devient plus sélective et par la suite moins de points sont retenus. Le problème consiste alors à trouver le meilleur compromis entre le taux de bons appariements et le nombre de points appariés. Certains résultats de l'approche (a) sont insuffisants en comparaison avec celle de [Nguyen *et al.*2009], par exemple lors de l'interrogation du symbole de la Figure.2.14(i), des faux positifs apparaissent pendant le processus de localisation en se basant sur notre approche ce qui peut être causé par le nombre initial de points choisis aléatoirement ou en raison de la taille du dictionnaire utilisé comme une entrée pour le fichier inverse. L'approche (a) est appropriée si nous privilégions la précision. Nous pouvons conclure que les descripteurs utilisés fournissent globalement tous de bons taux de détection et s'avèrent adéquats au contexte de la détection des symboles en contexte. Ce constat nous emmène à conclure que les descripteurs photométriques sont adéquats aux documents graphiques vu qu'ils présentent des performances semblables aux autres détecteurs et plus spécialement le détecteur aléatoire qui sert d'oracle.

	Fig.2.14(a)				Fig.2.14(f)				Fig.2.14(h)				Fig.2.14(i)				Fig.2.14(j)					
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)		
Doc-0	0/0/0	0/0/0	0/0/0	0/0/0	2/3/3	0/0/3	1/2/3	0/1/3	4/4/4	4/4/4	4/4/4	3/3/4	2/4/4	1/1/1	0/0/1	0/1/1	1/1/1	1/1/1	1/1/1	1/2/1	0/1/1	0/1/1
Doc-1	3/3/3	3/3/3	2/0/3	1/2/3	0/0/0	0/0/0	0/1/0	0/1/0	4/4/4	4/4/4	4/4/4	4/2/4	3/3/4	1/1/1	1/1/1	0/1/1	1/1/1	1/1/1	1/1/1	1/0/1	0/1/1	0/1/1
Doc-2	1/1/1	1/1/1	1/1/1	0/1/1	0/1/2	0/0/2	1/1/2	0/1/2	4/4/4	4/4/4	4/4/4	3/3/4	2/2/4	0/0/0	0/0/0	0/0/0	1/1/1	1/2/1	1/1/1	1/1/1	1/1/1	1/1/1
Doc-3	1/2/1	1/1/1	1/3/1	0/2/1	1/2/2	0/0/2	1/1/2	1/0/2	4/4/4	4/4/4	4/4/4	4/2/4	2/1/4	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1	0/0/1	0/0/1
Doc-4	1/2/2	2/2/2	1/1/2	1/2/2	0/2/1	0/0/1	1/2/1	1/0/1	4/4/4	4/4/4	4/4/4	4/4/4	4/2/4	0/0/0	0/0/0	0/0/0	1/1/1	1/1/1	1/1/1	1/0/1	1/1/1	1/1/1
Précision	6/7	7/7	5/7	2/7	3/8	-	4/8	2/8	20/20	20/20	20/20	18/20	13/20	3/4	3/3	2/4	3/3	5/5	5/5	5/5	2/5	2/5
Rappel	7/7	7/7	5/7	7/7	8/8	0/8	7/8	3/8	20/20	20/20	20/20	14/20	12/20	3/3	3/3	3/3	5/5	5/5	5/5	4/5	4/5	4/5

TABLE 2.3 – Résultats de localisation de symboles dans des document de la base SESYD [Delalandre et al.2007] (a) notre approche (b) l'approche de [Nguyen et al.2009] (Noter que les résultats de Nguyen sont extraits de [Nguyen et al.2009], (c) l'approche basée sur SIFT, (d) l'approche basée sur SURF. La notation $x/y/z$ représente : le nombre de bonnes détections de symboles (x), le nombre total des symboles retournés (y) et le nombre des occurrences de symbole dans le document graphique (z).

2.8 Conclusion

Dans ce chapitre, nous avons proposé une évaluation de l'impact de l'étape de représentation des primitives des documents graphiques sur les performances du système de localisation. Nous avons constaté que le choix d'une bonne technique d'extraction de primitives et leurs bonnes descriptions aboutit souvent à de très bonnes performances mais il est toujours difficile de se prononcer pour telle ou telle technique. Aucune méthode ne peut se démarquer du lot car elle dépend du contexte de son application. C'est pour cette raison que le choix d'une bonne représentation des symboles et des documents graphiques demeure un problème ouvert. Les résultats fournis par les différentes techniques d'extraction et de représentation sont proches et les différences ne sont pas significatives. Une conclusion que nous pouvons déduire est que les descripteurs jugés photométriques sont viables aux documents graphiques. La contribution de ce travail réside spécialement à proposer une évaluation de l'impact de l'étape de représentation des documents graphiques sur les performances de détection en ayant recours à une méthode référence en vue d'une évaluation réaliste. Les approches pixelaires de localisation de symboles dans les documents graphiques semblent bien adaptées au problème de localisation ce qui nous a conduit à poursuivre dans l'étude de ce type d'approches mais cette fois-ci, nous combinons l'approche pixelaire avec l'approche "structurelle" en introduisant les graphes permettant une organisation hiérarchique des caractéristiques de l'image au sein d'un treillis de concepts. En terme de perspectives, nous approfondissons actuellement notre approche, et souhaitons adapter la structure du treillis de concepts dans le cadre de la localisation (chapitre 3).

3

Localisation de Symboles dans les Documents Graphiques en se basant sur le Treillis de Galois

" La plupart des concepts sont eux-mêmes des théories partielles en ce qu'ils incarnent des explications sur les relations entre les constituants, de leurs origines et de leurs relations avec les autres groupes de fonctions".

Keil

Sommaire

3.1	Introduction	183
3.2	Approche Proposée	184
3.2.1	Description de l'approche	185
3.2.2	Extraction des primitives	187
3.2.3	Construction du Vocabulaire Visuel	188
3.2.4	Sélection des primitives	188
3.2.5	Détection de symboles dans les documents graphiques	189
3.3	Résultats Expérimentaux	191
3.3.1	Impact de l'étape de sélection des attributs et des concepts sur la taille du treillis	194
3.4	Conclusion	194

3.1 Introduction

Dans le chapitre précédent, nous avons présenté une évaluation de quelques approches de segmentation en points d'intérêt et constaté que les approches par intensité sont viables pour les processus de reconnaissance et de localisation des documents graphiques. Comme nous l'avons largement détaillé dans l'introduction de ce manuscrit, notre objectif est de bénéficier des capacités de la structure du treillis de concepts en l'intégrant dans diverses applications. D'après l'évaluation du système de localisation présenté dans le chapitre précédent et qui se base sur les structures du modèle vectoriel et du fichier inverse, nous avons

pu constater que les résultats dépendent de ces structures et plus spécialement de la façon de modélisation des mots visuels dans le fichier inverse. L'apparition de quelques faux positifs est due à l'absence de relations spatiales entre les entrées du fichier inverse. Pour cette raison, nous avons eu l'idée de modéliser ces relations en se basant sur l'organisation des mots visuels dans la structure du treillis. Nous avons vu que la structure du treillis permet d'organiser des données dans une hiérarchie respectant une relation de *{généralisation/spécialisation}* entre les concepts.

À l'égard des raisons évoquées ci-dessus, notre objectif dans ce travail consiste à intégrer la structure du treillis de concepts dans un système de détection de symboles dans leur contexte. Notre choix se justifie par les bons résultats prouvés par le treillis de Galois dans le cadre de la reconnaissance de symboles graphiques comme nous l'avons présenté dans la deuxième partie de ce manuscrit. Pour cette raison, un treillis de concepts est construit pour chaque document graphique en se basant sur l'ensemble des mots visuels et les points d'intérêt qui leur sont associés. Nous nous sommes orientés vers la structure du treillis de concepts pour ses nombreux avantages outre l'originalité de l'approche. Premièrement, la représentation des images à l'aide de cette structure permet de mettre en évidence la relation entre *{Primitives, Attributs}* ce qui permet d'apporter une modélisation riche en information au lieu de se contenter de l'aspect numérique apporté par l'utilisation des descripteurs statistiques. Deuxièmement, le treillis de concepts représente une structure de données très flexible et possède une capacité importante d'abstraction. Un autre avantage majeur de l'utilisation du treillis est cet aspect relationnel qui est mis en avant dans la représentation des objets. Dans notre approche, nous avons recours à la modélisation des relations entre les mots visuels et les points d'intérêt qui leur sont associés en utilisant la structure du treillis de concepts. Nous sommes partis de l'hypothèse qu'un mot visuel peut être associé à un ou plusieurs points d'intérêt d'où la hiérarchisation des mots visuels dans le treillis avec leurs points d'intérêt tout en bénéficiant de l'aspect sémantique offert par un tel outil. Nous avons choisi cette alliance intéressante entre les mots visuels caractérisants les documents graphiques et les points d'intérêt qui leur sont associés. La localisation de symboles requêtes dans les documents graphiques se base sur une étape de reconnaissance et une étape d'extraction des zones du document susceptibles de contenir des occurrences similaires à la requête.

Dans ce chapitre, nous allons détailler l'approche proposée pour localiser les symboles en se basant sur la structure du treillis de concepts. Dans une première section, nous présentons une description de notre approche de localisation en détaillant les différentes étapes de la chaîne de traitement présentée à savoir l'étape de description de primitives extraites à partir des images de documents graphiques, la construction du vocabulaire visuel, la sélection des primitives en se basant sur le treillis de Galois et la technique adoptée pour la détection de symboles dans les documents. Les résultats expérimentaux seront étalés dans la dernière section, suivie d'une discussion et d'une conclusion.

3.2 Approche Proposée

Comme nous l'avons détaillé dans les chapitres précédents, le treillis de concepts qui s'inscrit dans le cadre de l'AFC fournit une compréhension intuitive de la généralisation et de spécialisation des relations entre les objets et leurs attributs. Bien que cette structure s'est avérée performante dans le cadre de la reconnaissance de symboles graphiques, la visualisation du treillis de concepts s'est avérée un problème car le nombre de concepts augmente

de manière significative avec le nombre d'objets et d'attributs. Interpréter le treillis grâce à une visualisation directe du diagramme de Hasse devient rapidement impossible et des représentations plus synthétiques sont nécessaires [Melo *et al.*2011b]. Il est vrai que dans notre approche, l'objectif n'est plus de visualiser le treillis mais d'obtenir une localisation précise d'un symbole requête, mais malgré cela dans ce travail, nous proposons des alternatives à la représentation du treillis traditionnel, d'une part en améliorant la lisibilité de treillis de concepts, d'autre part par la sélection des attributs en se basant sur ce dernier. La sélection des attributs ou des primitives présente de nombreux avantages : elle élimine tous les concepts vides ainsi que tous les attributs qui ne sont pas porteurs d'information pouvant engendrer des erreurs lors de l'étape de reconnaissance, et par la suite la hiérarchie qui en résulte est aussi plus facile à interpréter et à représenter. Une vue d'ensemble du jeu de données mettant en évidence les relations entre $\{Objets, Attributs\}$ est alors maintenue. Le processus de localisation des symboles dans leur contexte est décomposé en deux étapes : une étape de reconnaissance et une étape de localisation qui consiste à récupérer les endroits du document susceptibles de contenir les occurrences d'un symbole requête. Durant l'étape de la reconnaissance, une recherche des similarités entre les attributs du symbole modèle et les attributs représentant les documents graphiques est préconisée : souvent il s'agit de comparer les signatures extraites à partir de chaque objet graphique et de garder celles optimisant un certain critère de similarité sauf que ce calcul peut s'avérer exhaustif dans le cas où nous gérons une large base de données.

Dans notre cas, nous traitons des cas de symboles assez complexes qui ne sont pas isolés mais qui sont souvent connectés à d'autres éléments graphiques sur le document, pour cela nous nous sommes basés sur la structure du treillis de concepts qui a prouvé sa robustesse dans le cadre de la classification. La détection de symboles est ensuite assurée par une recherche des mots visuels caractérisant une requête et existant dans la structure du treillis modélisant les indices visuels du document graphique. Une projection des coordonnées de primitives communes dans un espace 2D est ensuite faite. Les mots visuels peuvent correspondre à plusieurs points d'intérêt dans le document ce qui nécessite une méthode permettant de trouver l'endroit du document le plus probable à contenir les occurrences les plus vraisemblables à la requête. Pour cette raison, une technique de *template matching* permet de mettre en correspondance les points d'intérêt du symbole requête avec ceux correspondant aux mots visuels retenus lors de la phase de la reconnaissance. Les différentes étapes de la chaîne de traitement proposée et présentée dans la Figure.3.1, seront détaillées dans les sections suivantes.

3.2.1 Description de l'approche

Paradoxalement, le constat que nous pouvons faire est que malgré le grand succès des structures du treillis de concepts dans le contexte de la fouille de données et dans le contexte de la reconnaissance, aucune approche à notre connaissance n'aborde le problème de localisation. Dans notre travail, nous avons opté pour cette structure pour les arguments évoqués ci-dessus. Cette structure arborescente permettant d'organiser les données d'une façon hiérarchique, semble adaptée pour naviguer dans le document afin de localiser les zones susceptibles de contenir les occurrences d'une requête.

Une originalité de notre approche réside dans la possibilité de modéliser les relations entre les mots et leurs points d'intérêt associés dans une même structure. Un concept du treillis est alors composé d'un ensemble de mots visuels et les points d'intérêt qui leur sont associés. Un point d'intérêt \mathcal{P}_i est caractérisé par le quadruplet : $(x_i, y_i, \rho_i, \sigma_i)$ où x_i et y_i sont

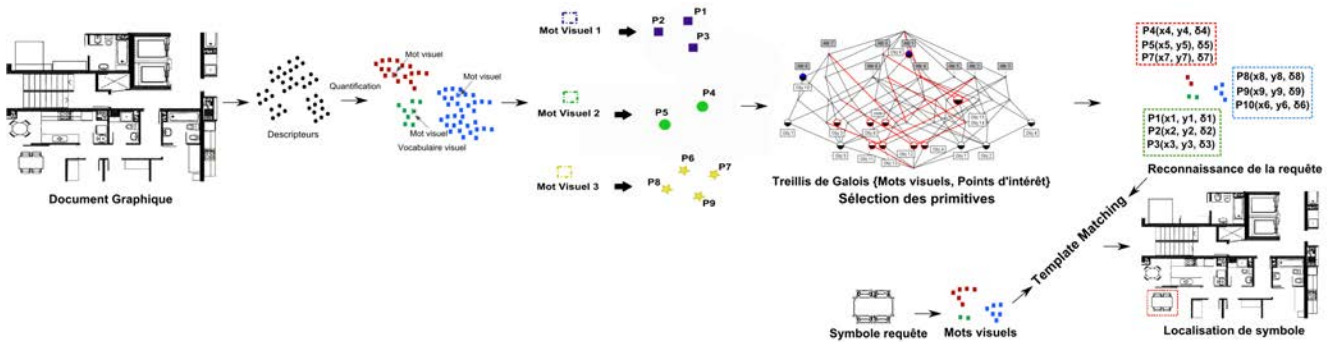


FIGURE 3.1 – Illustration des différentes étapes de la chaîne de traitement proposée dans le cadre de la conception d’un système de localisation de symboles dans les documents graphiques.

les coordonnées de \mathcal{P}_i , ρ_i son orientation et σ_i l’échelle de détection du point.

Étant donné la complexité de visualisation du treillis de concepts représentant un document, nous avons choisi de sélectionner les caractéristiques rendant plus facile l’interprétation ainsi que la représentation de données. De plus, cette représentation fournit toujours une vue d’ensemble et globale de l’ensemble de données tout en mettant en évidence les propriétés significatives du treillis. Pour sélectionner les attributs à partir du treillis de concepts, nous définissons un ensemble de critères de sélection de concepts, y compris le critère de stabilité de concepts en utilisant des indices fournis par l’Analyse Formelle de Concepts comme par exemple l’indice de confiance aussi bien que les caractéristiques topologiques du treillis [Melo *et al.*2011b].

Nous avons choisi de recourir à cette étape en vue de réduire la taille de ce dernier ainsi que d’éliminer les attributs non porteurs d’informations et pouvant apporter des erreurs lors de l’étape de la reconnaissance. Cette technique s’avère prometteuse car elle permet de sélectionner les caractéristiques les plus pertinentes ce qui aura un impact direct sur les résultats de localisation car l’ensemble des points projetés sera plus précis et le risque de tomber sur des fausses détections sera considérablement réduit. Nous avons choisi la réduction du nombre de caractéristiques de la structure du treillis de concepts afin de transformer le problème de localisation de symboles dans les documents graphiques à un simple problème de recherche d’un symbole requête dans un treillis représentant le document.

Dans cette partie, nous détaillons nos choix ainsi que la stratégie adoptée pour modéliser les primitives de symboles et de document à traiter. Comme indiqué dans la Figure.3.1, notre approche est composée des parties suivantes : (a) l’extraction des primitives ; (b) la quantification des descripteurs SIFTs pour construire le dictionnaire visuel ; (c) la sélection des primitives en utilisant la structure du treillis de Galois. Cette étape a pour objectif de réduire le nombre de concepts du treillis de Galois pour améliorer la lisibilité du graphe et la précision de la localisation, (d) l’étape suivante implique la recherche des concepts du treillis contenant les mots visuels du symbole requête.

Le problème est alors décomposé en deux problèmes :

- Un problème de reconnaissance : qui aboutit à un ensemble de mots visuels \mathcal{M} communs entre le symbole requête et le treillis modélisant le document.
- Un problème de détection : une projection des points d’intérêt correspondants aux mots visuels \mathcal{M} dans les concepts \mathcal{C} retenus à partir du treillis de concepts \mathcal{G} dans un espace

2D du document aboutit à définir des zones ou des régions susceptibles de contenir les occurrences du symbole. Partant de l'hypothèse qu'un document peut contenir tous les mots visuels du symbole requête sans que celui n'y soit présent et qu'un mot visuel peut correspondre à un ou plusieurs points d'intérêt dans le document, nous avons eu recours à une technique de *template matching* permettant de mettre en correspondance les points d'intérêt du symbole requête ainsi que ceux projetés sur le document suite à la phase de reconnaissance. Les régions du document correspondantes au symbole requête sont alors extraites et retenues comme les zones susceptibles de contenir les occurrences du symbole requête.

En se basant sur cette technique, nous avons pu réduire l'espace de recherche dans le document en un ensemble de régions d'intérêt les plus probables de contenir le symbole requête. Dans ce qui suit, nous allons détailler les étapes de la chaîne de localisation proposée.

3.2.2 Extraction des primitives

Nous avons effectué nos tests sur une base d'images du domaine architectural traitée dans le chapitre précédent. Les documents utilisés présentent une épaisseur de tracé homogène. Ces documents sont de dimension 500*500. Dans notre approche, des points de Harris [Harris and Stephens1988] sont détectés à partir de l'image du document. Une évaluation assez intéressante des détecteurs des points a été proposée par [Mikolajczyk and Schmid2004] ainsi que dans le chapitre précédent. Le détecteur proposé par [Harris and Stephens1988] s'est avéré robuste dans le contexte des configurations assez complexe. Nous optons donc pour ce détecteur : son principe est de trouver des singularités 2D du document. Notre choix pour un tel détecteur se justifie par l'obligation d'obtenir des coordonnées de points d'intérêt assez précises et certaines ainsi que l'échelle et l'orientation de détection de ces derniers pour pouvoir localiser les zones susceptibles de contenir le symbole requête.

Une fois que les points d'intérêt sont extraits, nous choisissons de sauvegarder leurs coordonnées spatiales afin de les intégrer par la suite dans le processus de localisation. Chaque point d'intérêt \mathcal{P}_i détecté est caractérisé par une matrice à 4 composantes : $(x_i, y_i, \rho_i, \sigma_i)$ où x_i et y_i sont les coordonnées de \mathcal{P}_i , ρ_i son orientation et σ_i l'échelle de détection du point. Ensuite, pour l'étape de description des primitives, nous avons opté pour un descripteur assez connu et qui a prouvé son efficacité et sa robustesse dans le traitement des problèmes de reconnaissance de symboles graphiques ainsi que dans le cadre des systèmes de localisation. Il s'agit du descripteur SIFT [Lowe1999] dont le principe réside dans l'extraction de vecteurs de caractéristiques locales. Ce descripteur a été testé dans le chapitre précédent, où notre constat était qu'il est bien adapté au contexte de détection des symboles spécialement quand l'objectif est de détecter le symbole et non de minimiser le nombre de fausses détections. À l'égard de ces raisons, et vu que notre objectif dans ce travail est de tester la structure du treillis dans le cadre de la localisation de symboles dans leur contexte, nous avons choisi ce descripteur car nous ne cherchons pas à minimiser le nombre des faux positifs qui peuvent apparaître lors de la localisation. Dans notre approche, des descripteurs locaux sont extraits à partir de chaque document, chaque région locale est décrite par un vecteur de dimension 128. Un document graphique \mathcal{D} est alors représenté comme suit :

$$\mathcal{D} \equiv \{\mathcal{F}_i | p_i \in \mathcal{P}\} \quad (3.1)$$

avec \mathcal{F}_i l'ensemble de descripteurs SIFTS calculés au voisinage des points de Harris \mathcal{P} avec $i \in [1, n]$.

3.2.3 Construction du Vocabulaire Visuel

Nous avons élaboré une approche basée sur la technique de sac-de-mots. Ce modèle est populairement utilisé dans le traitement automatique des langues et la recherche documentaire. L'idée de base de notre approche est qu'un ensemble de points de l'image du document graphique sont échantillonnés en utilisant un détecteur de points comme décrit ci-dessous et un vecteur de descripteurs visuels est ensuite évalué indépendamment. La distribution résultante de descripteurs est alors quantifiée en utilisant l'algorithme de *k-means*. Chaque cluster centre obtenu représente un mot visuel. La quantification engendre une réduction significative de la dimension du vocabulaire visuel tout en gardant le pouvoir discriminant du descripteur.

3.2.4 Sélection des primitives

La représentation de départ que nous utilisons ici est un treillis de concepts. Dans un tel treillis, chaque nœud correspond à un ensemble de mots visuels modélisés dans la partie *intension* et l'ensemble de points d'intérêt qui leur sont associés modélisés dans son *extension*. L'avantage d'une telle structure est qu'elle représente exhaustivement l'ensemble de concepts qui peuvent être distingués en fonction de leur *extension*. Nous proposons ici de réduire la taille du treillis, simplifiant ainsi la représentation des données, tout en tentant de conserver sa structure formelle et son exhaustivité. De plus, dans le schéma de la chaîne de traitement que nous avons proposé, le nombre important de primitives extraites à partir des documents rend la procédure de reconnaissance et de localisation très coûteuse. Cette liste qui paraît exhaustive pour les analystes met en exergue la nécessité d'élaborer une solution qui réduit la taille du treillis tout en gardant le maximum d'information. À l'égard de ces raisons, nous avons choisi d'intégrer le treillis de concepts dans un processus de sélection des primitives. La structure du treillis sera utilisée comme un outil de sélection en se basant sur des heuristiques et des techniques de la littérature. Dans ce qui suit, nous allons passer un panorama de quelques stratégies adoptées dans la littérature pour la sélection des caractéristiques dans le treillis de concepts.

Dans [Ventos and Soldano2005], les auteurs proposent de redéfinir la notion d'*extension* d'un terme de manière à ne tenir compte que d'un certain seuil α de cette partition, ce qui aboutit à des treillis de concepts particuliers appelés *treillis de Galois Alpha*.

[Correia *et al.*2000] proposent de réduire la taille du treillis en éliminant une partie de ses nœuds. Dans [Melo *et al.*2011b], plusieurs techniques de sélection des concepts ont été proposées.

- Sélection basée sur le critère de *stabilité* ou *support* : le critère de la stabilité indique la probabilité de conserver l'*intension* du concept tout en éliminant certains objets de son *extension*. Nous rappelons la définition de la stabilité comme donnée dans [Melo *et al.*2011b] : Soit $\mathcal{K} = (\mathcal{G}, \mathcal{M}, \mathbf{I})$ est un contexte et $(\mathcal{A}, \mathcal{B})$ est un concept de \mathcal{K} . *Card* désigne une fonction de cardinalité. La stabilité est définie comme suit :

$$\sigma(\mathcal{A}, \mathcal{B}) = \frac{\text{Card}(C \subseteq \mathcal{A} | C' = \mathcal{B})}{2^{\text{Card}(\mathcal{A})}} \quad (3.2)$$

- Sélection des concepts en se basant sur le critère de *confiance* : la valeur de confiance d'un concept estime la force de l'implication des attributs des concepts parents dans les objets des concepts fils. Soit $(\mathcal{A}, \mathcal{B})$ un concept parent avec $\mathcal{A} \subset \mathcal{G}$ et $\mathcal{B} \subset \mathcal{M}$. Soit $(\mathcal{C},$

D) le concept fils avec $\mathcal{C} \subset \mathcal{G}$ et $\mathcal{D} \subset \mathcal{M}$. La confiance d'un arc $E : (\mathcal{C}, \mathcal{D}) \rightarrow (\mathcal{A}, \mathcal{B})$:

$$\sigma(E) = \frac{\text{Card}(\mathcal{C})}{\text{Card}(\mathcal{A})} \quad (3.3)$$

Dans cette contribution, nous choisissons d'adopter les techniques proposées respectivement dans [Melo *et al.*2011b] et [Waiyamaï and Lakhal2000]. Pour la technique de [Melo *et al.*2011b], nous avons choisi le critère de stabilité détaillé ci-dessus, qui permet d'éliminer quelques objets de la partie *extension* des concepts du treillis tout en gardant l'*intension*. Quant à la deuxième technique de [Waiyamaï and Lakhal2000], une structure d'un treillis de concepts fréquent a été proposée. Les auteurs proposent de ne garder que les nœuds ayant une *extension* suffisamment grande (par rapport à un seuil). Cette méthode permet de réduire considérablement la taille du treillis en éliminant les concepts qui ne sont pas intéressants et de ne tenir compte que de ceux qui apportent le maximum d'informations avec leurs parties *intension*. L'enjeu de cette étape de notre chaîne de traitement consiste à filtrer les concepts afin de ne garder que les plus pertinents de manière à obtenir un nombre réduit de concepts mais en même temps qui soient les plus informatifs possible.

Plus particulièrement dans notre approche nous avons considéré les relations entre les mots visuels et les points d'intérêt qui leur sont associés, donc l'originalité de l'étape de sélection consiste à éliminer les points d'intérêt qui ne correspondent à aucun mot visuel ce qui permet de ne garder que les points porteurs d'information. Les concepts dont le support ne dépasse par un certain seuil σ sont éliminés ce qui permet de réduire le nombre de concepts qui s'avèrent non intéressants. À chaque document est ainsi associé un treillis, qui sera simplifié en enlevant des caractéristiques non essentielles dans un concept. En effet, une autre étape consiste à éliminer des attributs soit qui alourdissent inutilement un contexte local, soit qui ne permettent pas de distinguer les classes entre elles. Un attribut A est gardé dans un concept \mathcal{C} pour la classe ς si sa fréquence d'apparition $\text{Freq}_{\varsigma, A}$ dans la classe ς est bien supérieure à sa fréquence d'apparition $\text{Freq}_{(\varsigma^*, i)} \neq \varsigma$ dans toute autre classe ς^* du contexte. Notre heuristique permet de garantir un niveau de discrimination des concepts très élevé car le critère de stabilité utilisé permet de conserver l'information sur la classe ς même en éliminant quelques attributs de la partie *intension* du concept \mathcal{C} . Les attributs éliminés sont ceux alourdissant les concepts sans apporter une information supplémentaire sur les classes. Cette étape de sélection est cruciale dans notre contexte vu que les besoins du système consistent spécialement à obtenir de bons résultats de localisation, en éliminant les attributs qui alourdissent le treillis, garantissant ainsi l'obtention d'une reconnaissance plus nette et plus précise des symboles requêtes et rendant ainsi l'extraction des zones du document susceptibles de contenir ces derniers, plus robuste. Cette technique permet aussi de réduire le nombre de concepts ainsi que la partie *intension* des différents concepts tout en conservant l'information la plus pertinente des concepts du treillis.

3.2.5 Détection de symboles dans les documents graphiques

L'objectif de ce travail est double. Le premier objectif consiste à exploiter cette structure comme un outil de sélection de primitives. Le deuxième objectif est le développement d'un outil de localisation de symboles dans les documents graphiques. L'arborescence du treillis utilisée dans notre approche et modélisant les documents graphiques, est construite en suivant la même stratégie décrite dans la deuxième partie de ce manuscrit auquel s'ajoute l'étape de sélection des caractéristiques comme décrite dans la section précédente permettant de réduire la taille du treillis et le rendre digeste et facile à manipuler. Afin de localiser

un symbole requête dans l'image du document, une recherche des mots visuels du symbole modèle dans le treillis correspondant à un document graphique s'avère nécessaire. Le processus de détection est alors transformé à un processus de recherche des similarités entre un ensemble d'attributs désignant le symbole à localiser et un treillis désignant le document graphique.

Dans ce cadre, notons qu'un document peut contenir tous les mots du symbole requête sans que celui-ci n'y soit présent, mais l'objectif de cette approche n'est plus d'optimiser les résultats de localisation ou de minimiser le nombre des faux positifs mais plutôt ce qui nous intéresse c'est d'introduire le treillis dans le contexte de localisation et d'obtenir des détections correctes. La liste de mots visuels contenus dans les concepts du treillis et similaires entre la requête et le document est alors retournée comme sortie du processus de la reconnaissance. Néanmoins et vu que nous nous intéressons au problème de détection de symbole dans un document graphique, nous avons besoin de retourner un ensemble de zones du document susceptibles de contenir le symbole requête. Le problème n'est alors plus de déterminer la classe la plus vraisemblable à un symbole requête uniquement, mais aussi de chercher les régions du document où le symbole requête apparaît. Pour répondre à cet objectif, nous avons mis en exergue la nécessité de prendre en compte les coordonnées spatiales des indices visuels ou des primitives extraites à partir de documents graphiques. Ces primitives sont les points d'intérêt extraits à partir des images.

Nous nous sommes appuyés sur l'idée qu'un mot visuel peut correspondre à un ou plusieurs points d'intérêt (voir Figure.3.2), donc une fois que les caractéristiques similaires entre la requête et le document sont retrouvées, une projection des points d'intérêt correspondant à ces mots visuels dans l'espace 2D du document permet de localiser le symbole. À partir d'une technique de *template matching* entre les régions d'intérêt du document et celles de la requête permet de faire émerger des zones du document pouvant contenir les occurrences les plus fidèlement semblables au symbole requête. De cette manière, le système proposé identifie les symboles réellement présents dans le document. De nombreux travaux se basent

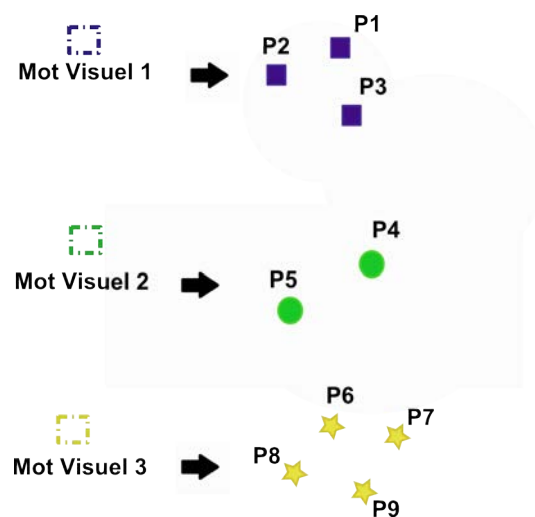


FIGURE 3.2 – Un ensemble de mots visuels et les points d'intérêt qui leur sont associés.

sur la technique de la mise en correspondance entre les points d'intérêt des images ont été proposés dans la littérature. Ces méthodes se divisent généralement en deux catégories : les méthodes locales et les méthodes globales. La mise en oeuvre des méthodes locales s'est

avérée simple et efficace nécessitant un temps de calcul relativement réduit. Le principe des méthodes globales consiste à minimiser une fonction de coût qui fait intervenir l'ensemble des points d'intérêt des images. Parmi les méthodes globales, il existe des méthodes s'appuyant sur des germes ou des points de contrôle qui correspondent à des points où la mise en correspondance est fiable. Ces germes sont utilisés pour conditionner la mise en correspondance globale. La mise en correspondance consiste à retrouver dans les images gauche et droite, les primitives homologues, c'est-à-dire les primitives qui sont la projection de la même entité de la scène. Les différents problèmes souvent rencontrés par les méthodes de mise en correspondance sont :

- Le bruit : peut affecter les niveaux de gris ou la couleur des pixels. Dans notre cas, les documents et les symboles que nous traitons sont parfaits donc ce problème n'existe plus.
- Les occultations peuvent poser le problème de la détermination des correspondant des pixels occultés notamment pour les méthodes locales, car dans ce cas l'hypothèse de départ sur les voisinages respectifs n'est plus respectée. Dans notre cas, ce problème ne se pose plus.

Plus précisément, dans le cadre de ce travail, nous nous sommes intéressés à la mise en correspondance de points d'intérêt du symbole requête avec les points projetés dans le document associés aux mots visuels trouvés lors de la recherche du symbole dans le treillis modélisant le document graphique. Une fois la liste des points d'intérêt établie, on peut procéder à une mise en correspondance. Au moment de l'appariement d'une paire de région requête-document, une comparaison des pixels des régions est effectuée. Ce processus consiste en pratique à effectuer plusieurs balayages couvrant toute la superficie de l'image. Les endroits les plus propices à la présence des mêmes points suivant un score de coupure, sont facilement identifiés par des minimums de distance entre les deux régions. Parmi les distances, nous avons eu recours à la norme L2 (distance euclidienne). L'usage de cette technique permet de corriger le problème de l'orientation estimée de la région car seuls les régions partageant le plus grand nombre de points d'intérêt sont retournées. Cette méthode garantit une précision de détection très élevée car seuls les points d'intérêt associés aux mots visuels retournés par la phase de reconnaissance sont projetés.

3.3 Résultats Expérimentaux

Cette section est dédiée à l'évaluation de notre système original de détection des symboles en contexte afin de tester son aptitude à bien localiser les symboles dans les documents graphiques. Nous évaluons notre système dans le cadre d'un système de détection de symboles dans des documents synthétiques, générés à partir des outils présentés dans [Delalandre *et al.*2007]. Cette évaluation peut sembler limitée mais vu l'absence de bases de documents de référence et la vérité terrain associée, nous nous sommes donc contentés d'évaluer la pertinence des résultats obtenus par notre système sur ces documents.

Généralement, les performances en reconnaissance de symboles sont quantifiées par un taux de reconnaissance, mais pour les problèmes de localisation ou de détection de symboles, comme est le cas de notre approche, les mesures de *rappel* et de *précision* s'avèrent utiles pour l'évaluation des performances fournies. La précision est définie comme suit :

$$Précision = \frac{NbObjDetCorr}{NbObjDet} \quad (3.4)$$

avec $NbObjDetCorr$ désigne le nombre d'objets correctement détectés et $NbObjDet$ désigne le nombre d'objets détectés. Les fausses détections peuvent être évaluées avec ce critère de

précision.

Le critère *Rappel* est défini comme suit :

$$Rappel = \frac{NbObjDetCorr}{NbObj} \quad (3.5)$$

avec *NbObjDetCorr* désigne le nombre d'objets correctement détectés et *NbObj* désigne le nombre d'objets devant être détectés. Des tests ont été effectués sur 25 documents de

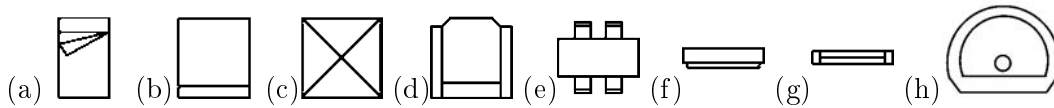


FIGURE 3.3 – Exemples de symboles requêtes interrogés lors du processus de localisation.

la base SESYD [Delalandre *et al.*2007] et 42 symboles requêtes dont quelques échantillons sont présentés dans la Figure.3.3. Bien qu'il est difficile de cerner l'apport de chaque étape du système séparément, nous proposons d'évaluer le système dans son intégralité tout en testant les bonnes détections qui peuvent avoir lieu au cours du processus de localisation. Pour mener cette évaluation, nous avons sélectionné quelques images architecturales sur lesquelles nous appliquons les différentes étapes de la chaîne de traitement décrite dans ce chapitre. La Figure.3.4 montre quelques résultats obtenus lors d'un processus de détection

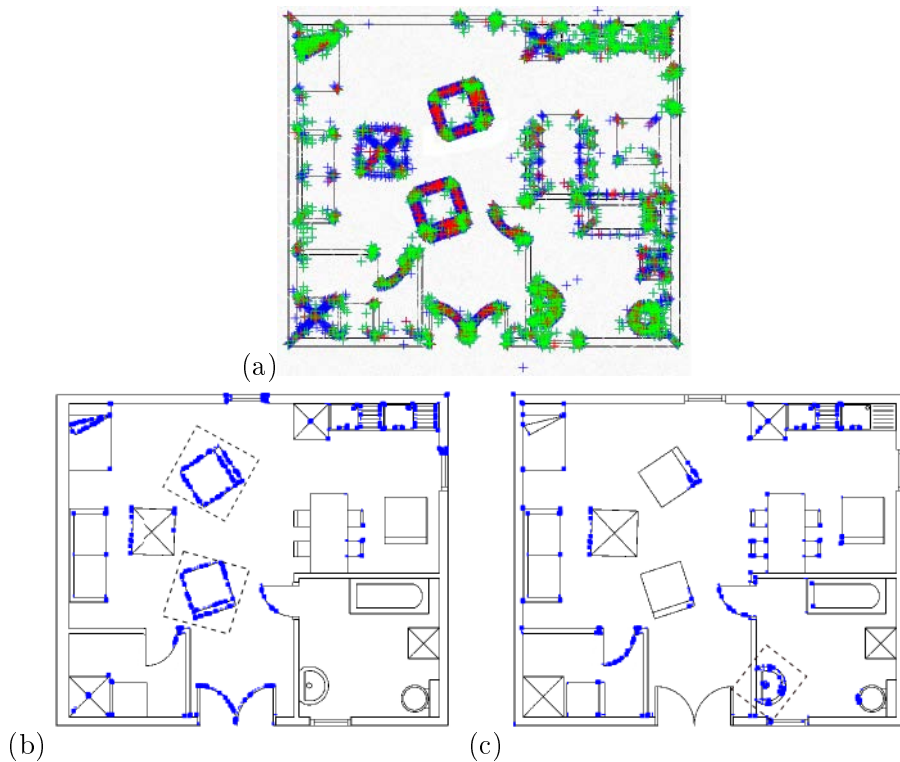


FIGURE 3.4 – Résultats de détection de quelques symboles dans un plan architectural. (a) Projection des points d'intérêt associés aux mots visuels d'un plan architectural, (b) Localisation d'un symbole requête de la Figure.3.3(b), (c) Localisation d'un symbole requête de la Figure.3.3(g).

de symboles dans un plan architectural. Nous constatons que les symboles sont bien localisés dans le document graphique. Parmi les 42 symboles, 35 symboles sont bien localisés

(voir quelques résultats dans la Table.3.1). Les résultats obtenus sont très prometteurs et permettent d'affirmer que l'alliance entre la reconnaissance des requêtes en se basant sur la structure du treillis de concepts et la localisation basée sur la mise en correspondance entre les points de contrôle du symbole modèle et ceux apparaissant dans le document, permet d'obtenir de bonnes performances de localisation. Dans la Figure.3.5, nous constatons que

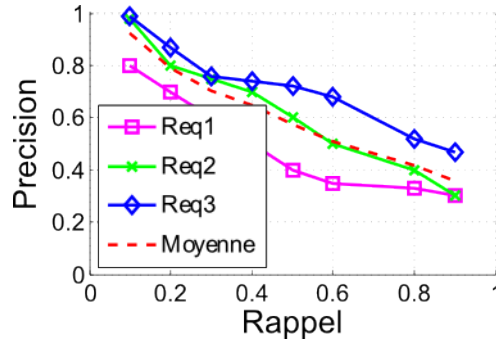


FIGURE 3.5 – Évaluation de quelques réponses du système de localisation lors de l'interrogation de 3 requêtes différentes. Req1 est le symbole de la Figure.3.3(f), Req2 est le symbole de la Figure.3.3(c) et Req3 est le symbole de la Figure.3.3(e).

le système montre un bon compromis entre les taux de *rappel/précision* lors de l'interrogation des 3 requêtes ce qui signifie que le nombre de fausses détections est considérablement réduit. Comme nous pouvons le voir, la précision atteint en moyenne 90%. Le système de localisation de la requête Req1 atteint de bonnes valeurs de précision quand le taux de rappel est réduit. Nous constatons ensuite que ces taux diminuent ce qui pourra se justifier par l'apparition de quelques faux positifs. Néanmoins, même pour des taux de précision assez faibles, le taux de rappel reste proche de la moyenne ce qui donne une idée sur la stabilité du système. Cette expérimentation a permis de montrer que le système proposé atteint de bonnes performances de détection malgré l'apparition de quelques faux positifs dans certains cas de symboles. Son comportement s'avère stable.

TABLE 3.1 – Résultats de localisation de symboles dans les documents graphiques : $NbOcc$: est le nombre d'occurrences de symbole présentes dans le document, CD : le nombre des détections correctes dans le document, FP : le nombre des faux positifs, FN : le nombre des faux négatifs.

Requête	$NbOcc$	CD	FP	FN	Précision	Rappel	Temps de calcul (sec)
Figure.3.3(a)	68	64	2	2	68/64	64/68	4.32
Figure.3.3(b)	27	27	10	3	27/37	27/27	2.04
Figure.3.3(c)	10	8	2	0	8/10	10/10	1.67
Figure.3.3(d)	42	28	10	2	28/42	28/38	2.01
Figure.3.3(e)	50	47	13	5	50/60	47/50	6.19

Dans le tableau.3.2, nous regroupons les mesures d'évaluation des résultats de la localisation obtenus par notre système ainsi que deux approches décrites dans le chapitre précédent. Pour les bases que nous avons utilisées, nous obtenons des performances de l'ordre de 90%. Les résultats fournis par notre système s'avèrent prometteurs et compétitifs en les comparant à l'approche de [Nguyen *et al.*2009] et notre approche décrite dans le chapitre précédent. Ce travail pourra être étendu en renforçant les expérimentations et en améliorant l'étape de sélection des attributs en se basant sur d'autres heuristiques ce qui pourra

améliorer les résultats prometteurs déjà obtenus. L'originalité de notre approche repose en grande partie sur sa simplicité.

	Figure.3.3(a)			Figure.3.3(b)			Figure.3.3(c)			Figure.3.3(d)			Figure.3.3(e)		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
Doc-0	0/0/0	0/0/0	0/0/0	2/3/3	2/2/3	1/1/3	4/4/4	4/4/4	4/3/4	1/0/1	1/1/1	1/1/1	1/1/1	0/1/1	1/1/1
Doc-1	3/3/3	3/3/3	2/2/3	1/0/3	1/2/3	0/0/3	4/4/4	4/4/4	4/4/4	1/1/1	1/1/1	1/1/1	1/0/1	1/1/1	1/1/1
Doc-2	1/1/1	1/1/1	1/1/1	0/1/2	2/2/2	1/1/2	4/4/4	4/4/4	3/4/4	0/0/0	0/0/0	0/0/0	1/1/1	1/2/1	1/1/1
Doc-3	1/2/1	1/1/1	1/2/1	1/2/2	1/1/2	0/0/2	4/4/4	4/4/4	2/4/4	1/2/1	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1
Doc-4	1/1/2	2/2/2	1/2/2	0/2/1	0/1/1	0/1/2	4/4/4	4/4/4	4/4/4	0/0/0	0/0/0	0/0/0	1/1/1	1/1/1	1/1/1
Précision	6/7	7/7	5/7	4/11	6/11	2/11	20/20	20/20	18/20	3/3	3/3	3/3	5/5	5/6	5/5
Rappel	7/7	7/7	7/7	8/11	8/11	3/11	20/20	20/20	18/20	3/3	3/3	2/3	5/5	5/5	5/5

TABLE 3.2 – Résultats de localisation de symboles dans des document de la base SESYD [Delalandre *et al.*2007] (a) notre approche décrite dans le chapitre précédent (b) l'approche de [Nguyen *et al.*2009] (Noter que les résultats de Nguyen sont extraits de [Nguyen *et al.*2009] et (c) notre approche. La notation x/y/z représente : le nombre de bonnes détections de symboles (x), le nombre total des symboles retournés (y) et le nombre des occurrences de symbole dans le document graphique (z).

3.3.1 Impact de l'étape de sélection des attributs et des concepts sur la taille du treillis

Dans la Table.3.3, nous présentons quelques résultats pour évaluer l'impact de l'étape de sélection des attributs et des concepts sur la taille du treillis. Les attributs éliminés sont ceux alourdissant les concepts sans apporter une information supplémentaire sur les classes. Un autre avantage est que la taille du treillis est considérablement réduite comme le montre l'exemple dans la Table.3.3 (ligne 4 par exemple) où le nombre de concepts initial du treillis est de 150 concepts tandis qu'après l'élimination des concepts selon l'heuristique de [Waiyamaï and Lakhali2000] le nombre de concepts est 80 concepts. Le nombre d'objets avant l'étape de la sélection des objets de [Melo *et al.*2011b] est 51 en comparaison à 29 après la sélection. Le taux de réduction des attributs après la sélection est d'environ 30%. Ces heuristiques permettent de réduire clairement la complexité algorithmique du treillis et améliore l'information contenue dans ce dernier (voir Figure.3.6).

	Avant sélection			Après sélection		
	Nombre de concepts	Nombre d'objets	Nombre d'attributs	Nombre de concepts	Nombre d'objets	Nombre d'attributs
Doc-1	70	45	50	64	32	32
Doc-2	150	51	70	80	29	45
Doc-3	48	42	50	39	28	39
Doc-4	50	28	100	34	25	75
(%) de Réduction	-	-	-	31.7%	31.3%	29.25%

TABLE 3.3 – Impact de l'étape de sélection des attributs et des concepts sur la taille du treillis.

3.4 Conclusion

Dans ce chapitre, nous avons présenté une approche originale et robuste pour la détection des symboles dans leur contexte. Le treillis de Galois associé à une méthode de sélection

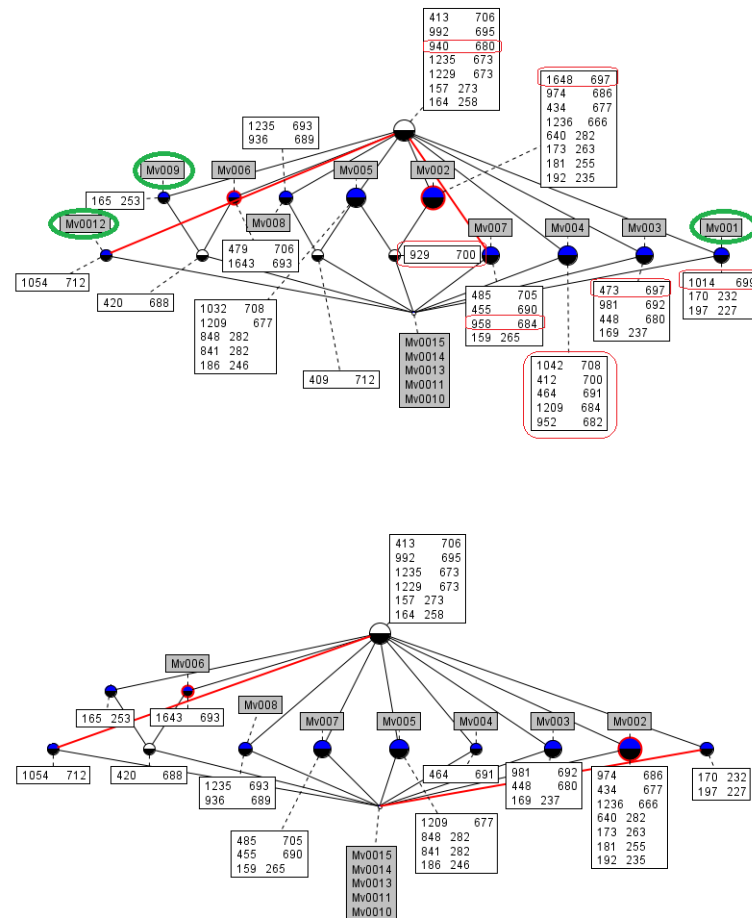


FIGURE 3.6 – Treillis de Galois $\{Objets, Attributs\}$ avant/après la sélection. Les objets entourés par \circ sont ceux éliminés suite à l'étape de la sélection des objets et les attributs entourés par \circ sont ceux éliminés après la sélection des attributs.

de caractéristiques, s'est montré efficace et compétitif avec les approches de la localisation présentées dans la littérature. Un autre constat que nous pouvons faire est que l'étape de sélection des attributs améliore considérablement la phase de la reconnaissance qui a un effet direct sur la phase de localisation d'où l'obtention des résultats de détections des symboles dans les documents graphiques assez prometteurs. Nous sommes conscients que nous sommes loin d'avoir résolu le problème de localisation de symboles dans les documents graphiques mais nous avons effleuré un peu le domaine de de l'Analyse Formelle de Concepts en ouvrant un nouvel axe de recherche et une voie originale via l'usage de la structure du treillis de Galois et son adaptation à divers contextes.

Les évaluations que nous avons faites sont importantes car elles nous renseignent sur le comportement du treillis de Galois dans un nouveau contexte qui est la localisation des symboles dans leur contexte. L'évaluation à partir des bases de documents restreintes conduit forcément à un biais quant aux performances obtenues pour la détection de symboles dans les documents graphiques. Nous avons également montré que le treillis de concepts peut être intégré dans une chaîne de traitement pour concevoir un système de localisation de symboles. La seule contrainte est sa taille, mais de nombreuses stratégies et heuristiques comme nous l'avons détaillées tout au long de ce manuscrit peuvent pallier à ce problème rendant ainsi le treillis une structure efficace dans plusieurs applications.

Quatrième partie
Conclusion Générale

1

Conclusion

Nous avons présenté et évalué plusieurs contributions que nous avons détaillées tout au long de ce manuscrit. Les contributions présentées dans cette thèse sont diverses présentant des alliances originales entre des thématiques différentes à savoir la vision par ordinateur et l'Analyse Formelle de Concepts (AFC).

Notre première contribution a été de proposer une méthode efficace pour la classification d'objets graphiques isolés de leur contexte qui a constitué le point de départ de nos travaux. Nous avons introduit une méthode basée sur une modélisation rigide par l'usage du treillis de concepts que nous avons combiné avec une approche par sac de mots. Nous avons choisi le treillis de concepts comme classifieur car il nous a paru une solution élégante et efficace de représentation des données. Il offre un aspect sémantique vu que les objets et les propriétés qu'il représente sont reliés sémantiquement ce qui permet de renforcer son pouvoir discriminant. Dans le cadre de notre système, l'exécution de la chaîne de traitement de classification de symboles conduit à assigner une classe à un nouvel objet passé en requête et lui associer les instances les plus semblables.

L'étude de cette méthode a permis d'évaluer l'impact et de souligner l'importance de certaines étapes de la chaîne de traitement de reconnaissance à savoir l'étape de représentation des objets graphiques ou encore le choix de la structure du classifieur. Les résultats présentés dans ce travail ont permis une comparaison entre l'utilisation de la signature discrétisée pour la construction du treillis de concepts et l'approche symbolique fondée sur les mots visuels. Nos choix ont été motivés par l'aspect symbolique offert par la représentation par sac de mots et sa contribution dans l'affinement et l'amélioration des résultats du processus de reconnaissance. Outre cette contribution, la charge du calcul a été considérablement réduite en comparaison avec les anciennes approches qui se basent sur le calcul des signatures numériques et leur discrétisation en intervalles figés.

Dans ce cadre et en vue de l'amélioration de cette chaîne de traitement, nous avons présenté une nouvelle approche qui a émergé de cette dernière, c'est ce que nous avons appelée le *Treillis de Galois Fréquent* construit non pas en terme de tout le dictionnaire visuel mais en terme d'une partie comportant les mots visuels ayant le pouvoir discriminant le plus élevé garantissant des performances de classification meilleures. Une comparaison a été proposée entre l'utilisation d'un dictionnaire visuel filtré pour la construction du treillis de Galois et l'approche basée sur le dictionnaire en sa totalité. La principale nouveauté de ce travail est l'utilisation des motifs graphiques les plus pertinents et les plus informatifs afin de construire notre classificateur ce qui garantit un taux de reconnaissance assez élevé et une taille très réduite ainsi qu'un temps de traitement raisonnable durant le processus de

classification.

Dans notre troisième contribution, nous avons présenté une application du treillis de concepts dans le cadre d'un système de classification d'images afin de tester sa robustesse. Notre travail fournit un cadre originale et intéressant d'application du treillis de Galois qui ouvre la voie à mieux explorer cette structure dans des domaines divers tout en intégrant les heuristiques capables de remédier à la complexité algorithmique de ce formalisme dans le pire des cas. Notre contribution vise à tirer profit de la capacité d'un tel classifieur dans le domaine de la classification d'images. L'idée de base est de fournir un moyen de reconnaître une banque d'images et d'adapter la structure du treillis de Galois, utilisé habituellement pour la fouille du texte, dans le domaine de la recherche d'images. En outre, étant donnée que la partie la plus coûteuse de l'approche se fait hors ligne, le système proposé est adapté pour les requêtes en ligne et les résultats obtenus en comparaison avec l'état de l'art sont compétitifs. Sur ce point, nous pouvons dire que le treillis s'est avéré assez robuste et a fourni des résultats prometteurs et semblables aux résultats des autres approches.

Notre travail élargit et étend le domaine de l'Analyse Formelle de Concepts en démontrant que le treillis de Galois peut être utilisé dans des domaines divers. Notre défi est d'appliquer ce genre de démarches pour reconnaître les symboles non segmentés dans le contexte de la localisation de symboles. Dans ce cadre, nous nous sommes intéressés à la localisation de symboles dans les documents graphiques. Nous avons proposé une évaluation de l'impact de l'étape de représentation des primitives des documents graphiques sur les performances d'un ensemble d'approches pixelaires de localisation. Nous avons constaté que le choix d'une bonne technique d'extraction de primitives et leurs bonnes descriptions aboutit souvent à de très bonnes performances. Nous avons présenté aussi une évaluation de quelques approches pixelaires de localisation qui ont prouvé leur efficacité. La contribution de ce travail réside spécialement à montrer expérimentalement que les approches de détection de points d'intérêt sont viables pour les documents graphiques.

Finalement, nous avons présenté une approche originale et robuste pour la détection des symboles dans leur contexte. Le treillis de Galois associé à une méthode de sélection de caractéristiques, s'est montré efficace et compétitif dans le cadre de la localisation de symboles dans les documents graphiques. Nous sommes conscients que nous sommes loin d'avoir résolu le problème de détection de symboles dans leur contexte. L'approche présentée représente un effort continu et une nouvelle contribution s'inscrivant dans le cadre de la résolution des problèmes de la localisation qui malgré le nombre important des travaux qui s'y intéressent, reste un problème ouvert. Les évaluations que nous avons faites sont importantes car elles nous renseignent sur le comportement du treillis de Galois dans un nouveau contexte qui est la localisation des symboles dans leur contexte. Un constat global que nous pouvons faire après ces nombreuses évaluations est que le treillis de Galois présente de nombreux avantages : il présente un espace de recherche exhaustif et concis, en plus il permet de hiérarchiser un grand nombre de données tout en gardant l'aspect relationnel et sémantique entre $\{Objets/Propriétés\}$ ce qui permet de pallier au problème du biais sémantique de certains classifieurs. Les résultats obtenus dans ces travaux ouvrent un certain nombre de perspectives pour des travaux futurs. Plusieurs perspectives qui se présentent naturellement à partir du contenu de cette thèse pourraient représenter des sujets de recherches futurs et ouvrir de nouvelles voies de recherche pour la communauté de vision par ordinateur. Alors que certaines pourraient être traitées par des extensions simples, d'autres nécessitent des travaux de recherche à plus long terme.

Les principales perspectives de recherche qui apparaissent à l'issue de cette thèse concernent

la réutilisabilité de notre travail, à travers la conception d'un système générique de reconnaissance d'objets graphiques. La structure du treillis de Galois utilisée dans nos travaux de thèse nous a permis d'obtenir de très bonnes performances dans le contexte de la reconnaissance, de classification et de localisation de symboles dans les documents graphiques. Une direction à explorer serait d'adapter l'approche présentée ici à une approche de fouille de documents graphiques. Il serait effectivement intéressant de voir, entre autre, l'influence de l'étape de représentation des primitives sur la qualité des détections.

Une des limitations de la représentation par sac de mots est le nombre important des mots visuels à explorer. Une piste à explorer serait de filtrer le vocabulaire comme nous l'avons fait mais en utilisant d'autres heuristiques de sélection des motifs les plus pertinents. Également, nous pouvons envisager de proposer une approche de fouille de motifs fréquents décrivant les objets graphiques en utilisant les règles d'association et les informations découlant de la structure du treillis. Les règles d'association extraites à partir de l'arborescence du treillis représentent une source d'information très fine offrant l'opportunité de fouiller les documents graphiques en extrayant des motifs apparaissant dans des documents graphiques comme les logos par exemple. En outre, une des limitations de notre méthode de localisation de symboles dans les documents graphiques est qu'il n'existe aucune relation spatiale entre les mots visuels extraits à partir de la base de symboles. Il semble donc opportun d'essayer d'améliorer le modèle proposé pour qu'il intègre les relations spatiales entre les mots. Une autre perspective concerne la stratégie de classification adoptée. Tout au long des expérimentations menées durant cette thèse, nous avons remarqué que les similarités ou les ressemblances des objets graphiques (symboles ou images) test influencent le taux de reconnaissance du treillis. En effet, une idée sera d'intégrer une mémoire de session dont le but est de sauvegarder les chemins lors de la navigation du treillis afin de réduire le temps de classification. Un raisonnement à partir de cas peut être défini pour parcourir le treillis. Par exemple, pour le premier objet passé en requête, au cours de la phase de la recherche des instances qui lui sont les plus fidèlement semblables, les concepts qui partagent les mêmes attributs seront stockés au fur et à mesure que nous avançons dans l'arborescence ce qui aura certainement un impact sur le temps de classification et la précision lors d'une deuxième interrogation de la base : un deuxième objet requête s'il appartient à la même catégorie que la première requête, sera reconnu en un temps réduit car il y a une forte probabilité que les chemins conservés lors de la reconnaissance de la première requête soient les chemins menant à la catégorie de la deuxième image. Néanmoins, quand la deuxième requête est différente de la première image qui la précède dans le processus de classification, il est nécessaire de construire des nouveaux nœuds donc le temps de classification sera plus lent. En plus, un grand manque à combler dans les outils existants et explorant le treillis de concepts provient du fait qu'ils ne profitent que très peu des techniques de visualisation efficaces qui permettent de faire mieux ressortir l'information pertinente et qui ont été proposées par des experts du domaine. Un outil intéressant en serait certainement un qui établirait le lien manquant entre les travaux effectués dans le domaine de la visualisation de l'information et ceux effectués dans le domaine de l'Analyse Formelle de Concepts. Ainsi, l'information qui doit retenir l'attention de l'utilisateur, en particulier les relations et les regroupements dans les données, serait efficacement mis en évidence et la quantité du temps investi dans la navigation du treillis pour trouver cette information pourrait être réduite de manière significative.

1.1 Publications Scientifiques

Conférences internationales

- Impact of a Codebook Filtering Step on a Galois Lattice Structure for Graphics Recognition, Ameni Boumaiza and Salvatore Tabbone : 21st International Conference on Pattern Recognition ICPR 2012, Tsukuba Science City, JAPAN.
- Graphics Recognition using a Galois Lattice for Frequent Graphical Patterns, Ameni Boumaiza and Salvatore Tabbone, 10th IAPR International Workshop on Document Analysis Systems, DAS 2012, Gold Coast, Australie.
- A Novel Approach for Graphics Recognition based on Galois Lattice and Bag of Words Representation, Ameni Boumaiza and Salvatore Tabbone : International Conference on Document Analysis and Recognition, ICDAR 2011, Pékin, Chine.

Conférences Nationales

- Vers l'utilisation d'un Treillis de Galois Fréquent pour la reconnaissance de symboles graphiques, Ameni Boumaiza and Salvatore Tabbone, Colloque International Francophone sur l'écrit et le Document 2012 (CIFED).
- Classification de symboles avec un treillis de Galois et une représentation par sac de mots, Ameni Boumaiza and Salvatore Tabbone : publié dans ORASIS - Congrès des jeunes chercheurs en vision par ordinateur (2011).

Travaux en cours

- On the use of the Galois Lattice structure for Image Classification (en cours), Ameni Boumaiza and Salvatore Tabbone.
- Galois Lattice : An efficient tool for Graphics Recognition (en cours), Ameni Boumaiza and Salvatore Tabbone.
- Symbol Spotting in Graphical Documents Based on the Galois Lattice Structure, Ameni Boumaiza and Salvatore Tabbone.

Glossaire

AFC : Analyse Formelle de Concepts

ART : Angular Radium Transform

BFTree : Best First Decision Tree

BoW : Bag of Words

CBSM : Circular Blurred Shape Model

CF : Contexte de Forme

CFPI : Contexte de Forme au niveau des Points d'Intérêt

DoG : Derivative of Gaussian

GFD : Generic Fourier Descriptor

k-ppv : *k*-plus proche voisin

LoG : Laplacian of Gaussian

MSER : Maximally stable extremal regions

NBTree : Naive Bayesian Tree

PPI : Plus Petit Intervalle

SIFT : Scale-Invariant Feature Transform

SURF : Speeded Up Robust Features

SVM : Support Vector Machine

Bibliographie

- [Agarwal *et al.*2004] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(11) :1475–1490, 2004.
- [Agrawal and Srikant1994a] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499, 1994.
- [Agrawal and Srikant1994b] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.
- [Baeza-Yates and Ribeiro-Neto2011] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.
- [Bagdanov *et al.*2007] Andrew D. Bagdanov, Lamberto Ballan, Marco Bertini, and Alberto Del Bimbo. Trademark matching and retrieval in sports video databases. In *Proceedings of the international workshop on Workshop on multimedia information retrieval, MIR '07*, pages 79–86, New York, NY, USA, 2007. ACM.
- [Barbu *et al.*2005] Eugen Barbu, Pierre Héroux, Sébastien Adam, and Éric Trupin. Using bags of symbols for automatic indexing of graphical document image databases. In *GREC*, pages 195–205, 2005.
- [Barbut and Monjardet1970] Marc Barbut and Bernard Monjardet. *Ordre et classification : algèbre et combinatoire, TOME I. Méthodes Mathématiques des Sciences de l'Homme, Collection Hachette Université*. Librairie Hachette, Paris, 1970.
- [Barrat and Tabbone2010] Sabine Barrat and Salvatore Tabbone. A bayesian network for combining descriptors : application to symbol recognition. *International Journal on Document Analysis and Recognition*, 13(1) :65–75, 2010.
- [Bay *et al.*2006] Herbert Bay, Tinne Tuytelaars, and Luc J. Van Gool. Surf : Speeded up robust features. In *ECCV (1)*, pages 404–417, 2006.
- [Bay *et al.*2008] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc J. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3) :346–359, 2008.
- [Belongie *et al.*2000] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context : A new descriptor for shape matching and object recognition. In *NIPS*, pages 831–837, 2000.
- [Belongie *et al.*2002] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4) :509–522, April 2002.

- [Berretti *et al.*2001] Stefano Berretti, Alberto Del Bimbo, and Enrico Vicario. Efficient matching and indexing of graph models in content-based retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(10) :1089–1105, October 2001.
- [Bertet *et al.*2008] Karell Bertet, Stéphanie Guillas, Muriel Visani, and Jean-Marc Ogier. A propos des liens entre arbre de décision et treillis dichotomique. In Antoine Tabbone et Thierry Paquet, editor, *Dixième Colloque International Francophone sur l'Écrit et le Document*, pages 25–30, France, 2008. Groupe de Recherche en Communication Ecrite.
- [Bi and Vapnik2003] Jinbo Bi and Vladimir Vapnik. Learning with rigorous support vector machines. In *COLT*, pages 243–257, 2003.
- [Birkhoff1967] G. Birkhoff. *Lattice theory*. Third edition. American Mathematical Society Colloquium Publications, Vol. XXV. American Mathematical Society, Providence, R.I., 1967.
- [Blostein and Kwon2002] Dorothea Blostein and Young-Bin Kwon, editors. *Graphics Recognition Algorithms and Applications, 4th International Workshop, GREC 2001, Kingston, Ontario, Canada, September 7-8, 2001, Selected Papers*, volume 2390 of *Lecture Notes in Computer Science*. Springer, 2002.
- [Boiman *et al.*2008] Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.
- [Bordat1986] J. P. Bordat. Calcul pratique du treillis de Galois d'une correspondance. *Mathématiques et sciences humaines*, 96 :31–47, 1986.
- [Bosch *et al.*2007] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Representing shape with a spatial pyramid kernel. In *CIVR*, pages 401–408, 2007.
- [Boullé2009] Marc Boullé. Une méthode de classification supervisée sans paramètre pour l'apprentissage sur les grandes bases de données. In *EGC*, pages 259–264, 2009.
- [Bouthinon *et al.*2006] Dominique Bouthinon, Henry Soldano, and Véronique Ventos. Logical settings for concept learning from incomplete examples in first order logic. *CoRR*, abs/cs/0607096, 2006.
- [Carpineto and Romano1993] Claudio Carpineto and Giovanni Romano. Galois : An order-theoretic approach to conceptual clustering. In *ICML*, pages 33–40, 1993.
- [Carpineto and Romano1996] Claudio Carpineto and Giovanni Romano. A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, 24(2) :95–122, 1996.
- [Chein1969] Michel Chein. Algorithme de recherche des sous-matrices premières d'une matrice. *Bull. Math. Soc. Sc. Math. de Roumanie*, 1(13) :21–25, 1969.
- [Cordella and Vento2000] Luigi P. Cordella and Mario Vento. Symbol recognition in documents : a collection of techniques ? *IJDAR*, 3(2) :73–88, 2000.
- [Correia *et al.*2000] Joachim Hereth Correia, Gerd Stumme, Rudolf Wille, and Uta Wille. Conceptual knowledge discovery and data analysis. In *ICCS*, pages 421–437, 2000.
- [Coustaty *et al.*2010] Mickaël Coustaty, Stéphanie Guillas, Jean-Marc Ogier, Muriel Visani, and Karell Bertet. Reconnaissance de symboles à partir d'une signature structurelle flexible et d'un classifieur de type treillis de galois. *Technique et Science Informatiques*, 29(6) :665–690, 2010.
- [Daubechies1992] I. Daubechies. *Ten Lectures on Wavelets*. Number 61 in CBMS/NSF Series in Applied Math. 1992.

-
- [Delalandre *et al.*2004] Mathieu Delalandre, Éric Trupin, and Jean-Marc Ogier. Symbols recognition system for graphic documents combining global structural approaches and using a xml representation of data. In *SSPR/SPR*, pages 425–433, 2004.
- [Delalandre *et al.*2007] Mathieu Delalandre, Tony P. Pridmore, Ernest Valveny, Hervé Locteau, and Éric Trupin. Building synthetic graphical documents for performance evaluation. In *GREC*, pages 288–298, 2007.
- [Delalandre *et al.*2008] Mathieu Delalandre, Ernest Valveny, and Josep Lladós. Performance evaluation of symbol recognition and spotting systems : An overview. 2008.
- [Delalandre *et al.*2010] Mathieu Delalandre, Ernest Valveny, Tony P. Pridmore, and Dimosthenis Karatzas. Generation of synthetic documents for performance evaluation of symbol recognition & spotting systems. *IJDAR*, 13(3) :187–207, 2010.
- [Derrode and Ghorbel2001] Stéphane Derrode and Faouzi Ghorbel. Robust and efficient fourier-mellin transform approximations for gray-level image reconstruction and complete invariant description. *Computer Vision and Image Understanding*, 83(1) :57–78, 2001.
- [Dosch and Lladós2003] Philippe Dosch and Josep Lladós. Vectorial Signatures for Symbol Discrimination. In Josep Lladós and Young-Bin Kwon, editors, *Graphics Recognition : Recent Advances and Perspectives, Fifth International Workshop - GREC 2003*, volume 3088 of *Lecture Notes in Computer Science*, pages 154–165, Barcelone, Espagne, 2003. Josep Lladós, Springer-Verlag.
- [Duchenne *et al.*2011] Olivier Duchenne, Armand Joulin, and Jean Ponce. A graph-matching kernel for object categorization. In *ICCV*, pages 1792–1799, 2011.
- [Duquenne1999] Vincent Duquenne. Latticial structures in data analysis. *Theor. Comput. Sci.*, 217(2) :407–436, April 1999.
- [Efron *et al.*2004] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32 :407–499, 2004.
- [Emmanuel and Samuelides2004] Emmanuel and Manuel Samuelides. Galois lattice theory for probabilistic visual landmarks. *J. UCS*, 10(8) :1014–1033, 2004.
- [Escalera *et al.*2009] Sergio Escalera, Alicia Fornés, Oriol Pujol, Alberto Escudero, and Petia Radeva. Circular blurred shape model for symbol spotting in documents. In *Proceedings of the 16th IEEE international conference on Image processing, ICIP'09*, pages 1985–1988, Piscataway, NJ, USA, 2009. IEEE Press.
- [Fang and Qiu2003] J. Fang and G. Qiu. Human face detection using angular radial transform and support vector machines. In *ICIP (1)*, pages 669–672, 2003.
- [Fernando *et al.*2012] Basura Fernando, élisha Fromont, and Tinne Tuytelaars. Effective use of frequent itemset mining for image classification. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV (1)*, volume 7572 of *Lecture Notes in Computer Science*, pages 214–227. Springer, 2012.
- [Ferré and Rudolph2009] Sébastien Ferré and Sebastian Rudolph, editors. *Formal Concept Analysis, 7th International Conference, ICFCA 2009, Darmstadt, Germany, May 21-24, 2009, Proceedings*, volume 5548 of *Lecture Notes in Computer Science*. Springer, 2009.
- [Foggia *et al.*1999] Pasquale Foggia, Carlo Sansone, Francesco Tortorella, and Mario Vento. Definition and validation of a distance measure between structural primitives. *Pattern Anal. Appl.*, 2(3) :215–227, 1999.

- [Fonseca and Jorge2003] Manuel J. Fonseca and Joaquim A. Jorge. Towards content-based retrieval of technical drawings through high-dimensional indexing. *Computers & Graphics*, 27(1) :61–69, 2003.
- [Fonseca *et al.*2005] Manuel J. Fonseca, Alfredo Ferreira, and Joaquim A. Jorge. Content-based retrieval of technical drawings. *IJCAT*, 23(2-4) :86–100, 2005.
- [Frejlichowski and Forczmanski2010] Dariusz Frejlichowski and Pawel Forczmanski. General shape analysis applied to stamps retrieval from scanned documents. In *AIMSA*, pages 251–260, 2010.
- [Fu *et al.*2004a] Huaiyu Fu, Huaiguo Fu, Patrick Njiwoua, and Engelbert Mephu Nguifo. A comparative study of fca-based supervised classification algorithms. In Peter W. Eklund, editor, *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings*, volume 2961 of *Lecture Notes in Computer Science*, pages 313–320. Springer, 2004.
- [Fu *et al.*2004b] Huaiyu Fu, Huaiguo Fu, Patrick Njiwoua, and Engelbert Mephu Nguifo. A comparative study of fca-based supervised classification algorithms. In *ICFCA*, pages 313–320, 2004.
- [Galleguillos *et al.*2008] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, 2008.
- [Ganter and Wille1999] Bernhard Ganter and Rudolf Wille. *Formal concept analysis - mathematical foundations*. Springer, 1999.
- [Ganter1996] Bernhard Ganter. Attribute exploration with background knowledge. *Theoretical Computer Science*, 217 :215–233, 1996.
- [Ganter2010] Bernhard Ganter. Two basic algorithms in concept analysis. In *ICFCA*, pages 312–340, 2010.
- [Gehler and Nowozin2009] Peter V. Gehler and Sebastian Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228, 2009.
- [Gilbert *et al.*2009] Andrew Gilbert, John Illingworth, and Richard Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *ICCV*, pages 925–931, 2009.
- [Godin and Missaoui1994] Robert Godin and Rokia Missaoui. An incremental concept formation approach for learning from databases. *Theor. Comput. Sci.*, 133(2) :387–419, 1994.
- [Grauman and Darrell2005] Kristen Grauman and Trevor Darrell. The pyramid match kernel : Discriminative classification with sets of image features. In *ICCV*, pages 1458–1465, 2005.
- [Guénoche1990] A. Guénoche. Construction du treillis de galois d’une relation binaire. In *Revue Math. Inf. Sci. Hum*, 109, pages 41–53, 1990.
- [Guigues and Duquenne1986] J.-L. Guigues and V. Duquenne. Familles minimales d’implications informatives résultant d’un tableau de données binaires. *Mathématiques et Sciences Humaines*, 95 :5–18, 1986.
- [Guillas *et al.*2005] Stéphanie Guillas, Karell Bertet, and Jean-Marc Ogier. A generic description of the concept lattices’ classifier : Application to symbol recognition. In *GREC*, pages 47–60, 2005.
- [Harris and Stephens1988] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.
- [Hu1962] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2) :179–187, February 1962.

-
- [Im and Cho2006] Seung-Bin Im and Sung-Bae Cho. Context-based scene recognition using bayesian networks with scale-invariant feature transform. In *ACIVS*, pages 1080–1087, 2006.
- [Jurie and Triggs2005] Frédéric Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, pages 604–610, 2005.
- [Kass *et al.*1988] Michael Kass, Andrew P. Witkin, and Demetri Terzopoulos. Snakes : Active contour models. *International Journal of Computer Vision*, 1(4) :321–331, 1988.
- [Kauppinen *et al.*1995] Hannu Kauppinen, Tapio Seppänen, and Matti Pietikäinen. An experimental comparison of autoregressive and fourier-based descriptors in 2d shape classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(2) :201–207, February 1995.
- [Kawaguchi and Nishii2007] Kawaguchi Kawaguchi and Ryuei Nishii. Hyperspectral image classification by bootstrap adaboost with random decision stumps. *IEEE T. Geoscience and Remote Sensing*, 45(11-2) :3845–3851, 2007.
- [Kaytoue-Uberall *et al.*2010] Mehdi Kaytoue-Uberall, Zainab Assaghir, Nizar Messai, and Amedeo Napoli. Two complementary classification methods for designing a concept lattice from interval data. In *FoIKS*, pages 345–362, 2010.
- [Khan *et al.*2009] Fahad Shahbaz Khan, Joost van de Weijer, and Maria Vanrell. Top-down color attention for object recognition. In *ICCV*, pages 979–986, 2009.
- [Kodratoff and Tecuci1988] Y. Kodratoff and G. Tecuci. Learning based on conceptual distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10 :897–909, 1988.
- [Kourie and Oosthuizen1998] Derrick G. Kourie and G. Deon Oosthuizen. Lattices in machine learning : Complexity issues. *Acta Inf.*, 35(4) :269–292, 1998.
- [Kuznetsov and Obiedkov2001] Sergei O. Kuznetsov and Sergei A. Obiedkov. Algorithms for the construction of concept lattices and their diagram graphs. In *PKDD*, pages 289–300, 2001.
- [Kuznetsov and Obiedkov2002] Sergei O. Kuznetsov and Sergei A. Obiedkov. Comparing performance of algorithms for generating concept lattices. *J. Exp. Theor. Artif. Intell.*, 14(2-3) :189–216, 2002.
- [Latiri and Yahia2001] Cherif Chiraz Latiri and Sadok Ben Yahia. Generating implicit association rules from textual data. In *AICCSA*, pages 137–143, 2001.
- [Lazebnik *et al.*2004] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Semi-local affine parts for object recognition. In *British Machine Vision Conference*, volume volume 2, pages 779–788, 2004.
- [Liu *et al.*2009] Rujie Liu, Yuehong Wang, Takayuki Baba, and Daiki Masumoto. Shape detection from line drawings by hierarchical matching. In *Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns, CAIP '09*, pages 922–929, Berlin, Heidelberg, 2009. Springer-Verlag.
- [Lladós and Kwon2004] Josep Lladós and Young-Bin Kwon, editors. *Graphics Recognition, Recent Advances and Perspectives, 5th International Workshop, GREC 2003, Barcelona, Spain, July 30-31, 2003, Revised Selected Papers*, volume 3088 of *Lecture Notes in Computer Science*. Springer, 2004.
- [Lladós *et al.*2001] J. Lladós, E. Marti, and J.J. Villanueva. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23 :1137–1143, 2001.
- [Locteau *et al.*2007] Hervé Locteau, Sébastien Adam, Eric Trupin, Jacques Labiche, and Pierre Héroux. Symbol Spotting using Full Visibility Graph Representation. In *Proceedings of the IAPR Workshop on Graphics Recognition*, pages 49–50, Brésil, July 2007.

- [Lowe1999] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.
- [Lowe2004] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2) :91–110, November 2004.
- [Luqman *et al.*2010] Muhammad Muzzamil Luqman, Mathieu Delalandre, Thierry Brouard, Jean-Yves Ramel, and Josep Lladós. Fuzzy intervals for designing structural signature : an application to graphic symbol recognition. In *Proceedings of the 8th international conference on Graphics recognition : achievements, challenges, and evolution*, GREC’09, pages 12–24, Berlin, Heidelberg, 2010. Springer-Verlag.
- [MacLean and Tsotsos2008] W. James MacLean and John K. Tsotsos. Fast pattern recognition using normalized grey-scale correlation in a pyramid image representation. *Mach. Vision Appl.*, 19(3) :163–179, February 2008.
- [MacQueen1967] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [Marée *et al.*2005] Raphaël Marée, Pierre Geurts, Justus H. Piater, and Louis Wehenkel. Bio-medical image classification with random subwindows and decision trees. In *CVBIA*, pages 220–229, 2005.
- [Marszalek and Schmid2007] Marcin Marszalek and Cordelia Schmid. Semantic hierarchies for visual object recognition. In *CVPR*, 2007.
- [Martinez and Loissant2002] José Martinez and Erwan Loissant. Browsing image databases with galois’ lattices. In *SAC*, pages 791–795, 2002.
- [Matas *et al.*2004] Jiri Matas, Ondrej Chum, Martin Urban, and Tomáš Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Comput.*, 22(10) :761–767, 2004.
- [Melo *et al.*2011a] Cassio Melo, Benedicte Le-Grand, Marie-Aude Aufaure, and Anastasia Bezerianos. Extracting and visualising tree-like structures from concept lattices. In *Proceedings of the 2011 15th International Conference on Information Visualisation*, IV ’11, pages 261–266, Washington, DC, USA, 2011. IEEE Computer Society.
- [Melo *et al.*2011b] Cássio A. Melo, Bénédicte Le Grand, Marie-Aude Aufaure, and Anastasia Bezerianos. Extracting and visualising tree-like structures from concept lattices. In *IV*, pages 261–266, 2011.
- [Mephu Nguifo *et al.*1998] Engelbert Mephu Nguifo, M.S. Lagrange, and M. Renaud. Plata : an application of legal, a machine learning based system, to a typology of archaeological ceramics. *Intl. Journ. of Computers and the Humanities (CHUM)*, 31(3) :169–187, 1998.
- [Mikolajczyk and Schmid2001] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, pages 525–531, 2001.
- [Mikolajczyk and Schmid2004] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1) :63–86, 2004.
- [Mikolajczyk and Schmid2005] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10) :1615–1630, 2005.

-
- [Moravec1977] Hans Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, page 584, August 1977.
- [Müller and Rigoll2000] Stefan Müller and Gerhard Rigoll. Engineering drawing database retrieval using statistical pattern spotting techniques. In *Selected Papers from the Third International Workshop on Graphics Recognition, Recent Advances, GREC '99*, pages 246–255, London, UK, UK, 2000. Springer-Verlag.
- [Nguifo and Njiwoua2005] Engelbert Mephu Nguifo and Patrick Njiwoua. Treillis de concepts et classification supervisée. *Technique et Science Informatiques*, 24(4) :449–488, 2005.
- [Nguyen *et al.*2009] Thi-Oanh Nguyen, Salvatore Tabbone, and Alain Boucher. A symbol spotting approach based on the vector model and a visual vocabulary. In *ICDAR*, pages 708–712, 2009.
- [Njiwoua and Mephu Nguifo1997] Patrick Njiwoua and Engelbert Mephu Nguifo. Forwarding the choice of bias. legal-f : Using feature selection to reduce the complexity of legal. In *Proc. of Benelux Conf. on Machine Learning (BENELEARN-97)*, pages 79–88, Tilburg, Pays-Bas, oct 1997.
- [Njiwoua and Mephu Nguifo1999] Patrick Njiwoua and Engelbert Mephu Nguifo. Améliorer l'apprentissage à partir d'instances grâce à l'induction de concepts : le système cible. *Revue d'intelligence artificielle(RIA)*, 13(2) :413–440, 1999. <http://ria.revuesonline.com/>.
- [Njiwoua2000] Patrick Njiwoua. *Contributions à l'apprentissage symbolique automatique par l'usage du treillis de Galois*. PhD thesis, CRIL, Lens, France, jan 2000.
- [Norris1978] E. M. Norris. An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées*, 23(2) :243–250, 1978.
- [Nourine and Raynaud2002] Lhouari Nourine and Olivier Raynaud. A fast incremental algorithm for building lattices. *J. Exp. Theor. Artif. Intell.*, 14(2-3) :217–227, 2002.
- [Ogier and Trupin2003] Jean-Marc Ogier and Éric Trupin, editors. *Pattern Recognition in Information Systems, Proceedings of the 3rd International Workshop on Pattern Recognition in Information Systems, PRIS 2003, In conjunction with ICEIS 2003, Angers, France, April 2003*. ICEIS Press, 2003.
- [Oosthuizen1988] Gerhard Deon Oosthuizen. *The use of a lattice in knowledge processing*. PhD thesis, Glasgow, Scotland, UK, UK, 1988. UMI Order No. GAXD-96805.
- [Pasquier *et al.*1999] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient mining of association rules using closed itemset lattices. *Inf. Syst.*, 24(1) :25–46, 1999.
- [Perronnin and Dance2007] Florent Perronnin and Christopher R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*. IEEE Computer Society, 2007.
- [Qureshi *et al.*2008] Rashid Jalal Qureshi, Jean-Yves Ramel, Didier Barret, and Hubert Cardot. Graphics recognition. recent advances and new opportunities. chapter Spotting Symbols in Line Drawing Images Using Graph Representations, pages 91–103. Springer-Verlag, Berlin, Heidelberg, 2008.
- [Ricard *et al.*2004] Julien Ricard, David Coeurjolly, and Atilla Baskurt. Generalizations of Angular Radial Transform for 2D and 3D Shape Retrieval. Technical Report RR-LIRIS-2004-023, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon, June 2004.
- [Rusiñol and Lladós2005] Marçal Rusiñol and Josep Lladós. Symbol spotting in technical drawings using vectorial signatures. In *GREC*, pages 35–46, 2005.

- [Rusiñol and Lladós2007] Marçal Rusiñol and Josep Lladós. A region-based hashing approach for symbol in technical documents. In *GREC*, pages 104–113, 2007.
- [Rusiñol and Lladós2008] Marçal Rusiñol and Josep Lladós. Word and symbol spotting using spatial organization of local descriptors. In *Document Analysis Systems*, pages 489–496, 2008.
- [Sahami1995] Mehran Sahami. Learning classification rules using lattices (extended abstract). In *ECML*, pages 343–346, 1995.
- [Sivic and Zisserman2009] Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4) :591–606, 2009.
- [Smeulders *et al.*2000] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12) :1349–1380, December 2000.
- [Tabbone and Zuwala2007] Salvatore Tabbone and Daniel Zuwala. An indexing method for graphical documents. In Flavio Bortolozzi and Robert Sabourin, editors, *9th International Conference on Document Analysis and Recognition - ICDAR'07*, volume 2, pages 789 – 793, Curitiba, Brésil, 2007. IAPR, IEEE Computer Society.
- [Tabbone *et al.*2006] Salvatore Tabbone, Laurent Wendling, and J.-P. Salmon. A new shape descriptor defined on the radon transform. *Computer Vision and Image Understanding*, 102(1) :42–51, 2006.
- [Tanaka1995] Eiichi Tanaka. Theoretical aspects of syntactic pattern recognition. *Pattern Recognition*, 28(7) :1053–1061, 1995.
- [Teh and Chin1988] C.H. Teh and R.T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10 :496–513, 1988.
- [Tibshirani1996] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58 :267–288, 1996.
- [Tombre and Chhabra1998] Karl Tombre and Atul K. Chhabra, editors. *Graphics Recognition, Algorithms and Systems, Second International Workshop, GREC'97, Nancy, France, August 22-23, 1997, Selected Papers*, volume 1389 of *Lecture Notes in Computer Science*. Springer, 1998.
- [Veltkamp *et al.*2001] Remco C. Veltkamp, Hans Burkhardt, and Hans-Peter Kriegel, editors. *State-of-the-Art in Content-Based Image and Video Retrieval [Dagstuhl Seminar, 5-10 December 1999]*. Kluwer, 2001.
- [Venter *et al.*1997] F. J. Venter, G. Deon Oosthuizen, and J. D. Roos. Lattice-based knowledge discovery in network management data. *Informatika (Slovenia)*, 21(2), 1997.
- [Ventos and Soldano2005] Véronique Ventos and Henry Soldano. Les treillis de galois alpha. *Revue d'Intelligence Artificielle*, 19(4-5) :799–827, 2005.
- [Waiyamai and Lakhal2000] Kitsana Waiyamai and Lotfi Lakhal. Knowledge discovery from very large databases using frequent concept lattices. In *Proceedings of the 11th European Conference on Machine Learning*, ECML '00, pages 437–445, London, UK, UK, 2000. Springer-Verlag.
- [Wang *et al.*2009] Gang Wang, Derek Hoiem, and David A. Forsyth. Learning image similarity from flickr groups using stochastic intersection kernel machines. In *ICCV*, pages 428–435, 2009.
- [Weber and Tabbone2012] Jonathan Weber and Salvatore Tabbone. Symbol spotting for technical documents : An efficient template-matching approach. In *ICPR*, pages 669–672, 2012.

-
- [Wenyin *et al.*2007] Liu Wenyin, Wan Zhang, and Luo Yan. An interactive example-driven approach to graphics recognition in engineering drawings. *Int. J. Doc. Anal. Recognit.*, 9(1) :13–29, February 2007.
- [Wille2009] Rudolf Wille. Restructuring lattice theory : An approach based on hierarchies of concepts. In *ICFCA*, pages 314–339, 2009.
- [Xie *et al.*2002] Zhipeng Xie, Wynne Hsu, Zongtian Liu, and Mong-Li Lee. Concept lattice based composite classifiers for high predictability. *J. Exp. Theor. Artif. Intell.*, 14(2-3) :143–156, 2002.
- [Yang *et al.*1993] Der-Shung Yang, Julie L. Webster, Larry A. Rendell, James H. Garrett Jr., and Doris S. Shaw. Management of graphical symbols in a cad environment : A neural network approach. In *ICTAI*, pages 272–279, 1993.
- [Yang *et al.*2009] Jingjing Yang, Yuanning Li, YongHong Tian, Lingyu Duan, and Wen Gao. Group-sensitive multiple kernel for object categorization. In *ICCV*, pages 436–443, 2009.
- [Yuan and Wu2008] Junsong Yuan and Ying Wu. Context-aware clustering. In *CVPR*, 2008.
- [Zahn and Roskies1972] Charles T. Zahn and Ralph Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, 21 :269–281, 1972.
- [Zanibbi and Yu2011] Richard Zanibbi and Li Yu. Math spotting : Retrieving math in technical documents using handwritten query images. In *ICDAR*, pages 446–451. IEEE, 2011.
- [Zenou and Samuelides2003] Emmanuel Zenou and Manuel Samuelides. Characterization of image sets : The galois lattice approach. In *International Conference on Advanced Concepts on Intelligent Vision Systems*, 2003.
- [Zhang and Lu2002] Dengsheng Zhang and Guojun Lu. Shape-based image retrieval using generic fourier descriptor. *Sig. Proc. : Image Comm.*, 17(10) :825–848, 2002.
- [Zhang and Lu2003] Dengsheng Zhang and Guojun Lu. A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval. *J. Visual Communication and Image Representation*, pages 39–57, 2003.
- [Zhang and Wang1994] Edward Y. Y. Zhang and Patrick Shen-Pei Wang. A new parallel thinning methodology. *IJPRAI*, pages 999–1011, 1994.
- [Zhang2010] Nan Zhang. Computing optimised parallel speeded-up robust features (p-surf) on multi-core processors. *International Journal of Parallel Programming*, 38(2) :138–158, 2010.
- [Zhao *et al.*2003] Wen-Yi Zhao, Rama Chellappa, P. Jonathon Phillips, and Azriel Rosenfeld. Face recognition : A literature survey. *ACM Comput. Surv.*, 35(4) :399–458, 2003.
- [Zhou *et al.*2010] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV(5)*, pages 141–154, 2010.
- [Zuwala and Tabbone2006] Daniel Zuwala and Salvatore Tabbone. A method for symbol spotting in graphical documents. In *Document Analysis Systems*, pages 518–528, 2006.