Localisation de symboles dans les documents graphiques

Thi Oanh Nguyen

NANCY 2 UNIVERSITE

UFR mathématiques et informatique **École doctorale IAEM Lorraine**

Département de formation doctorale en informatique

Localisation de symboles dans les documents graphiques

THÈSE

présentée et soutenue publiquement le 16 décembre 2009

pour l'obtention du

Doctorat de l'université Nancy 2

(spécialité informatique)

par

NGUYEN Thi Oanh

Composition du jury

Président :	Jean-Marc OGIER	
Rapporteurs :	Josep LLADÓS	Universitat Autònoma de Barcelona
	Jean-Marc OGIER	Université de la Rochelle
Examinateurs :	Pierre HÉROUX	Université de Rouen
	Marie-Dominique DEVIGNES	CNRS
Directeur de thèse :	Salvatore-Antoine TABBONE	Université Nancy 2
Co-directeur de thèse :	Alain BOUCHER	Institut de la Francophonie pour l'Informatique

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

Loria

Mis en page avec la classe thloria.

Remerciements

Je remercie vivement M. Karl Tombre, directeur de l'INRIA Lorraine, qui était chef de l'équipe QGAR, de m'avoir accueillie dans son équipe pour mon stage de master puis pendant les trois années de thèse. Je le remercie pour son accueil cordial ainsi que pour son support pour m'avoir permis d'intégrer dans son équipe.

Je tiens à remercier Jean-Marc Ogier et Josep Llados qui ont accepté d'être mes rapporteurs, ainsi que Pierre Héroux, Marie-Dominique Devignes, Salvatore-Antoine Tabbone et Alain Boucher qui ont accepté de faire partie de mon jury de thèse.

J'exprime toute ma reconnaissance à Salvatore-Antoine Tabbone, mon directeur de thèse, qui a consacré tant de temps et de patience durant mon stage master ainsi que les trois années de thèse. Il m'a beaucoup aidé à orienter ce travail et m'a encouragé pendant les périodes difficiles. Je le remercie non seulement pour ses précieux conseils mais aussi pour son soutien qui m'a permis d'effectuer cette thèse dans les meilleures conditions de travail.

Je voudrais remercier Alain Boucher, mon co-encadrant à l'IFI -Vietnam, pour ses conseils scientifiques ainsi que sa disponibilité malgré la distance. Les discussions avec lui me permettent de mieux comprendre la recherche et d'avoir plus de confiance en moi.

Je souhaite adresser mes remerciements sincères à Patrick pour sa gentillesse et pour le temps énorme qu'il a consacré à lire mon manuscrit et à corriger mon français. Merci également à Hervé pour ses corrections et ses conseils qui m'ont aidé à améliorer ce manuscrit.

Mes remerciements vont aussi à tous les membres de l'équipe QGAR pour leur amitié, leur aide durant mes séjours au sein de l'équipe. Merci à tous mes collègues avec qui j'ai partagé le bureau pour la bonne ambiance qu'ils ont apporté tout au long de cette thèse. Merci à mes collègues à MSI-IFI (Vietnam) pour l'ambiance amicale qu'ils m'ont offert durant mes séjours au Vietnam.

Merci beaucoup à tous mes amis qui, de près ou de loin m'ont aidé et encouragé aux moments opportuns. Je les remercie pour tout le temps précieux que nous avons passé ensemble.

J'adresse enfin mes profonds remerciements à ma famille, particulièrement à mes parents et à mes soeurs qui sont toujours à côté de moi et prêts à me soutenir.

Résumé

Cette thèse s'inscrit dans le domaine de la recherche d'images par le contenu et plus spécifiquement dans celui de l'analyse de documents.

Nous abordons le problème complexe de la localisation de symboles dans les documents où les symboles ne sont pas isolés de leur contexte. Bien qu'il existe beaucoup de travaux visant à la définition de bons descripteurs pour la représentation d'un symbole, ces derniers ne peuvent généralement pas être utilisés directement pour localiser des symboles dans les documents car on se heurte au paradoxe suivant : pour reconnaître les symboles il faudrait au préalable segmenter le document et réciproquement pour bien segmenter il faudrait au préalable reconnaître le contenu du document.

Dans ce contexte, nous présentons nos contributions pour la localisation de symboles dans les documents graphiques où le problème de la localisation est abordé d'un point de vue différent de la plupart des méthodes existantes dans la littérature. Dans le contexte de l'analyse de documents graphiques, pour le problème de la localisation de symboles, presque toutes les études se focalisent sur l'aspect structurel du document, ce qui nécessite de résoudre plusieurs autres problèmes difficiles qui se situent soit en amont de la chaîne de traitements telle la vectorisation soit en aval telle la détection d'isomorphisme de (sous-) graphes. Cette thèse tente de voir ce problème de localisation sous l'aspect pixelaires qui est très rarement abordé dans les travaux précédents. Ainsi, dans nos travaux, nous avons abordé deux points essentiels pour résoudre ce problème. Le premier concerne le choix d'une représentation des informations des images de documents et le second est lié au processus de localisation de ces symboles.

Afin de décrire les symboles, nous proposons un descripteur de formes qui s'adapte bien aux symboles graphiques et qui peut être étendu pour décrire le contenu des documents entiers ayant des symboles non-segmentés. Ce descripteur est basé sur le contexte de formes et prend en compte des informations associées aux seuls points d'intérêt associés à une forme. Le descripteur proposé assure l'invariance à la rotation et au changement d'échelle. Il est également tolérant à la déformation et à l'occultation partielle de l'objet.

La localisation de symboles dans les documents graphiques s'appuie sur les techniques de traitement des documents textuels grâce à la notion de *mots visuels*. Un vocabulaire visuel est construit à partir d'un classifieur non-supervisé sur la base d'informations issues du descripteur de formes proposé et étendu aux documents entiers. Les documents graphiques sont ainsi *"textualisés"* grâce au vocabulaire visuel avec une technique d'appariements multiples. Lors de la localisation, les régions candidates sont identifiées dans les documents en fonction de l'appariement local entre la requête et les documents. La détermination des régions, parmi les régions candidates, contenant les occurrences du symbole requête est opérée à l'aide d'un système de vote adaptant le modèle vectoriel usuellement utilisé en recherche d'informations.

Bien que la méthode ne soit pas encore validée sur les documents réels, les expérimentations sur des documents synthétiques et la comparaison avec une autre méthode montrent la performance de la méthode proposée en termes de précision, rappel.

Mots-clés: localisation de symboles, symboles graphiques, mots visuels, descripteur de formes, documents graphiques.

Symbol Spotting in Graphical Documents

Abstract

This thesis addresses the complex problem of symbol spotting in graphical documents where symbols are not segmented a priori. Many works have been proposed to define good descriptors for isolated symbol representation. However, they cannot be directly used to locate symbols in documents because of the recognition/segmentation paradox : to recognise symbols, documents should be segmented first and vice versa, to well segment documents its content (symbols) should be recognised in advance.

In this context, we present our contributions on the symbol spotting problem for graphical documents. This problem is addressed under a viewpoint which is rarely explored in the literature. In fact, most of the existing symbol spotting methods focus on structural aspect and require solving difficult and related problems in the pre-processing step such as the vectorisation or in the detection step like the graph matching. Here, we approach the symbol spotting problem directly from the pixels point of view. There are two essentials points to be addressed. The first concerns the choice of a appropriate shape descriptor to represent document content. The second refers to the process of finding a query symbol in documents.

For describing symbols, a shape descriptor is proposed which is well-suited to graphic symbols and can be adapted to whole documents with non-segmented symbols. This descriptor is defined on the Shape Contexts using only the information associated to interest points. The proposed descriptor is invariant under rotation and scaling. It is also robust to deformations and partial occlusions of objects.

Our symbol spotting approach is based on text retrieval techniques using the concept of "visual words". A visual vocabulary is built on information extracted from entire documents using an extension of the proposed descriptor. An unsupervised clustering algorithm is applied on the computed descriptors to create a set of visual words. The descriptor/visual words assignment is achieved by a fuzzy matching technique. In the spotting process, regions of interest are identified according to the local matching results between the query symbol and documents. The vector model is adapted and applied on these regions to determine the regions containing occurrences of the query.

The method has not yet been evaluated on real documents, however, our experiments on synthetic ones show that the proposed method has good performance in terms of precision and recall.

Keywords: symbol spotting, graphical symbols, visual words, shape descriptors, graphical documents.

Table des matières

Table des figures

Chapitre 1

Introduction Générale

Ces dernières décennies ont vu l'explosion de la capacité de stockage des ordinateurs. Cette explosion a permis de multiplier les ressources numériques et ainsi de faciliter de nombreuses tâches. Cependant, l'accès aux ressources souhaitées au sein d'une grande base n'est pas une tâche facile. Se plaçant dans cette perspective, les méthodes de recherche d'informations permettent de trouver l'information pertinente dans de grandes bases de données. Les premiers travaux en recherche d'informations se sont surtout intéressés à la recherche de documents correspondant à un ensemble de mots clefs donnés. Ces mots clefs sont généralement en lien direct avec le contenu des documents. De nombreuses techniques de recherche d'informations textuelles ont ainsi été proposées avec succès. Transposer de telles techniques dans le cadre de documents visuels tels que les images ou les vidéos pose de nombreux problèmes. En effet, en s'appuyant sur l'annotation des documents, le résultat dépend fortement de la qualité d'annotation et de la subjectivité des personnes réalisant les annotations. C'est pourquoi, la recherche par des requêtes visuelles devient de plus en plus importante lors de la recherche d'images et de vidéos. On parle ainsi de systèmes de *recherche d'images par le contenu (CBIR*[1]*)* ou de *recherche de vidéos par le contenu (CBVR*[2]*).*

Notre thèse s'inscrit dans le domaine de la recherche d'images par le contenu et plus particulièrement dans celui, plus spécifique, de l'analyse de documents. Nous abordons dans cette thèse le problème de la localisation de symboles dans les *documents graphiques*. Typiquement, nous nous intéressons à la recherche d'éléments graphiques dans des documents graphiques tels que des plans architecturaux ou électriques.

1.1 Notions de *symbole* et de *document graphique*

Notre objectif est de proposer une approche de localisation de symboles dans les documents graphiques. Une première question à se poser sur cet objectif est que désigne le terme *symbole* ? Il n'existe pas de définition formelle de ce terme car celui-ci peu regrouper différentes significations en fonction du domaine dans lequel il est employé. De même, ce terme comprend un part

[1]Content-Based Image Retrieval
[2]Content-Based Video Retrieval

de subjectivité. Dans le cadre d'une analyse des plans architecturaux, [Dosch 00] a défini un symbole comme *une représentation simple et compacte d'éléments à laquelle est associée une forte connotation sémantique pour un type de document donné*. Dans sa thèse [Locteau 08], Locteau a défini un symbole par une extension de la définition proposée par Dosch. Un symbole est considéré comme *une représentation graphique :*

- *issue de l'assemblage géométrique de primitives graphiques (arcs de cercles, segments, formes géométriques simples),*
- *simple et compacte,*
- *à laquelle une connotation sémantique est associée,*
- *qui se suffit à elle-même,*
- *et dont la segmentation est non ambigüe.*

Dans le cadre du travail de cette thèse, nous ne chercherons pas à proposer une nouvelle définition du terme de symbole, nous nous contenterons de considérer qu'un symbole est une représentation graphique qui porte un sens particulier propre au type de document concerné.

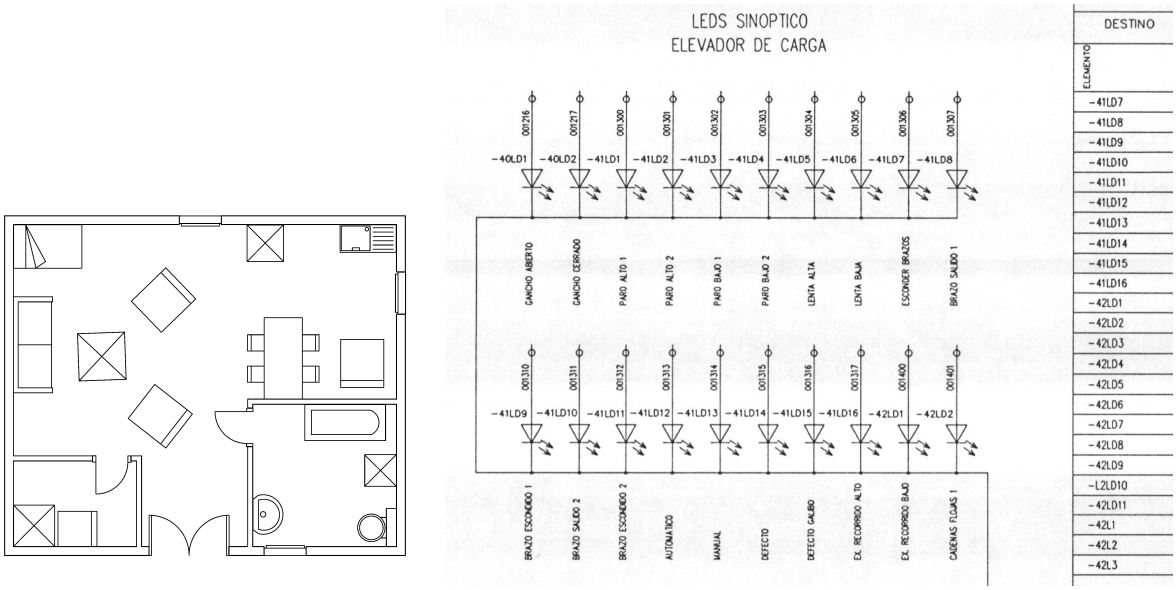
Dans le contexte de l'analyse de documents, le terme "document graphique" désigne des images de documents techniques. Ces documents se composent de symboles connectés selon des règles propres au domaine concerné. Il existe de nombreux types de documents graphiques tels que les *plans électriques*, les *documents mécaniques*, les *cartes géographiques*, les *plans cadastraux*, les *plans architecturaux*, les *plans de câblage téléphoniques*, etc. [Dosch 00]. Quelques exemples de documents graphiques sont montrés en Fig. 1.1. Chaque type de document possède ses propres caractéristiques. Généralement, ces documents nécessitent une étape préalable de binarisation avant tout autres traitements et le résultat de la binarisation a un impact sur les traitements postérieurs. En particulier lorsqu'il s'agit de séparer le document en différentes couches : les principales étant le texte et le graphique. Lorsque le texte touche le graphique (cf. Fig. 1.2), la binarisation devient difficile et cela nécessite des post-traitements dans une étape de séparation.

Dans le cadre de cette thèse, nous n'avons pas l'ambition d'apporter une proposition générique à tous les types de documents, mais de réduire le nombre de contraintes ou de traitements sous-jacents. Ainsi, si les expérimentations menées dans ce travail ont porté sur des plans architecturaux et des plans électriques, notre approche pourrait convenir pour le traitement d'autres types de documents graphiques.

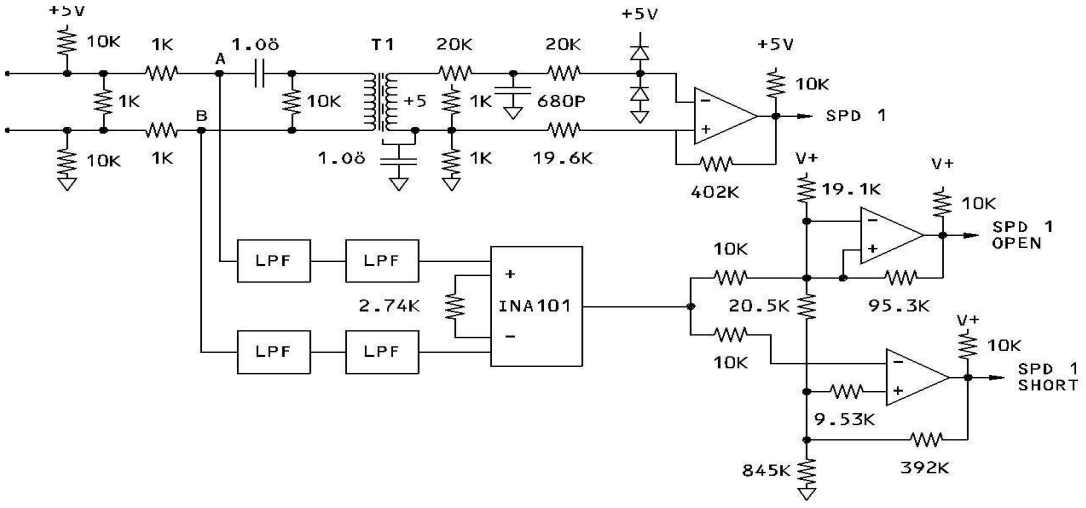
1.2 De la recherche d'informations à la localisation de symboles dans les documents graphiques

Il n'existe pas, à notre connaissance, de définition formelle de la *localisation de symboles*[3]. La localisation d'un symbole dans des documents vise à déterminer s'il existe ou non des occurrences du symbole dans ces documents, et le cas échéant, quelles sont leurs positions exactes. Elle peut être considérée à la fois comme un problème de détection et de reconnaissance de symboles. Cependant, dans la littérature concernant le traitement des symboles, de nombreuses méthodes proposées visent uniquement à résoudre le problème de la reconnaissance de symboles isolés en supposant que ces symboles sont bien segmentés du reste du document. Ainsi, la

[3]Traduit de "symbol spotting"

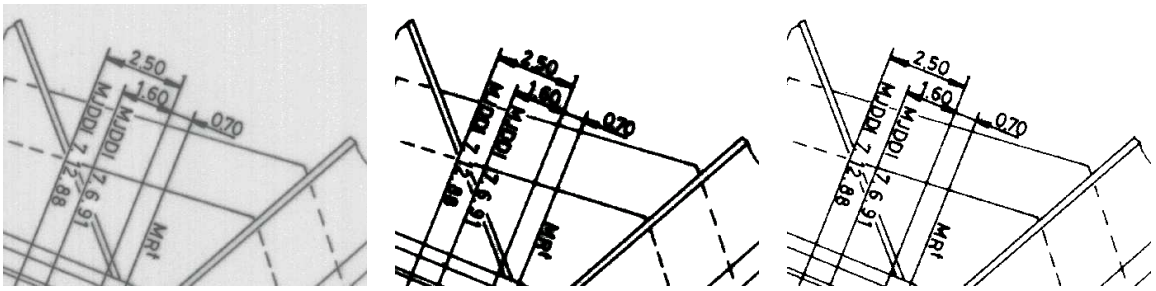


(a) plan architectural (b) plan électronique



(c) plan électrique

FIG. 1.1 – Exemples de différents types de documents graphiques.



(a) Image originale

(b) image binaire obtenue par la méthode de [Otsu 79] avec seuil global

(c) image binaire obtenue par la méthode de [Tabbone 06a] en découpage quaternaire

FIG. 1.2 – Différents résultats de la binarisation.

plupart des travaux sur le problème de la localisation suit un processus commun. La première étape de ce processus consiste généralement en une étape de segmentation des documents en régions contenant probablement des symboles. Cette étape est alors suivie par une étape de reconnaissance ou de recherche de symboles similaires avec un symbole requête. Ainsi, le résultat final de la localisation ne dépend pas seulement de la méthode de reconnaissance mais également de la segmentation des documents en régions. En fait, chaque document peut être décomposé en primitives qui sont ensuite regroupées en régions pouvant potentiellement contenir des symboles. Ces regroupements nécessitent donc des hypothèses sur les symboles traités. La conséquence est que certains symboles ne peuvent pas être appréhendés. De plus, les techniques classiques de segmentation de documents comme la vectorisation ont montré leurs faiblesses. Les résultats obtenus se dégradent rapidement lors de la présence du bruit. Dans ce cas, il peut être intéressant d'explorer la piste consistant à résoudre le problème de localisation sans avoir à pré-segmenter les documents.

Bien que la localisation de symboles soit considérée comme une sous-branche du domaine de la recherche d'images par le contenu, et plus largement du domaine de la recherche d'information, rares sont les travaux en localisation de symboles proposant de s'appuyer sur des techniques de ces domaines. Or, ces deux domaines sont riches en techniques pouvant apporter de nombreux avantages par rapport aux techniques couramment employées en localisation de symboles. C'est donc dans cette direction que nous proposons de placer cette thèse. Nous nous intéresserons ainsi au travers de cette thèse à deux aspects clefs de la localisation de symboles que sont la représentation du symbole et du contenu de l'image et le processus de localisation de symboles dans des documents graphiques.

1.3 Contributions

Les contributions de cette thèse se focalisent sur deux points principaux du problème de la localisation de symboles.

– La première contribution concerne la définition d'un descripteur pour représenter des symboles graphiques. Ce descripteur se base sur le contexte de forme. Il est invariant à la

translation, au changement d'échelle et à la rotation. Il est également robuste aux occlusions partielles. Ce descripteur peut être utilisé pour représenter des informations locales dans les images de documents.
– La deuxième contribution réside dans notre solution proposée pour résoudre le problème de localisation en adaptant une technique étudiée dans le domaine de la recherche d'informations textuelles aux documents graphiques. La méthode proposée n'impose pas de contraintes particulières sur les symboles, ni de pré-traitements "lourds" comme la vectorisation.

1.4 Plan de la thèse

Afin de clarifier nos contributions et les travaux liés, notre manuscrit se divise en quatre chapitres principaux.

Le chapitre 2 est dédié à un état de l'art des descripteurs de formes. Nous ne prétendons pas présenter un état de l'art exhaustif de ces descripteurs mais juste un aperçu des différentes approches existantes pour la description de formes. Notre objectif est ici de mettre en évidence certaines pistes pouvant être utilisées pour la représentation de nos données (symboles et documents graphiques).

Nous présentons ensuite, dans le chapitre 3, un panorama des méthodes de localisation de symboles dans les documents graphiques. Nous reviendrons dans ce cadre sur leurs principes, leurs avantages ainsi que leurs limites.

Dans le chapitre 4, nous aborderons notre première contribution sur la représentation de symboles graphiques. Le descripteur proposé est testé sur quelques bases de symboles isolés et comparé avec d'autres descripteurs de formes performants. Ce descripteur est une adaptation du descripteur *contexte de forme* aux points d'intérêt.

Nous présentons ensuite, dans le chapitre 5, notre deuxième contribution sur l'approche de localisation de symboles dans les documents graphiques que nous proposons. Dans ce chapitre, nous décrirons une extension du descripteur que nous avons proposé précédemment afin de lui permettre de décrire des informations locales dans les documents graphiques. Ensuite, nous évoquerons le processus de localisation de symboles que nous proposons. Celui-ci est basé sur des techniques de recherche d'informations textuelles et sur la notion de *mots visuels* que nous avons adaptés aux documents graphiques. Nous présentons enfin, dans ce chapitre, différents tests menés sur notre approche.

Le mémoire se conclut par une synthèse des travaux que nous avons réalisés dans le cadre de cette thèse. Les points forts et les points faibles des propositions sont analysés et des perspectives sont présentées.

1.5 Publications

Cette thèse a donné, à l'heure actuelle, les publications suivantes.

- Thi-Oanh Nguyen, Salvatore Tabbone, Alain Boucher & O. Ramos-Terrades. *Une approche de localisation de symboles non-segmentés dans des documents graphiques.* À paraître : Traitement du Signal, 2009.
- Thi-Oanh Nguyen, Salvatore Tabbone & Alain Boucher. *A Symbol Spotting Approach Based on the Vector Model and a Visual Vocabulary.* In The 10th International Conference on Document Analysis and Recognition - ICDAR'09, Barcelona, Spain, July 2009.
- Thi-Oanh Nguyen, Salvatore Tabbone & Oriol Ramos-Terrades. *Symbol descriptor based on shape context and vector model of information retrieval.* In The 8th IAPR International Workshop on Document Analysis Systems - DAS'08, pages 191–197, Nara, Japan, September 2008.
- Thi-Oanh Nguyen, Salvatore Tabbone & Oriol Ramos-Terrade. *Proposition d'un descripteur de formes et du modèle vectoriel pour la recherche de symboles.* In Colloque International Francophone sur l'Ecrit et le Document - CIFED'08, pages 79–84, Rouen, France, 2008.

Chapitre 2

État de l'art des descripteurs de formes

Sommaire

Un élément clef de toutes applications de vision par ordinateur 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 de ces informations ont été proposées dans la littérature [Zhang 04, Mikolajczyk 05, Tuytelaars 08]. Le choix d'une méthode de représentation est généralement lié au type d'application considérée. Le contexte de cette thèse concerne les images de documents graphiques qui sont généralement binaires ou en niveaux de gris et contiennent des symboles ayant des orientations et échelles différentes. Les descripteurs de formes doivent bien s'adapter à ce type d'images. Nous proposons, dans ce chapitre, un état de l'art sur des descripteurs de formes.

Un descripteur est défini comme la connaissance utilisée pour caractériser l'information contenue dans les images. Cette connaissance peut être acquise à partir d'études, d'expériences ou d'enseignements. Il existe dans la littérature de nombreuses méthodes de construction de descripteurs qui peuvent être divisées en plusieurs catégories suivant des classifications différentes. La combinaison de différents critères permet d'établir des classifications plus détaillées [Pavlidis 78, Vernon 91]. Nous en présentons ci-dessous quelques-unes :

– Classification basée sur la source d'informations extraites. Il est possible de distinguer les descripteurs définis à partir des *contours* de ceux définis à partir des *régions* [Pavlidis 78].

- Classification basée sur la capacité de reconstruction d'images à partir des informations extraites. Les descripteurs sont classés en deux classes, l'une contenant des informations *préservées* et l'autre contenant des informations *non-préservées* [Pavlidis 78].
- Classification basée sur la région pour laquelle le descripteur est extrait. Les descripteurs sont ainsi classés en tant que descripteurs *locaux* ou *globaux*.
- Classification basée sur la façon dont sont organisées les données [Lladós 02]. Les descripteurs peuvent être divisés en descripteurs *structurels* ou *statistiques*.
- Classification proposée par Terrades et al. [Terrades 07] qui est fondée sur trois critères : les primitives (descripteurs *1D* ou *2D*), les méthodes d'extraction, la structure des descripteurs (*multi-résolution* ou *structurel*).

Dans cette partie, nous proposons de reprendre la classification proposée par [Lladós 02]. Suivant cette classification, les descripteurs sont divisés en deux grandes catégories : les descripteurs structurels (présentés en sous-partie 2.1) et les descripteurs statistiques (présentée en sous-partie 2.2).

2.1 Descripteurs structurels

Dans les approches structurelles, les symboles sont généralement représentés par des ensembles de primitives reliées par des relations. Cette représentation permet de représenter l'organisation spatiale des différentes parties de la forme. Une primitive peut être un coin, un segment, une courbe, une région (composante connexe, région de voronoï, ..) ou une forme simple. Les relations existantes entre primitives sont fréquemment représentées par des graphes ou des chaînes respectant des règles spécifiques de syntaxe (approches syntaxiques [Fu 74]).

Les approches syntaxiques se basent sur la théorie du langage formel [Fu 74]. L'idée générale est qu'une forme peut être décomposée en une séquence de primitives comme une phrase en une suite de mots. Une forme est donc décrite par une chaîne de primitives (préalablement définies) qui respecte un ensemble de règles de syntaxe. Soit un ensemble de primitives prédéfinies et G un ensemble de règles (dite grammaire G), le langage engendré par G, L(G), permet de décrire une classe de formes. Les problèmes de la classification ou la recherche de formes similaires sont considérés comme des problèmes d'analyse syntaxique des représentations des formes afin de déterminer si ces formes partagent la même grammaire.

La Fig. 2.1 présente un exemple très simple dans lequel deux primitives sont définies (Fig. 2.1(b)). L'ensemble des rectangles (de tailles différentes) est décrit par le langage $L = \{a^m b^n a^m b^n | m, n \geq 1\}$. Il est produit à partir d'une grammaire définie sur deux mots $\{a, b\}$. Un carré sera représenté avec cette grammaire par $\{abab\}$.

Bien que ces approches aient pour objectif de simuler la vision humaine, elles s'avèrent peu utilisables pour les applications génériques [Zhang 04]. En effet, l'utilisation de ces approches doit faire face à plusieurs difficultés : limite de la capacité d'expression, difficulté de la construction d'une grammaire à partir d'exemples, non-adaptation pour l'apprentissage, etc. [Tanaka 95]. Il est donc aussi difficile des les rendre invariantes aux transformations. De plus, l'utilisation de ces descripteurs nécessite d'avoir des connaissances a priori sur la base de données afin de pouvoir

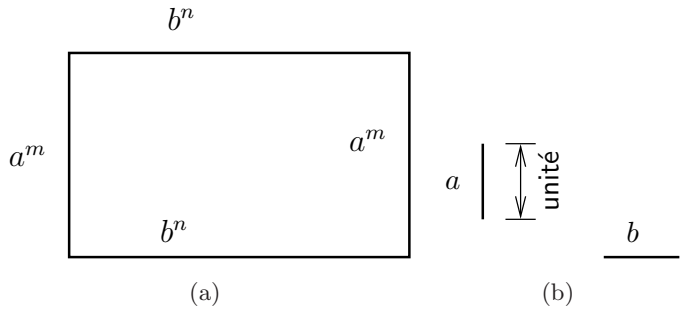
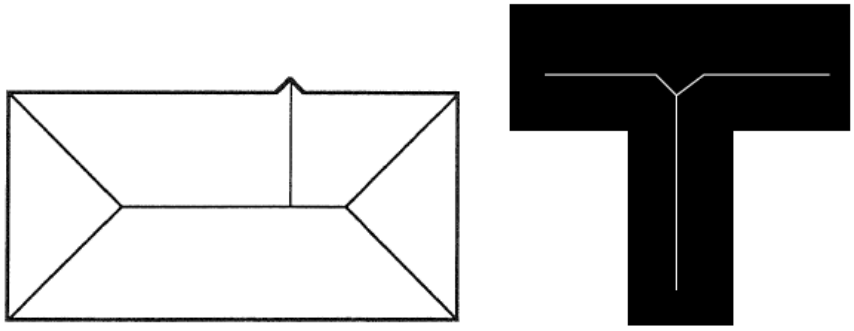


FIG. 2.1 – Exemple d'une représentation d'un rectangle par des primitives. (a) Un rectangle. (b) Deux primitives a, b.

définir des primitives. Elles sont souvent utilisées pour définir des formes complexes tels que les partitions musicales [Baumann 95, Couäsnon 96]. Ces méthodes sont souvent combinées avec d'autres représentations tels que le graphe et/ou les informations statistiques [Bunke 82, Fu 86, Baumann 95] afin de renforcer la performance.

Contrairement aux approches basées sur la théorie du langage formel, les approches se basant sur les graphes n'ont pas besoin de définition préalable des primitives : elles sont directement extraites de la base de données. Ainsi ces approches utilisent des techniques d'approximation polygonale, de vectorisation, d'extraction de squelettes, etc. afin de définir des primitives au préalable de la construction d'un graphe. Dans le cadre de ces approches, la similarité entre deux formes est obtenue par une mise en correspondance de leurs graphes respectifs [Ullmann 76, Cordella 00, Cordella 01, Conte 04]. La similarité entre deux formes est devenue le problème de la recherche des (sous-)isomorphismes qui est un problème NP-complet. Pour réduire la complexité, plusieurs approches tentent de représenter des graphes sous forme de vecteurs numériques ou de signatures vectorielles [Dosch 04, Zhang 07, Sidère 08, Jouili 09].

Le squelette (ou l'axe médian) est l'une des primitives les plus utilisées pour construire ces descripteurs. Elle permet de donner une première caractérisation de la topologie de l'objet. Un point de la forme appartient à l'axe médian si et seulement si il est au centre d'un cercle tangent aux contours de la forme en deux points non-adjacents. Le MAT (*Medial Axis Transform*) représente une forme par un ensemble de couples composés du centre de ces cercles et de leur rayon [Blum 67]. Dans le cadre d'une représentation par un graphe, le squelette peut être décomposé en segments. Cette représentation permet la reconstruction ultérieure de la forme. Néanmoins, cette transformée de l'axe médian est très sensible à la distorsion locale (Fig. 2.2(a)) et au niveau des points de jonctions (Fig. 2.2(b)). Le SLS (*Smoothed Local Symmetries*) [Brady 84] utilise aussi une variation de l'axe médian comme primitive pour décrire une forme. L'axe de symétrie locale est défini par l'ensemble des points correspondant au milieu des segments reliant les couples de points (p, q), dits localement symétrique, où p, q sont deux points de contours symétriquement opposés par rapport à l'axe de symétrie. L'intérêt du SLS est qu'il représente la forme à partir d'informations provenant à la fois du contour et de la région. Il se compose de l'ensemble des axes de symétrie locale (crêtes), d'une description des contours (les paramètres des courbes ainsi que les primitives concernant la discontinuité de courbure) et de celle des régions localement symétriques (la mesure de la région, la courbure de sa crête). Le SLS



(a) sensibilité à la déformation locale (reprise de [Vernon 91])

(b) déformation au squelette aux points de jonctions (reprise de [Tabbone 05b])

FIG. 2.2 – Problèmes liés de la squelettisation.

peut également être utilisé dans le cadre de formes complexes qui sont alors considérées comme un ensemble de parties localement symétriques [Vernon 91] (Fig. 2.3).

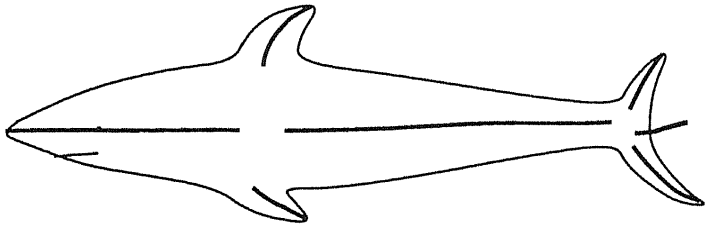


FIG. 2.3 – SLS pour représentation de la forme du poisson (reprise de [Vernon 91]).

Le squelette, l'axe de symétrie locale ou d'autres primitives de bases telles que le contour sont décrits par des chaînes de pixels. Ainsi, afin d'avoir une représentation de plus haut niveau que les pixels comme par exemple une structure de graphe, il est nécessaire de vectoriser pour reconstruire des segments de droite ou des arcs [Rosin 97, Rosin 03, Hilaire 06]. Des formes plus complexes peuvent être représentées par une composition de ces segments (appelées poly-lignes[4]) grâce à une technique d'approximation polygonale (appelées également vectorisation). En plus des erreurs engendrées par la squelettisation de l'image, la vectorisation pose des problèmes sur l'invariance des descripteurs. Un même symbole à tailles différentes n'a pas toujours la même représentation sous forme de vecteurs. De plus, la vectorisation est souvent tributaire du point de départ pour les courbes fermées (Fig. 2.4) [Tabbone 05b].

L'enveloppe convexe est également une primitive utilisée pour décrire les formes [Sonka 99]. Une région est dite convexe lorsque, pour tout couple de points x_1, x_2 tiré de cette région, le *segment* x_1x_2 est entièrement contenu dans la région. L'enveloppe convexe E d'un objet S est définie comme la région convexe la plus petite qui le contient. La différence $D = E \setminus S$ est appelée *résidu concave* (*"convex deficiency"*). Ainsi, l'objet peut être représenté par un arbre

[4]Traduit de "polyline"

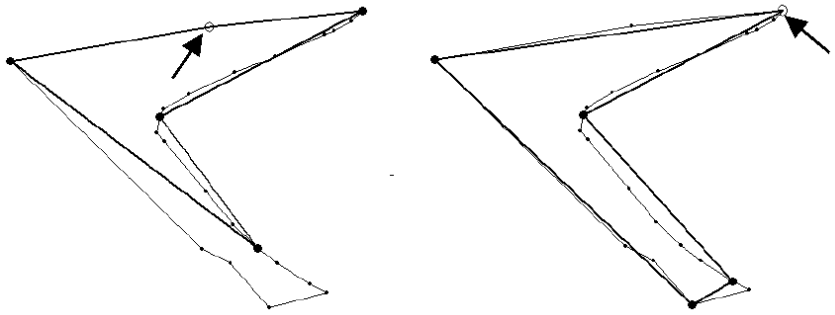
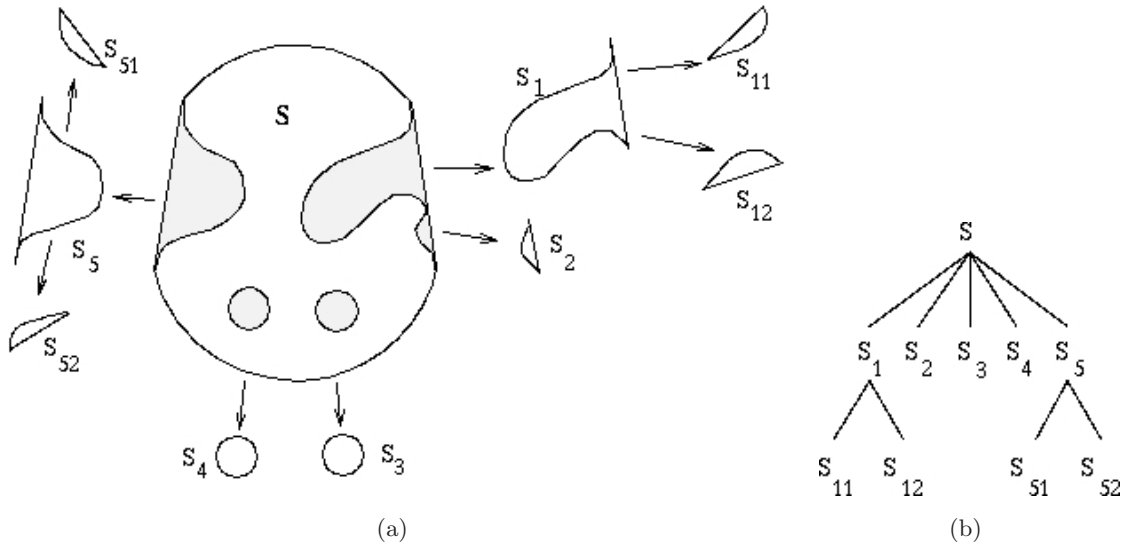


FIG. 2.4 – Vectorisation d'une courbe fermée avec une même méthode pour deux points de départ différents (reprise de [Kolesnikov 03]).

de concavités construit simultanément à son enveloppe convexe. Cet arbre est construit par un processus récursif (voir Fig. 2.5). Dans une première étape, l'enveloppe convexe de l'objet ainsi que ses résidus concaves sont calculés. On calcule ensuite, dans une deuxième étape, l'enveloppe convexe et les résidus concaves de chaque résidu concave précédemment obtenu. Le processus est réitéré jusqu'à ce que les résidus soient tous convexes.



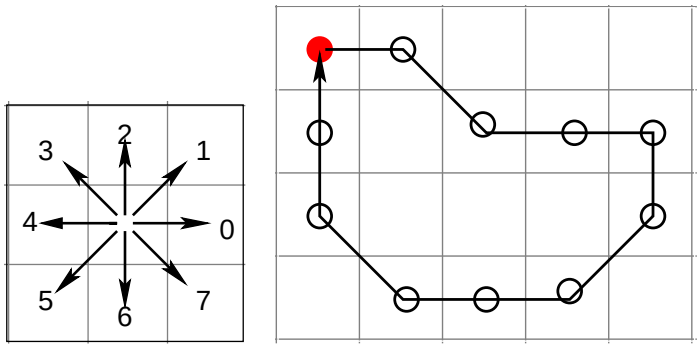
(a) (b)

FIG. 2.5 – Construction de l'arbre de concavités. (a) l'enveloppe convexe et ses résidus concaves. (b) l'arbre de concavités (reprise de [Sonka 99]).

L'enveloppe convexe d'un objet peut être construite par parcours du contour de l'objet [Sonka 99] ou par application d'opérateurs morphologiques [Gonzalez 02]. Dans le cas d'un polygone simple (c'est-à-dire sans intersection), l'enveloppe convexe peut être calculée par un algorithme de complexité $O(n)$ (au lieu de $O(n^2)$), n étant le nombre de sommets du polygone. Ainsi, dans le cas d'un objet dont le contour peut être représenté par un polygone simple, l'approche par approximation de polygone est toujours préférée.

Le contour, quant à lui, fournit une vue intuitive de l'objet. Beaucoup de travaux se basent sur

celui-ci pour décrire une forme. Le BCC (*Boundary Chain Code* ou *Freeman code*) [Freeman 74] décrit un objet par une chaîne de couples segment/direction. Le contour est divisé en segments grâce à une grille, chaque segment est ensuite associé à un code de direction de K-connexité (K = 4, 8, ..., 2i, ...). Les objets sont ainsi représentés par une chaîne de codes dérivée à partir d'un point de départ donné. La Fig. 2.6 donne un exemple de codes de direction correspondant à 8-connexité et un exemple de représentation d'une forme par un BCC. On peut remarquer que les segments dans les directions 1, 3, 5, 7 et ceux dans les directions 0, 2, 4, 6 n'ont pas les mêmes longueurs. Afin de les rendre identiques, il est possible de re-échantillonner le contour de sorte à ce que chaque segment ait une longueur unitaire [Vernon 91]. Le codage BCC est



(a) Codes de 8-connexité (b) BCC : 0 7 0 0 6 5 4 4 3 2 2

FIG. 2.6 – Exemple d'une représentation par BCC.

dépendant du point de départ, de la taille de la grille et n'est pas invariant à la rotation et au changement d'échelles. Des normalisations doivent donc être effectuées afin de faire face à ces problèmes [Lu 97]. Ainsi, on prend souvent comme point de départ celui qui minimise l'entier obtenu par codage BBC de la forme. L'invariance à la rotation est obtenue par utilisation des différences de directions successives dans la chaîne de codes. Cette différence est déterminée en comptant, dans le sens anti-horaire, le nombre de directions qui séparent deux éléments adjacents de la chaîne. La chaîne des différences formant le plus petit entier est appelée *numéro de forme* (*"shape number"*). Afin d'obtenir l'invariance au changement d'échelle, les tailles des formes sont normalisées afin que les axes majeurs aient tous la même longueur et la relation existante entre l'axe majeur et la grille est fixée. Une fois ces normalisations effectuées, le *numéro de forme* d'une forme et la chaîne des codes créée sont uniques et invariants à la rotation et au changement d'échelle. Le descripteur BCC est sensible au bruit et souvent de dimension élevée. Par conséquent, d'autres descripteurs dérivés de BBC sont souvent utilisés pour les processus d'appariement. Le CCH (*chain code histogram* [Iivarinen 96]) en est un exemple. Il est basé sur les probabilités liées aux différentes directions du contour. Le CCH prend la forme d'un histogramme dont chaque élément est calculé par $p_k = n_k/n$ où n_k est le nombre de codes k dans la chaîne, n la longueur totale de la chaîne. Le descripteur CCH est invariant à la translation et au changement d'échelle. La rotation entraîne un décalage circulaire. Ce descripteur permet de réduire la taille de la dimension mais ne résout pas le problème de sensibilité au bruit.

Basé sur la même idée de coder le contour, le descripteur VCC (Vertex Chain Code) [Bribiesca 99] a été proposé pour représenter des formes fermées. Ainsi, le descripteur VCC propose de représen-

ter le contour d'une forme composée de cellules régulières (triangles, rectangles ou hexagones) par une chaîne dont chaque élément indique le nombre de sommets des cellules touchant le contour en un point. Le VCC est invariant à la translation et à la rotation après normalisation du point de départ. Néanmoins il n'est pas invariant au changement d'échelle. Ce descripteur permet de faire des comparaisons locales et ainsi d'effectuer des recherches sur des formes partiellement visibles en utilisant des sous-chaînes de VCC.

Nishida [Nishida 95] propose une représentation structurelle des arcs et des courbes fermées basée sur des caractéristiques $2N$-dimensionnelles. Ainsi, $2N$ directions sont définies dans un espace 2D avec N axes introduits (avec N, un nombre entier). Une courbe est divisée en un ensemble de *sous-segments* (primitives) suivant chacun des N-axes qui sont alors concaténés pour former un *segment* selon la direction de convexité. Chaque segment ainsi créé est représenté par un couple $< rot, idr >$ où rot est le nombre de rotations et idr la direction initiale du segment, c'est-à-dire la direction de la première primitive de la chaîne dans l'espace de $2N$ directions. Deux segments adjacents partageant une même primitive peuvent avoir deux types de connexion : *h-connexion* s'ils partagent la même première primitive ou *t-connexion* s'ils partagent la même dernière primitive. La structure d'une courbe est ainsi décrite par une chaîne de segments et leurs connexions. La Fig. 2.7 présente un exemple de représentation d'un contour fermé. Afin de

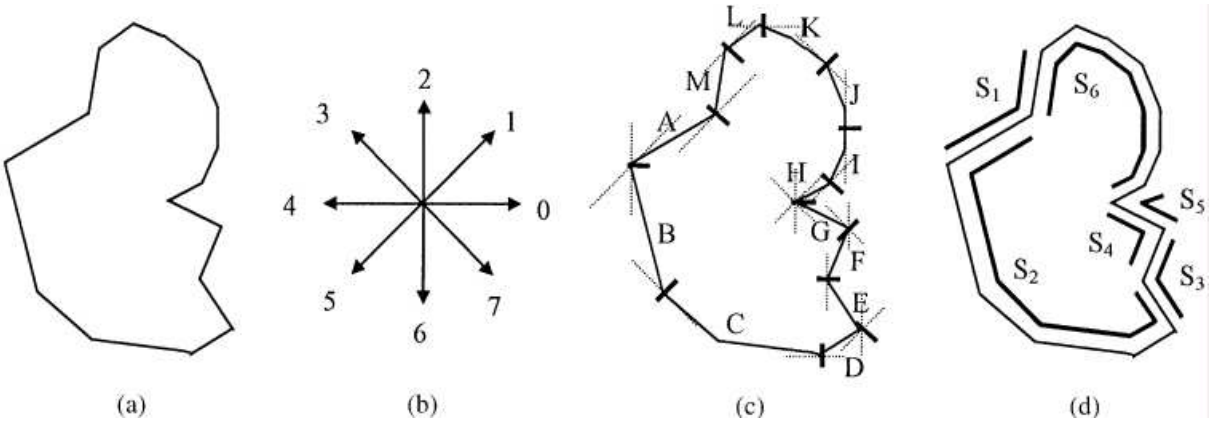


FIG. 2.7 – (a) Approximation polygonale d'un contour fermé. (b) codes des directions quand $N = 4$. (c) sous-segments. (d) segments de la forme lorsque $N = 4$ (reprise de [Nishida 02]).

supporter l'indexation et la recherche de formes similaires, des signatures de formes sont définies par paire de segments adjacents. La signature d'une paire de segments adjacents S_i et S_{i+1} (avec les étiquettes : $< rot_i, idr_i >$ et $< rot_{i+1}, idr_{i+1} >$) concaténés par la connexion c ($c \in \{h, t\}$ est définie par le triplet :

$$\left(rot_i, rot_{i+1}, c, \left\lfloor Q \frac{l_{i+1}}{l_i + l_{i+1}} \right\rfloor \right),$$

où l_i, l_{i+1} représentent les longueurs des segments, Q le nombre de niveaux de quantification pour les paramètres de ratio-longueurs $l_{i+1}/(l_i + l_{i+1})$. Des résultats expérimentaux ont montré l'intérêt de cette représentation pour la recherche de formes partiellement visibles [Nishida 02].

Fonseca et al. [Fonseca 05] propose d'utiliser des polygones comme primitives. Un document technique est alors représenté par un graphe de topologie dans lequel les nœuds sont des

polygones et les arcs des relations topologiques. Les relations topologiques sont choisies afin de permettre l'invariance à la translation et à la rotation. On utilise généralement trois relations topologiques : 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. Cette représentation permet de décrire un document en différents niveaux de détails. La Fig. 2.8 donne un exemple de document décrit selon trois niveaux de détails. Afin de simplifier

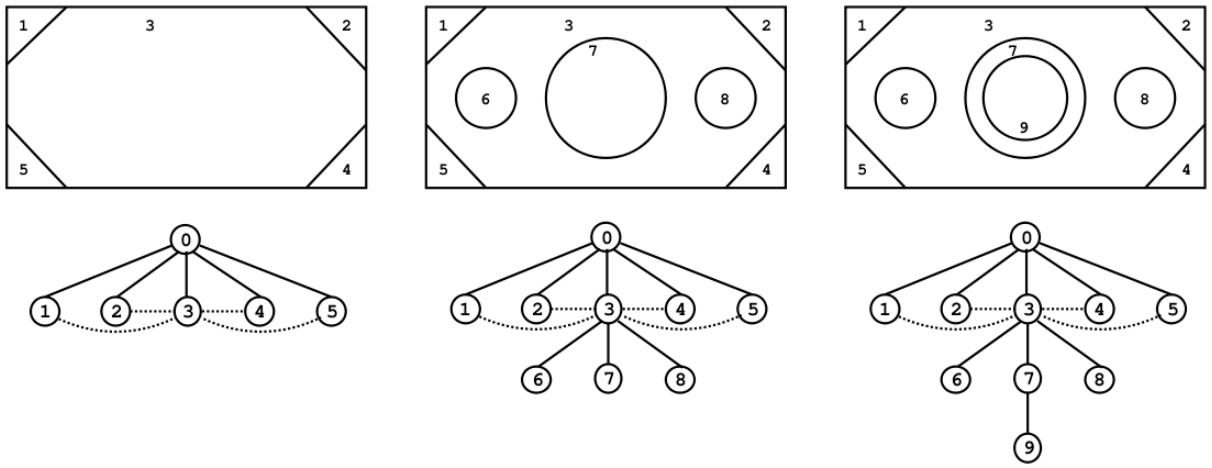


FIG. 2.8 – Différents niveaux de détail avec leur graphe correspondant (reprise de [Fonseca 05]).

le processus d'appariement dans le cadre de recherche ou de reconnaissance des formes, le graphe est souvent traduit sous la forme d'un descripteur topologique. Ce dernier représente le spectre du graphe. Il est calculé à partir des valeurs propres de la matrice d'adjacence du graphe. Des descripteurs géométriques supplémentaires sont utilisés afin d'affiner et de classer les résultats obtenus.

Partant de l'hypothèse que l'utilisateur tend à dessiner un objet à partir des primitives simples, Xiaogang et al. [Xiaogang 04] proposent de décrire un objet par un graphe de relations spatiales (SRG - *Spatial Relation Graph*, voir Fig. 2.9). Ainsi des primitives géométriques (arc, ligne et ellipse) forment les nœuds du graphe. Les relations spatiales entre primitives forment, elles, les arcs du graphe. Elles peuvent être liées à la contiguïté entre deux primitives (interconnexion, tangence, intersection, parallélisme, concentricité) ou à la position (*"BEFORE"* selon quatre directions). La complexité de l'appariement entre deux graphes SRGs peut être réduite par l'utilisation de contraintes et d'appariements locaux.

Les travaux de [Park 00, Park 03] proposent, pour leur part, de décrire une image en utilisant des lignes reliées par des relations binaires comme primitives. La relation entre deux lignes est caractérisée par l'angle qu'elles forment ainsi que par leur distance (rapports des longueurs) (voir Fig. 2.10). Une image est donc représentée par un graphe complet dont les nœuds sont des primitives (lignes) et les arcs des relations binaires (Fig. 2.11). Afin de prendre en compte des informations globales de l'objet et de diminuer la complexité de l'étape d'appariement, [Park 00] propose une représentation modale de l'objet construite à partir de la matrice des relations entre primitives. L'étape d'appariement d'un modèle avec une image consiste alors à sélectionner un ensemble de modèles candidats par des tests préliminaires, à calculer la similarité

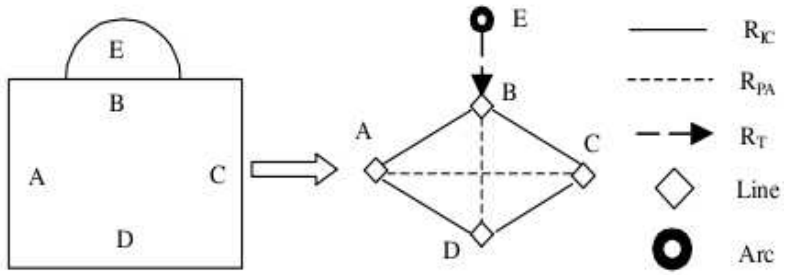


FIG. 2.9 – Exemple d'un objet composé de 5 primitives et sa représentation par SRG avec les relations : R_{IC} (interconnexion), R_{PA} (parallélisme), R_T (tangence) (reprise de [Xiaogang 04]).

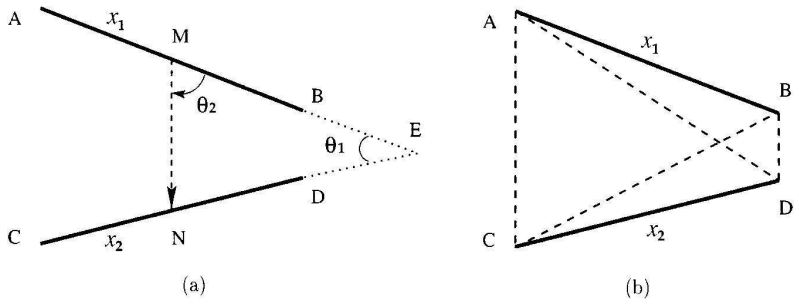


FIG. 2.10 – Relations binaires d'une paire de lignes. (a) relations angulaires (θ_1, θ_2). (b) relations de distance $(\overline{AB}/\overline{CD}$, $(\overline{AB}+\overline{CD})/((\overline{AC}+\overline{AD}+\overline{BC}+\overline{BD})/4)$) (reprise de [Park 00]).

entre le modèle et chaque candidat par utilisation de leur représentation modale, et enfin à sélectionner le candidat ayant la plus forte similarité. Similairement, un objet peut être décrit par un graphe de relations attribuées ARG (*"Attributed Relational Graph"*) [Park 03]. Dans ce cadre, l'appariement entre le graphe du modèle de référence et celui d'une image est réalisé par utilisation d'un ensemble de sous-graphes de candidats extraits du graphe de l'image. La construction de ces sous-graphes est fondée sur l'utilisation d'espaces de vecteurs de relations et sur la mesure de correspondance entre deux primitives. Cette représentation permet de trouver des objets partiellement occultés à partir d'appariements partiels. En effet, l'analyse de l'espace des vecteurs des relations permet la détection de primitives occultées ou bruitées. Malgré le nombre important de paramètres à définir, cette représentation permet d'obtenir de très bons résultats.

L'utilisation aussi des segments pour la réprésentation du symbole, mais afin d'éviter le problème de mise en correspondance des graphes, [Zhang 07] a construit une signature vectorielle pour chaque symbole en se basant sur les relations entre les primitives. Les auteurs ont pris en compte les relations entre ligne-ligne, arc-arc, ligne-arc, ligne-cercle (Fig. 2.12). Ces relations sont mesurées par l'angle et la distance relative entre primitives. La signature vectorielle est invariante à la rotation et au changement d'échelle et elle donne de bons taux de reconnaissance des symboles. Cependant, cette signature a besoin d'une segmentation initial des primitives et n'arrive pas à distinguer quelques paires de modèles (Fig. 2.13).

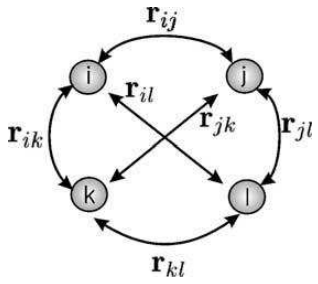
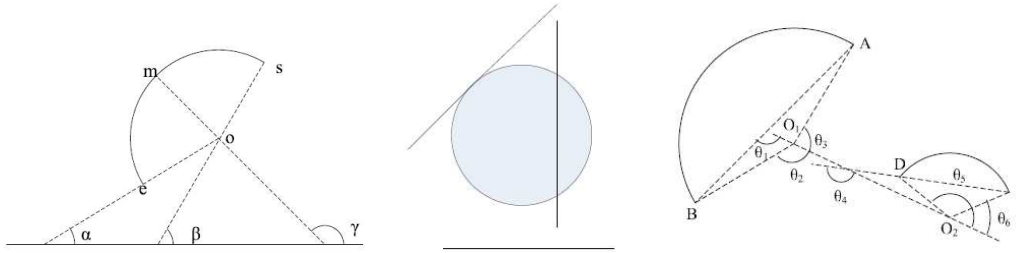


FIG. 2.11 – Représentation d'un modèle par un graphe de relations dont les nœuds sont des lignes et les arcs représentent leurs relations (reprise de [Park 03]).



(a) Relation entre une ligne et un arc

(b) Trois relations possibles entre une ligne et un cercle

(c) Relation entre deux arcs

FIG. 2.12 – Relations entre les primitives (extraite de [Zhang 07]).

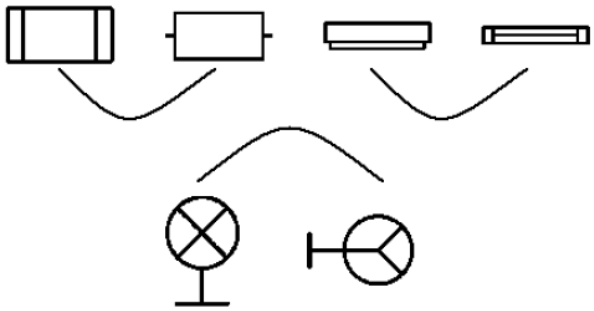


FIG. 2.13 – Paires de modèles qui partagent la même signature vectorielle proposée par [Zhang 07].

2.2 Descripteurs statistiques

2.2.1 Descripteurs géométriques simples

Les descripteurs géométriques simples, utilisés individuellement, ne permettent pas de bien représenter les formes et les objets. Néanmoins, une fois combinés, ils offrent des descriptions plus représentatives des formes et des objets. Nous présentons ci-dessous quelques descripteurs simples les plus utilisés :

- *Le périmètre (P)* : longueur des contours qui forment l'objet.
- *L'aire (A)* : correspond généralement au nombre de points formant l'objet.
- *Le rapport entre la largeur et la longueur du rectangle englobant.*
- *Le nombre et la taille des concavités résiduelles situées à l'intérieur de l'enveloppe convexe de la forme.*
- *La circularité* : A/P^2. Un objet de forme circulaire aura une circularité maximale.
- *La rectangularité* : rapport entre la surface de la forme et celle du rectangle englobant minimal.
$$\frac{A}{\text{Surface du rectangle englobant minimal}}$$
- *L'excentricité (E)* : rapport entre la longueur de l'axe principal et celle de l'axe mineur de l'objet.
$$E = \frac{\text{Longueur de l'axe principal}}{\text{Longueur de l'axe mineur}}$$
- *La convexité (C)* :
$$C = \frac{\text{Périmètre de l'enveloppe convexe}}{P}$$
- *L'énergie de courbure (Bending energy)* :
$$BE = \frac{1}{P}\int_0^P |K(p)|^2 dp$$
avec $K(p)$ la courbure au point p et P la longueur totale de la courbe.

2.2.2 Descripteurs à base des moments

Les descripteurs basés sur les moments font partie de la famille des descripteurs régions, c'est-à-dire des descripteurs définis par analyse des régions. Ces descripteurs sont très utilisés dans le domaine de la reconnaissance de formes [Teh 88, Liao 96, Prokop 92, Zhang 04, Kotoulas 05]. Déterminer les moments d'une image revient à la décomposer en un ensemble de fonctions de bases, pouvant être orthogonales ou non-orthogonales. Soit une image dont la fonction de densité, $f(x, y)$, est bornée, les moments d'ordre $(p + q)$ de cette image sont définis par :

$$m_{p,q} = \int_x \int_y \phi_{p,q}(x, y) f(x, y) dx\ dy,\ \forall p, q \in \mathbb{N} \tag{2.1}$$

avec x, y les coordonnées cartésiennes et $\phi_{p,q}(x, y)$ une fonction de base d'ordre $(p+q)$. Plusieurs familles de fonctions de base ont été proposées. Le choix d'une famille dépend de l'application et

des propriétés désirées d'invariance . Deux grandes familles de fonctions peuvent être distinguées : les *orthogonales* et les *non-orthogonales*.

Moments non-orthogonaux : Les moments géométriques, complexes et rotationels font partie de la famille des moments non-orthogonaux. Les fonctions de base correspondant aux moments géométriques et complexes sont respectivement définies par $\phi_{p,q}(x,y) = x^p y^q$ et $\phi_{p,q}(x,y) = (x+jy)^p(x-jy)^q, j^2 = -1$. Les moments rotationels (appelé aussi moments de Fourier-Mellin, cas particulier de la transformée Fourier-Mellin [Sheng 94]) d'ordre p avec répétition $l \in \mathbb{Z}$ sont déterminés dans le système polaire par :

$$D_{p,l} = \int_0^{2\pi} \int_0^{\infty} r^p e^{-jl\theta} f(r\cos\theta, r\sin\theta)\ r\ dr\ d\theta$$

Parmi ces différents types de moments, les moments géométriques sont les plus utilisés, en particulier leurs dérivés proposés par Hu [Hu 62]. Pour obtenir l'invariance à la translation, Hu a introduit les moments centraux définis par l'équation (2.2). L'invariance au changement d'échelle est obtenue par la normalisation présentée dans l'équation (2.3).

$$M_{p,q} = \int_x \int_y (x-\bar{x})^p (y-\bar{y})^q f(x,y)\ dx\ dy,\ \forall p,q \in \mathbb{N} \tag{2.2}$$

$$\mu_{p,q} = \frac{M_{p,q}}{M_{0,0}^{(p+q)/2+1}},\ \forall p,q \in \mathbb{N}, p+q \geq 2 \tag{2.3}$$

où $(\bar{x},\bar{y}) = (m_{1,0}/m_{0,0}, m_{0,1}/m_{0,0})$ désigne le centroïde de l'image. L'invariance absolue (invariances à la rotation, à la translation, au changement d'échelle et à la réflexion) des descripteurs est obtenue par combinaisons des moments (voir (2.4)). Les descripteurs ainsi obtenus sont connus sous le nom de *moments invariants de Hu*. A l'exception du septième moment qui n'est pas invariant à la réflexion, les six premiers conservent l'invariance absolue.

$$\begin{aligned}
HM_1 &= \mu_{2,0} + \mu_{0,2}, \\
HM_2 &= (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2, \\
HM_3 &= (\mu_{3,0} - 3\mu_{1,2})^2 + (3\mu_{2,1} - \mu_{0,3})^2 \\
HM_4 &= (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{2,1} + \mu_{0,3})^2 \\
HM_5 &= (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2})[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2] \\
&\quad +(3\mu_{2,1} - \mu_{0,3})(\mu_{2,1} + \mu_{0,3})[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2] \\
HM_6 &= (\mu_{2,0} - \mu_{0,2})[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2] \\
&\quad +4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{2,1} + \mu_{0,3}) \\
HM_7 &= (3\mu_{2,1} - \mu_{0,3})(\mu_{3,0} + \mu_{1,2})[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2] \\
&\quad -(\mu_{3,0} - 3\mu_{1,2})(\mu_{2,1} + \mu_{0,3})[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2]
\end{aligned} \tag{2.4}$$

Le calcul des moments non-orthogonaux est peu coûteux en termes de temps de calcul. Néanmoins, ils s'avèrent peu robustes en présence de bruit [Teh 88, Kotoulas 05]. En effet, les moments d'ordres élevés, qui représentent les informations détaillées de l'image, sont plus sensibles au bruit que ceux d'ordres inférieurs. Il est donc nécessaire de définir un ordre optimal

permettant l'obtention d'un compromis raisonnable entre le niveau de détails et la présence de bruit. Un autre problème concerne les larges variations des intervalles dynamiques de valeurs dans le cas d'ordres différents. Ces variations ont pour conséquence des instabilités numériques dans l'analyse d'images de grande taille [Mukundan 01].

Teh et Chin [Teh 88] ont montré que les moments orthogonaux sont meilleurs que les moments non-orthogonaux en termes de redondances des informations et de robustesse au bruit pour les ordres élevés et permettent donc la reconstruction de l'image avec une précision plus élevée.

Concernant les moments orthogonaux, on distingue les moments se basant sur des fonctions continues de ceux se basant sur des fonctions discrètes.

Moments de polynômes orthogonaux continus :

Comme les moments non-orthogonaux présentés précédemment, les fonctions de base des moments orthogonaux sont continues. Néanmoins, ces fonctions assurent l'orthogonalité, ce qui permet d'éviter le problème de redondance d'information. Parmi les moments orthogonaux, les plus utilisés sont les moments de Legendre, de Zernike et de pseudo-Zernike. Proposés par Teague [Teague 80], les moments de Zernike et de pseudo-Zernike sont plus performants que les moments non-orthogonaux et que les moments orthogonaux de Legendre en termes d'erreur de reconstruction pour les images homogènes et bruitées [Teh 88]. De plus, les moments de Zernike offrent l'avantage d'utiliser un module invariant à la rotation ainsi que de contenir des informations compactes dans les premiers ordres. Cependant le calcul de ces moments est de complexité très élevée. De nombreux travaux de recherche ont porté sur l'amélioration de leur performance et sur la diminution de leur complexité de calcul [Belkasim 91, Kim 99, Bober 01, Kotoulas 08, Revaud 09]. Concernant les invariances à la translation et au changement d'échelle, elles peuvent être obtenues soit par une approche de pré-normalisation de l'objet, soit par l'utilisation de moments géométriques [Teh 88, Belkasim 91] ou de moments centraux [Chong 03], ou soit directement à partir d'une représentation cartésienne de l'image [Belkasim 07]. Afin d'ignorer la dépendance à la rotation, on n'utilise généralement que le module des moments lors de l'appariement entre deux images. Récemment, Revaud et al. [Revaud 09] ont proposé une nouvelle méthode de comparaison qui prend en compte les informations angulaires et assure l'invariance à la rotation permettant ainsi de rendre plus précise la mesure de similarité. W-Y Kim et Y-S Kim [Kim 99, Bober 01] ont proposé une amélioration des moments de Zernike pour obtenir un descripteur plus pertinent, le descripteur ART (*Angular Radial Transformation*). Dans ce cadre, des travaux se sont intéressés à la recherche d'algorithme de complexité moindre pour le calcul de l'ART [Hwang 06, Kotoulas 08].

Les moments complexes de Zernike d'ordre $p, p \geq 0$ et de répétition $l, |l| \leq p, (p - |l|)$ *pair* sont définis par :

$$Z_{p,l} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^{\infty} [V_{p,l}(r,\theta)]^* \; f(r\cos\theta, r\sin\theta) \; r \; drd\theta \tag{2.5}$$

où $[V_{pl}(r,\theta)]^*$ désigne le conjugué du polynôme de Zernike $V_{pl}(r,\theta)$ déterminé dans un disque unitaire $x^2 + y^2 \leq 1$.

$$V_{p,l}(r,\theta) = \sum_{s=0}^{(p-|l|)/2} (-1)^s \frac{(p-s)!}{s!(\frac{p+|l|}{2}-s)!(\frac{p-|l|}{2}-s)!} r^{p-2s} \; e^{jl\theta}; \; j^2 = -1 \tag{2.6}$$

Par ailleurs, l'image des polynômes orthogonaux de Zernike et de Legendre sont calculés dans un domaine limite (caractérisé par un disque unitaire pour Zernike et par un carré unitaire pour Legendre [Courant 53]). Afin de pouvoir utiliser ces descripteurs, il est donc nécessaire de procéder à une transformation de l'image sur ces domaines, ce qui pose des problèmes de robustesse dans le cadre de l'invariance au changement d'échelle et à la translation [Tabbone 05b]. On peut noter que les moments présentés précédemment (non-orthogonaux et orthogonaux continus) souffrent d'un problème majeur d'approximation numérique de l'intégral continu car les fonctions de base de ces moments ne sont pas discrètes dans l'espace de l'image. Ce problème entraîne l'apparition d'erreurs numériques lors du calcul des moments et d'analyse des propriétés comme les invariances. Cette limitation est l'une des raisons du succès des moments se basant sur les polynômes orthogonaux discrets.

Parmi d'autres moments orthogonaux continus, on peut citer les moments orthogonaux de Fourier-Mellin [Sheng 94], les moments de Chebyshev-Fourier [Ping 02], les moments de Fourier harmonique [Ren 03] et les moments de pseudo-Zernike généralisé [Xia 07].

Moments de polynômes orthogonaux discrets :

Cette famille de moments permet d'éviter non seulement les problèmes de discrétisation, d'instabilité numérique mais également ceux dûs à la transformation de l'image dans le domaine limite des fonctions de base (disque unitaire pour les moments de Zernike, carré unitaire pour ceux de Legendre). De récentes recherches ont porté sur l'introduction de tels moments pour le traitement d'images. Parmi ces moments orthogonaux discrets, les plus connus sont les moments de Tchebichef [Mukundan 01], de Krawtchouk [Yap 03], de Hahn [Zhou 05] et de Hahn dual (*dual Hahn moments*) [Zhu 07b].

Concernant l'analyse de l'erreur de reconstruction d'images, les résultats obtenus avec les moments de Tchebichef se sont montrés meilleurs que ceux obtenus avec les moments de Legendre. Cependant, les propriétés d'invariance sont complexes à obtenir avec ces moments, particulièrement pour l'invariance à la rotation. Généralement, ces invariances sont obtenues par la normalisation de l'image, ou par l'utilisation de combinaisons linéaires de moments invariants géométriques. Une autre approche, pour obtenir des invariances à la translation et au changement d'échelle, consiste à analyser directement des polynômes de Tchebichef [Zhu 07a]. Soit une image de taille $N \times N$, les moments de Tchebichef d'ordre $(p+q)$, $p, q \in [0, N-1]$ sont définis par :

$$T_{p,q} = \frac{1}{\tilde{\rho}(p,N)\tilde{\rho}(q,N)} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \tilde{t}_p(x)\tilde{t}_q(y) f(x,y). \tag{2.7}$$

où

$$\tilde{\rho}(n,N) = \frac{N(1-1/N^2)(1-2^2/N^2)...(1-n^2/N^2)}{2n+1}, \ n \in [0, N-1]. \tag{2.8}$$

et $\tilde{t}_n(x)$, le polynôme discret de Tchebichef d'ordre n, est défini par une relation de récurrence :

$$\tilde{t}_n(x) = \frac{(2n-1)\tilde{t}_1(x)\tilde{t}_{n-1}(x) - (n-1)(1-(n-1)^2/N^2)\tilde{t}_{n-2}(x)}{n}, \ n \in [2, N-1]. \tag{2.9}$$

avec $\tilde{t}_0(x) = 1$ et $\tilde{t}_1(x) = (2x+1-N)/N$.

L'analyse des paramètres proposée dans [Zhu 07b] montre que les polynômes de Hahn, de Tchebichef et de Krawtchouk sont des cas particuliers des polynômes duals de Hahn. Les moments de Hahn dual sont fondés sur l'utilisation de polynômes duals pondérés de Hahn, leur permettant ainsi d'éviter le problème d'instabilité numérique des moments d'ordre élevé. Les résultats expérimentaux obtenus ont montré que ces moments sont plus performants que ceux de Legendre, de Tchebichef et de Krawtchouk pour la reconstruction d'images à partir de moments d'ordre élevé. Concernant les problèmes d'invariances aux transformations, [Zhu 07b] ont proposé de recourir à une combinaison linéaire de moments géométriques pour dériver des moments invariants à la translation, au changement d'échelle et à la rotation. Une expérimentation concernant la classification d'objets a montré que les moments invariants de Hahn dual sont plus performants que les moments invariants de Hu en termes de précision de la reconnaissance.

Les moments de Hahn dual d'ordre $(p+q)$ pour une image de taille $N \times N$ sont définis par (2.10) :

$$W_{p,q} = \sum_{s=a}^{b-1} \sum_{t=a}^{b-1} \hat{w}_p^{(c)}(s,a,b) \hat{w}_q^{(c)}(t,a,b) f(s,t), \;\; p,q = 0,1,...,N-1 \tag{2.10}$$

où a, b, c sont des paramètres tel que : $-1/2 < a < b$, $|c| < 1 + a$, $b = a + N$ et $\hat{w}_n^{(c)}(s,a,b)$ les polynômes duals pondérés d'ordre n de Hahn :

$$\hat{w}_n^{(c)}(s,a,b) = w_n^{(c)}(s,a,b) \sqrt{\frac{\rho(s)}{d_n^2}[\triangle x(s-1/2)]}, \;\; x(s) = s(s+1) \tag{2.11}$$

$$w_n^{(c)}(s,a,b) = \frac{(a-b+1)_n (a+c+1)_n}{n!} \; {}_3F_2(-n, a-s, a+s+1; a-b+1, a+c+1; 1), \quad s = a, a+1, ..., b-1 \tag{2.12}$$

avec ${}_3F_2(a_1,a_2,a_3;b_1,b_2;z) = \sum_{k=0}^{\infty} \frac{(a_1)_k (a_2)_k (a_3)_k}{(b_1)_k (b_2)_k} . \frac{z^k}{k!}$, $(u)_k = u(u+1)...(u+k-1) = \frac{\Gamma(u+k)}{\Gamma(u)}$, $\triangle x$ l'opérateur de différenciation et

$$\rho(s) = \frac{\Gamma(a+s+1)\Gamma(c+s+1)}{\Gamma(s-a+1)\Gamma(b-s)\Gamma(b+s+1)\Gamma(s-c+1)} \tag{2.13}$$

$$d_n^2 = \frac{\Gamma(a+c+n+1)}{n!(b-a-n-1)!\Gamma(b-c-n)} \tag{2.14}$$

En dépit du nombre important de travaux de recherche ayant porté sur l'étude de ces moments, les problèmes du choix d'un ordre optimal et de l'invariance aux transformations pour les moments d'ordre élevé restent ouverts.

2.2.3 Descripteurs de Fourier

Les Descripteurs de Fourier (DFs) font partie des descripteurs les plus populaires pour les applications de vision par ordinateur et de reconnaissance de formes. En effet, les descripteurs de Fourier sont facilement calculables et facilitent l'étape d'appariement. De plus, ils permettent de décrire la forme de l'objet à différents niveaux de détails. Les descripteurs de Fourier sont calculés à partir du contour des objets. Leur principe est de représenter le contour de l'objet

par un signal 1D, puis de le décomposer en séries de Fourier. Les DFs sont généralement connus comme une famille de descripteurs car ils dépendent de la façon dont sont représentés les objets sous forme de signaux.

Dans le cadre d'objets ayant pour contour une courbe fermée γ de longueur L, Granlund [Granlund 72] et Zahn et al. [Zahn 72] proposent deux approches pour définir le signal représentant l'objet. Ainsi, dans [Zahn 72], Zahn et Roskies proposent de représenter γ par $c(t) = (x(t), y(t))$ où $(x(t), y(t))$ sont les coordonnées du contour de l'objet, t un paramètre de longueur de la courbe défini tel que $0 \leq t \leq L$. En définissant $\theta(t)$ la direction angulaire au point $(x(t), y(t))$, $\delta_0 = \theta(0)$ celle au point initial et $\phi(t)$ une fonction représentant les changements de $\theta(t)$ le long de la courbe, $\phi(t) = \theta(t) - \theta(0)$ permet de décrire la courbe. Afin d'avoir une fonction périodique et invariante à la translation, à la rotation et au changement d'échelle, la fonction $\phi(t)$ est normalisée dans l'intervalle $[0, 2\pi]$:

$$\Phi_N(t) = \phi(Lt/2\pi) + t; \tag{2.15}$$

ainsi, $\Phi_N(0) = \Phi_N(2\pi) = 0$ et $\Phi_N(t)$ peut être décomposée en séries de Fourier :

$$\Phi_N(t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k); \tag{2.16}$$

Les couples $\{A_k, \alpha_k\}$ représentent les modules et les phases des descripteurs de Fourier de la courbe fermée γ. Cependant, la reconstruction à partir de sous-ensembles de coefficients $\{A_k, \alpha_k\}, k = 1, 2, ...$ n'assure pas toujours la fermeture de la courbe. Par conséquent, l'approche de représentation de l'objet en signal utilisant la fonction complexe de Granlund [Granlund 72] est généralement préférée. Granlund propose de représenter la courbe γ par un signal complexe, $u(t) = x(t) + jy(t)$. Dans le cas où γ est une courbe fermée, ce signal est périodique $u(t + iL) = u(t), i = 0, 1, 2, ...$ Il est donc possible de le décomposer en séries de Fourier complexe :

$$u(t) = \sum_{-\infty}^{\infty} FD(s) e^{j2\pi st/L} \tag{2.17}$$

$$FD(s) = \frac{1}{L} \int_0^L u(t) e^{-j2\pi st/L} dt \tag{2.18}$$

Les coefficients de Fourier $FD(s)$ sont utilisés comme descripteurs de forme. En pratique, étant donné que les courbes sont rarement continues, les coefficients de Fourier sont estimés en appliquant la transformée de Fourier discrète (TFD) :

$$FD(s) = \frac{1}{N} \sum_{n=0}^{N-1} u(n) e^{-j2\pi sn/N}, s = -N/2 + 1, .., N/2 \tag{2.19}$$

Cette représentation ne permet pas d'assurer l'invariance des descripteurs de Fourier à la translation, à la rotation, au changement d'échelle et au choix du point de départ. Cependant, les propriétés des coefficients de Fourier permettent facilement d'obtenir l'invariance à la translation et au changement d'échelle. En effet, ces invariances peuvent être obtenues en imposant que $FD(0)$ soit égale à 0 et en normalisant les autres coefficients de Fourier par $FD(1)$ [Shen 94,

Rui 98, Bartolini 05]. La rotation et le changement du point de départ entraînent des décalages dans les phases des coefficients de Fourier. Il est possible de faire face à ce problème soit en ne prenant en compte que les amplitudes des coefficients de Fourier [Zhang 02a, Rafiei 02], soit en définissant des mesures de distance, lors d'étapes d'appariement, indépendantes de la rotation et du point de départ [Rui 98, Bartolini 05], soit en normalisant les phases des coefficients de Fourier [Persoon 77, Bartolini 05].

Dans le cadre de la représentation des objets par un signal, la majeure partie de l'énergie du signal se trouvent dans les basses fréquences, c'est-à-dire dans les coefficients de Fourier proches de zéro. Ces derniers contiennent alors les informations essentielles concernant la forme des objets. Au contraire, les coefficients de Fourier se trouvant loin de zéro fournissent des informations plus détaillées mais également plus bruitées sur la forme. Ainsi, on n'utilise généralement que les informations en basses fréquences (coefficients de Fourier proches de zéro) pour construire les descripteurs de Fourier. Le nombre de coefficients retenus dépend du niveau de détails nécessaire à l'application considérée. Zhang et al. [Zhang 02a] ont montré que 10 coefficients sont suffisants pour décrire une forme et qu'un nombre de coefficients de Fourier supérieur à 60 ne permet pas d'obtenir d'amélioration significative des performances dans le cadre de problèmes de recherche d'images.

Il est possible d'utiliser les descripteurs de Fourier pour des courbes ouvertes. En effet, dans ce cadre, un contour fermé minimal est défini pour la courbe et les descripteurs de Fourier sont calculés à partir de celui-ci [Persoon 77]. Il est, par contre, difficile d'utiliser les descripteurs de Fourier pour les objets ayant des contours non connexes.

2.2.4 Descripteurs à base d'une transformée de l'image

Il existe, en marge des descripteurs directement calculés à partir de l'espace de l'image, un ensemble de méthodes permettant de décrire des objets à partir d'une projection de l'image dans un autre espace. Dans [Fränti 00], des images se composant de lignes sont projetées dans l'espace de Hough afin d'en extraire les informations. L'appariement sera alors directement effectué dans cet espace. En raison de la complexité du calcul nécessaire à l'extraction des informations, cette méthode n'est convenable que pour les images non-complexes de petite taille. J.Zhang et al. [Zhang 03] utilisent un *espace de formes* pour représenter l'objet. Dans cet espace, un point de n-dimensions correspond à un objet. Les coordonnées du point sont déterminées à partir de points extrema de l'objet dans l'espace original. Cette représentation assure l'invariance à la rotation, à la translation et au changement d'échelle. Elle est, de plus, relativement insensible au bruit. Néanmoins, elle peut s'avérer complexe à utiliser dans le cas d'objets de k-dimensions, $k > 2$. Introduit par D.Zhang et Lu [Zhang 02b], le descripteur GFD (*Generic Fourier Descriptor*) est prouvé plus performant que celui basé sur les moments de Zernike en termes de robustesse pour les grandes bases de formes et de complexité de calcul. Il s'agit d'une extension des descripteurs de Fourier à la région. En fait, l'image est convertie à partir de coordonnées polaires (voir Fig. 2.14) pour conserver l'invariance à la rotation. Puis, la transformée de Fourier 2D de cette image projetée est calculée. Les coefficients de Fourier dans l'espace polaire sont invariants à la rotation et à la translation. Ainsi, le GFD défini par (2.20) assure l'invariance aux transformations similaires.

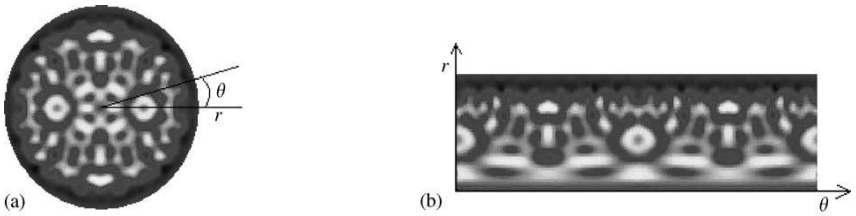


FIG. 2.14 – (a) l'image originale en coordonnées polaires (b) l'image transformée en coordonnées cartésiennes (reprise de [Zhang 02b]).

$$GFD = \{\frac{|PF(0,0)|}{area}, \frac{|PF(0,1)|}{|PF(0,0)|}, ..., \frac{|PF(0,n)|}{|PF(0,0)|}, ..., \frac{|PF(m,0)|}{|PF(0,0)|}, ..., \frac{|PF(m,n)|}{|PF(0,0)}\} \tag{2.20}$$

où *area* est l'aire du cercle minimal englobant l'objet, m et n sont respectivement les fréquences radiales et angulaires maximales et $PF(.,.)$, les coefficients de Fourier dans l'espace polaire :

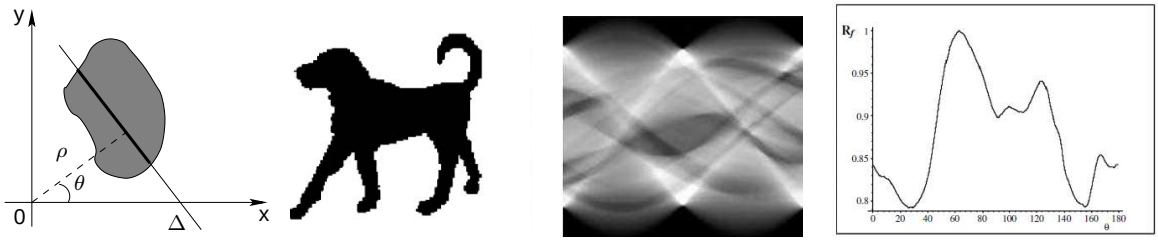
$$PF(\rho,\phi) = \sum_{r=0}^{R}\sum_{i=0}^{T} f(r,\theta_i) exp[j2\pi(\frac{r}{R}\rho + \frac{i}{T}\phi)] \tag{2.21}$$

avec $\theta_i = i(2\pi/T)$, $r = \sqrt{(x-x_c)^2 + (y-y_c)^2}$; R et T, les fréquences radiales et angulaires ; (x_c, y_c) le centroïde de l'objet.

La transformée de Radon est également utilisée pour construire des descripteurs. Elle consiste à projeter l'image sur une droite ($\Delta : \rho = x\cos\theta + y\sin\theta$) pour n'importe quelle valeur de (ρ, θ) avec ρ la distance de Δ à l'origine du repère et θ l'angle entre l'axe des abscisses et la normale à (Δ) (voir Fig. 2.15(a)). La transformée de Radon d'une fonction $f(x,y)$ est définie par l'équation (2.22) :

$$T_{Rf}(\rho,\theta) = \int_x \int_y f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)\ dxdy \tag{2.22}$$

avec $\delta(x)$ la fonction de Dirac, $\theta \in [0,\pi]$ et $\rho \in]-\infty, +\infty[$.



(a) (b) (c) (d)

FIG. 2.15 – (a) Définition de la transformée de Radon. (b) Une forme 2D. (c) Transformée de Radon de la forme (b). (d) $\mathcal{R} - signature$ de (b) (extraite de [Tabbone 05b]).

La rotation de l'objet entraîne un déphasage dans l'espace de Radon qui peut être corrigé simplement. La difficulté de l'utilisation de la transformée de Radon réside surtout dans la translation (qui modifie la transformée de manière non-linéaire) et le changement d'échelle

(qui implique des modifications sur les coordonnées radiales ρ et les amplitudes de la transformée) [Tabbone 05b]. Dans [Li 03], l'invariance à la translation et au changement d'échelle est obtenue grâce à la normalisation des moments centraux de la transformée de Radon tandis que la transformation SVD (*Singular Value Decomposition*) est utilisé pour la rendre invariante à la rotation. Dans le but de définir un descripteur multirésolution pour l'objet, [Ramos 04] propose d'appliquer la transformée de Radon pour des niveaux différents de résolutions construits à partir de la décomposition en ondelette.

Afin de définir un descripteur basé sur la transformée de Radon invariant à la rotation, à la translation et au changement d'échelle, [Tabbone 06b] introduit une adaptation originale de cette transformée (dite $\mathcal{R}-signature$). Il s'agit de convertir la transformée de Radon de 2D en 1D par une intégration des informations radiales selon les coordonnées angulaires (Fig. 2.15(d)) :

$$\mathcal{R}_f(\theta) = \int_{-\infty}^{+\infty} T^2_{Rf}(\rho, \theta)\ d\rho \tag{2.23}$$

La $\mathcal{R}-signature$ est invariante à la translation et au changement d'échelle si elle est normalisée par un facteur d'échelle (l'aire de la forme par exemple). Afin de la rendre invariante à la rotation, on ne garde que les modules des coefficients de Fourier calculés sur la $\mathcal{R}-signature$. La $\mathcal{R}-signature$ fournit une représentation très compacte de la forme qui ne donne pas une bonne discrimination lors de la recherche dans une grande collection de formes. Les performances de ce descripteur peuvent être améliorées en prenant compte, dans un même temps, la $\mathcal{R}-signature$ de la forme à différentes résolutions (obtenues par seuillages d'une carte de distances de Chamfrein). Une forme est alors représentée par :

$$\left(\frac{FR^0(1)}{FR^0(0)}, ..., \frac{FR^0(\pi)}{FR^0(0)}, ..., \frac{FR^l(1)}{FR^l(0)}, ..., \frac{FR^l(\pi)}{FR^l(0)}\right) \tag{2.24}$$

où $FR^k(.)$ sont les modules des coefficients de Fourier de la $\mathcal{R}-signature$ calculée à la résolution k.

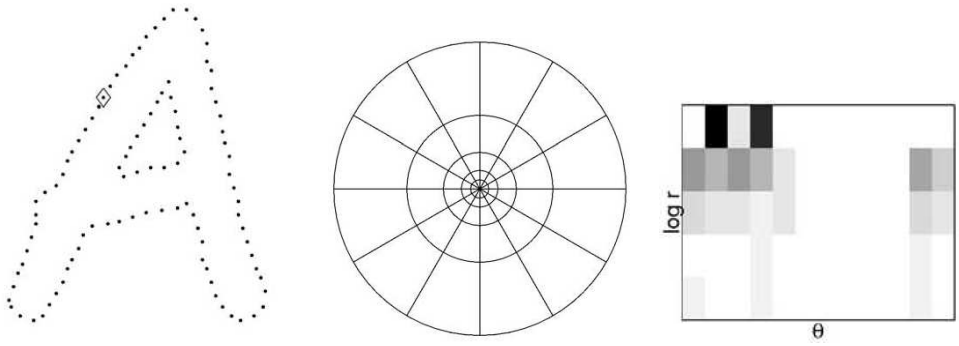
La transformée d'ondelettes est également utilisée dans le travail de [Shen 99] pour déterminer les moments invariants de l'image. Les fonctions d'ondelettes sont choisies comme fonctions de base pour calculer les moments. Les auteurs proposent également une méthode permettant de choisir, à partir d'une base d'apprentissage, des moments discriminants en termes de différenciation entre formes de différentes classes. Les résultats expérimentaux montrent que les moments invariants proposés sont plus performants que les moments de Zernike et que ceux de Li [Li 92] (extension des moments de Hu).

Adam et al. [Adam 01] ont utilisé les invariants issus de la transformée de Fourier-Mellin lors de la construction des descripteurs de formes pour reconnaître des formes multi-orientées et multi-échelles. L'application à la reconnaissance des caractères de l'alphabet et des chiffres extraits des documents techniques a obtenu du succès avec un taux élevé de reconnaissance pour les caractères isolés ainsi que pour les caractères connectés.

2.2.5 Descripteurs basés sur les relations entre pixels formant l'objet

Les descripteurs peuvent aussi être extraits à partir des relations spatiales entre pixels de la forme. Ces relations peuvent être des relations entre un point particulier et les autres points ou des relations entre tous les points du contour [Bernier 03, Belongie 02, Grigorescu 03]. Ces descripteurs sont souvent basés sur les coordonnées polaires de l'image. Dans ce contexte, [Bernier 03] représente un objet par une fonction basée sur les distances et les angles des points de contour par rapport au centroïde de l'objet (le point de référence). Normaliser les distances et les angles par le point ayant la distance maximale permet de rendre ce descripteur invariant au changement d'échelle et à la rotation.

Contrairement à l'approche de [Bernier 03], [Belongie 02] utilise tous les points de contour comme points de référence pour calculer les *Contextes de Forme*[5] correspondant. Le contexte de forme d'un point p_i d'une forme est déterminé par la distribution des points de contour par rapport à p_i en utilisant le système de coordonnées log-polaires ayant p_i comme origine, *point de référence* (voir Fig. 2.16). Une forme ayant N points de contour sera représentée par



(a) (b) (c)

FIG. 2.16 – (a) Points de contour d'une forme. (b) Espace de calcul de *contexte de forme* avec 5 plages pour les log-distances et 12 plages pour les coordonnées angulaires θ. (c) contexte de forme du point $\diamond$ de la forme (a) (reprise de [Belongie 02]).

l'ensemble des N *contextes de forme* des points de contour. Pour que le *contexte de forme* soit invariant à la rotation, le vecteur tangeant au point de référence est utilisé comme axe abscisse lors du calcul du *contexte de forme*. Afin de le rendre invariant au changement d'échelle, les distances entre points de contour sont normalisées par la distance moyenne des paires de ces points. L'appariement bipartite est utilisé pour calculer la mesure de similarité. Cette approche fournit de bons résultats pour la reconnaissance de formes.

De façon similaire, [Grigorescu 03] propose de représenter un objet par un ensemble d'informations extraites aux points de contour, dite *"distance set"*. Le *"distance set"* du point p_i de contour est l'ensemble des distances entre p_i et les K points de contour les plus proches. Cette représentation est invariante à la translation et à la rotation. L'invariance au changement d'échelle est obtenue en normalisant les distances par le diamètre de l'objet. En raison du fait

[5]Traduit de "Shape Contexts"

que ce descripteur est calculé à partir de points locaux, il est robuste à la déformation et aux occlusions [Ghosh 05].

Dans [Yang 05], les informations locales au point de référence sont définies à partir de deux contraintes de distance et d'angle formant par ce point de référence et les deux autres points décrivant l'objet (Fig. 2.17). En discrétisant les informations de ces deux contraintes séparément, chaque objet est représenté par deux matrices caractéristiques, l'une contenant les contraintes de rapport de distances et l'autre les contraintes angulaires. Des évaluations sur des bases de symboles graphiques ont montré la robustesse de ce descripteur face aux rotations, aux translations, aux changements d'échelle, aux déformations et aux bruits. Cependant, ce descripteur ne peut être appliqué qu'aux symboles segmentés. La complexité de calcul est par ailleurs très élevée, $O(n^3)$ avec n le nombre de points du squelette du symbole.

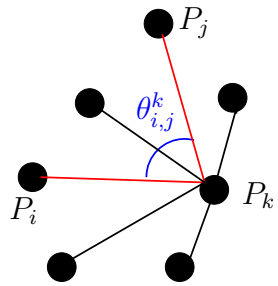


FIG. 2.17 – Exemple de deux contraintes entre deux points P_i et P_j au point de référence P_k : contrainte de rapport de distances $L^k_{i,j} = \min\left(\frac{|P_kP_i|}{|P_kP_j|}, \frac{|P_kP_j|}{|P_kP_i|}\right)$ et contrainte angulaire $\theta^k_{i,j} = \widehat{P_iP_kP_j}$.

2.3 Conclusion

Nous avons proposé dans cette partie un panorama des descripteurs de formes. Ces descripteurs sont généralement divisés en deux classes : les descripteurs structurels et les descripteurs statistiques. Par comparaison aux descripteurs statistiques, les descripteurs structurels sont plus difficiles à mettre en œuvre. En effet, leur utilisation entraîne une complexité algorithmique élevée pour l'indexation et l'appariement entre formes. De plus, ils nécessitent des étapes de pré-traitement telles que la vectorisation ou la poly-lignes pour segmenter les formes en primitives. La qualité de cette segmentation est un facteur clef pour la stabilité du descripteur. Un avantage de ces descripteurs est qu'ils permettent de faire des appariements partiels. Concernant les descripteurs statistiques, ils sont généralement plus simples à mettre en œuvre et plus stables car ils ne dépendent pas fortement d'une étape intermédiaire. Parmi ces descripteurs, le descripteur GFD et ceux basés sur les moments de Zernike sont les plus performants en termes de discrimination de formes. Néanmoins, ils n'assurent pas la tolérance à la déformation ou à l'occlusion partielle. Au contraire, le *contexte de forme* et le *"distance set"* fournissent une bonne représentation de l'objet tout en offrant la possibilité de tenir compte d'informations locales. Cette dernière propriété permet de rendre ces descripteurs tolérants à la déformation et à l'occlusion partielle.

Dans le cadre de ce travail de thèse, qui porte sur la localisation de symboles dans les documents graphiques, nous nous intéressons plus particulièrement aux descripteurs de formes pouvant s'adapter à la représentation de symboles dans les documents. Il nous faut en plus, en vue de répondre à notre problème de localisation, choisir un descripteur qui soit tolérant à la déformation et à l'occlusion partielle du symbole. En effet, le problème de la localisation de symboles nécessite souvent de recourir à une première étape de segmentation, qui ne fournit pas une décomposition parfaite des symboles dans l'image. Un descripteur sensible à la déformation et à l'occlusion ne permet donc pas de retrouver correctement des symboles. Nous proposons donc dans le chapitre 4 un descripteur basé sur le *contexte de forme*. Ce descripteur, qui permet de fournir une représentation des symboles moins complexe que celle obtenue avec le *contexte de forme*, conserve les propriétés d'invariance aux transformations linéaires, de tolérance à la déformation et à l'occlusion. Il permet également de décrire des symboles non-segmentés contenus dans les documents (chapitre 5).

Chapitre 3

État de l'art sur les approches de localisations de symboles dans les documents

Sommaire

Nous abordons dans ce chapitre le problème complexe de la *localisation de symboles*[6] dans les documents où les symboles ne sont pas isolés de leur contexte. Bien qu'il existe beaucoup de travaux visant à la définition de bons descripteurs pour la représentation d'un symbole, ces derniers ne peuvent généralement pas être utilisés directement pour le problème de localisation de symboles dans les documents. En effet, un descripteur peut fournir de bons résultats pour la recherche de symboles segmentés ou isolés mais être difficilement adaptable au cadre de la description des symboles d'un document. Une stratégie classique de description d'un document consiste à le décomposer en plusieurs composantes, puis à appliquer des descripteurs prédéfinis sur chacun de ces composantes. Souvent une étape supplémentaire de vectorisation est appliquée [Wenyin 07, Fonseca 05, Locteau 07, Rusinol 06]. Certains travaux proposent de ne considérer que les symboles satisfaisants certaines conditions (par exemple : la convexité, la connectivité ou la fermeture du symbole) [Tabbone 07, Rusinol 06, Rusinol 07].

Dans ce chapitre, nous proposons un panorama sur des techniques de localisation de symboles dans les documents graphiques. Malgré de nombreux travaux dans l'analyse de documents, il y a peu de travaux qui attaquent le problème de la localisation de symboles. Nous les divisons suivant deux catégories : les approches structurelles et les pixelaires. Les approches se basant sur la représentation structurelle de l'image seront considérées comme des approches structurelles. Elles sont basées souvent sur les graphes et possèdent généralement une étape de segmentation

[6] Traduit de *"symbol spotting"*

des documents en primitives. Le symbole est ensuite détecté via une étape de regroupements de primitives sous certaines conditions. Dans les approches pixelaires, la localisation est effectuée directement sur l'image entière sans étape préalable de segmentation.

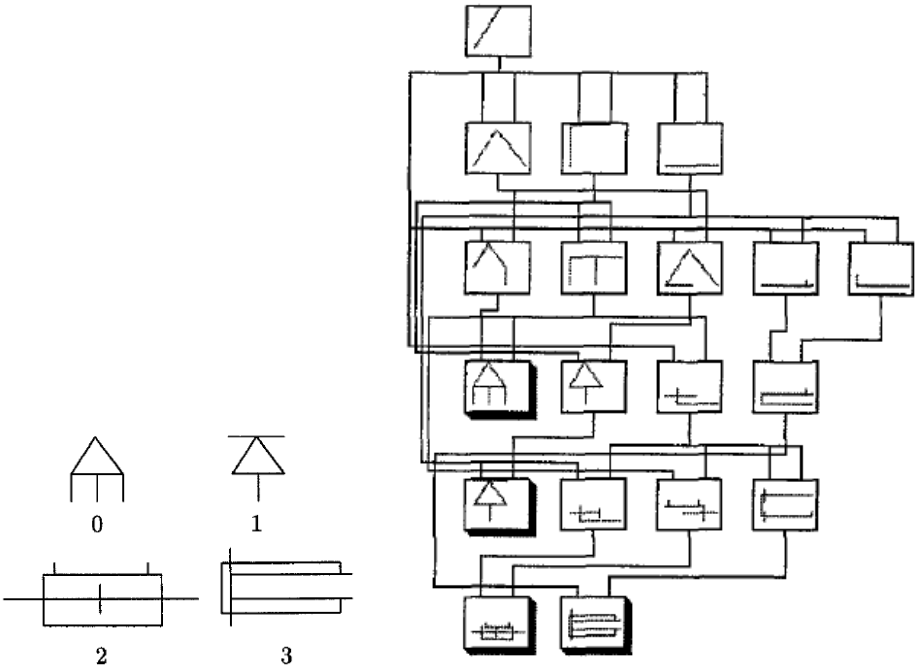
3.1 Approches structurelles

Une étape indispensable pour les approches structurelles consiste à décomposer l'image en primitives comme des lignes, des courbes, des régions ou bien des formes géométriques simples telles que des cercles, des rectangles, etc. Le critère d'adjacence ainsi que les caractéristiques de chaque primitive sont pris en compte pour agréger des parties qui répondent aux hypothèses prédéfinies ou qui correspondent aux (sous-)structures connues par le système. Les relations entre les primitives sont souvent représentées par un graphe et la mise en correspondance des objets revient à la recherche d'un isomorphisme de (sous-)graphes.

Le segment est l'une des primitives les plus simples à utiliser pour la description des symboles [Messmer 95, Park 00, Park 03]. Dans [Messmer 95], les symboles connus par le système sont représentés dans un réseau qui décrit le processus de construction au fur à mesure à partir des segments. Ce réseau se compose de plusieurs niveaux qui correspondent aux parties possibles des symboles. Ces parties s'élargissent de plus en plus en combinant des parties des niveaux précédents. Le niveau le plus bas contient les symboles complets (voir Fig. 3.1). Le module d'appariement à chaque partie permet de trouver des (sous-)isomorphismes des symboles dans le document. La représentation des symboles connus par un réseau rend la recherche des isomorphismes de (sous-)graphes plus efficace au niveau de la complexité des calculs. Ce système fournit aussi la capacité d'apprendre de nouveaux symboles à partir des documents. Cependant, ces nouveaux symboles doivent satisfaire trois heuristiques : i) le graphe du symbole doit être connecté ; ii) il doit contenir au moins un circuit fermé, et si possible, des circuits fermés adjacents ; iii) le segment qui croise le circuit fermé appartient aussi au symbole.

Afin de localiser rapidement les régions de l'image pouvant potentiellement contenir le symbole requête, [Dosch 04] propose d'utiliser la signature vectorielle pour représenter le symbole requête ainsi que les différentes régions de l'image. La signature d'une entité (symbole) est définie par l'ensemble des relations existantes entre segments. Cinq types de relations sont considérés : le recouvrement, la colinéarité, le parallélisme, la jonction "T" et la jonction "V" (Fig. 3.2). Pour effectuer la localisation, l'image est divisée en plusieurs régions disjointes (appelées "bucket") dont la taille dépend de la taille du plus grand symbole de référence. Un test d'inclusion des signatures entre les symboles de référence et ces "bucket" permet d'étiqueter des symboles qui peuvent être présents dans la région. Cette étape de pré-traitement ne permet pas de déterminer la localisation exacte de chacun des symboles mais permet de détecter rapidement le type de symbole potentiellement contenu dans chacune des régions (Fig. 3.3). Ce travail est à considérer comme une étape préalable à la localisation plus précise de symboles.

Parmi les autres approches structurelles, [Zuwala 06b, Tabbone 07] proposent de segmenter les symboles contenus dans les documents pour les localiser par rapport à un symbole requête. Ainsi, le document est décomposé en chaînes de points connectés qui sont ensuite regroupées pour redéfinir de nouveaux symboles grâce à la construction d'un dendrogramme. En effet, les points



(a) (b)

FIG. 3.1 – Réseau de représentation des symboles connus par le système. (a) exemple de quatre symboles. (b) la partie du réseau qui représente les symboles $0, 1, 2, 3$ dans (a).

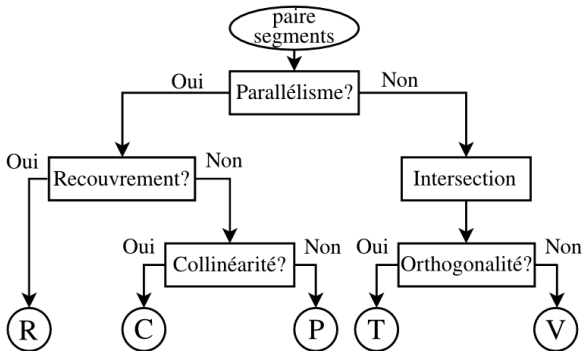


FIG. 3.2 – Détermination des relations existantes entre deux segments pour la construction de la signature vectorielle : le recouvrement (R), la collinéarité (C), le parallélisme (P), la jonction "T" (T) et la jonction "V " (V) (extraite du [Dosch 04]).

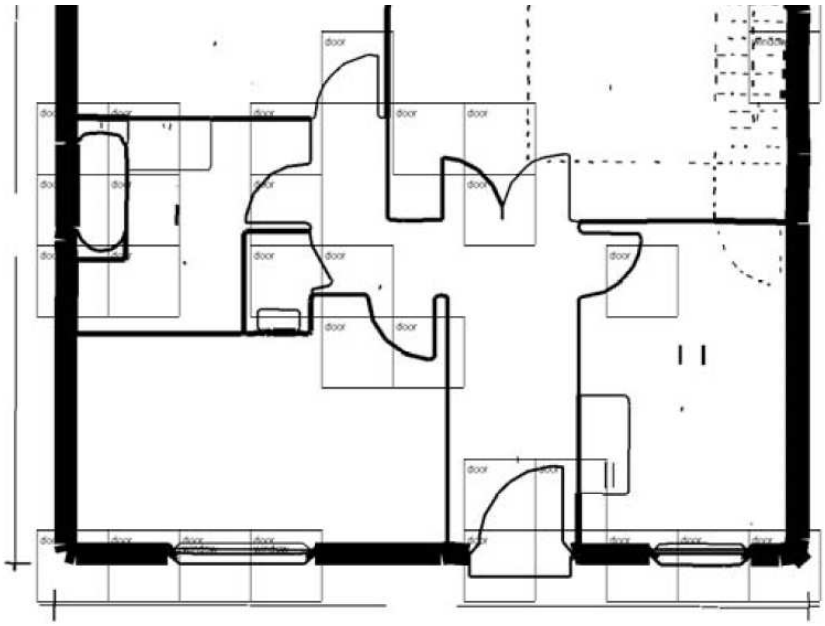
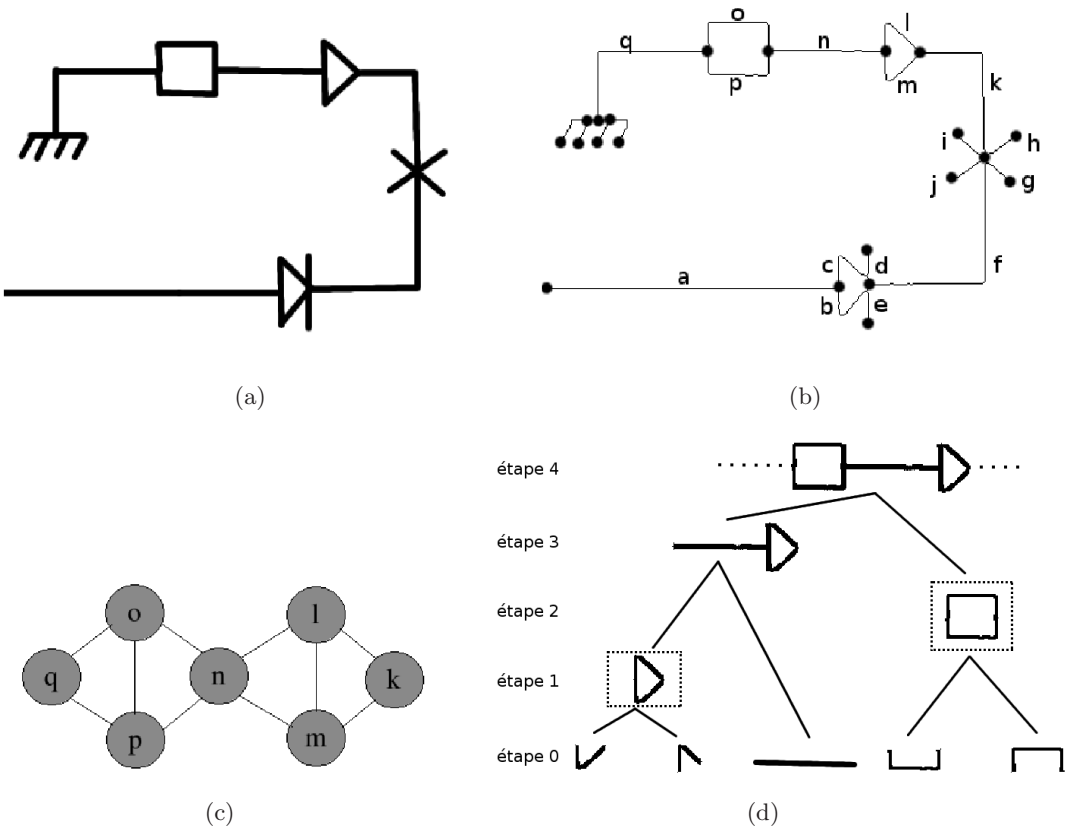


FIG. 3.3 – Extrait du résultat obtenu par la méthode de [Dosch 04].

de jonction et les points terminaux du document sont localisés pour décomposer le document en chaînes de points connectés (Fig. 3.4(b)). Le document est ensuite représenté par un graphe de jonctions dans lequel les nœuds correspondent aux chaînes de points et les arcs au fait que deux chaînes de points soient connectées (Fig. 3.4(c)). L'objectif de la décomposition est d'isoler des sous-ensembles de chaînes de points qui pourraient former un symbole. Ces sous-ensembles peuvent être obtenus par la recherche de toutes les combinaisons possibles de chaînes, néanmoins une telle approche est difficilement applicable pour les grands documents. Afin de faire face à cette difficulté, les auteurs proposent de construire un dendrogramme (3.4(d)). La construction de ce dernier se fait de façon itérative en fusionnant à chaque étape deux chaînes de points. Le choix des fusions est guidé par deux hypothèses : (i) un symbole est constitué d'un ensemble de chaînes de points connectés ; (ii) les chaînes de points d'un symbole ont tendance à être convexes. La localisation d'un symbole dans ces documents revient ainsi à chercher les sous-ensembles de chaînes les plus proches parmi ceux créés par les dendrogrammes. Chaque sous-ensemble est représenté par un descripteur global de formes tel que l'ART. L'évaluation de cette approche sur 100 documents et en prenant compte les 500 plus proches voisins a montré que celle-ci ne permettait pas d'obtenir de bonnes performances pour la localisation de symboles (avec à peu près 57% des symboles manqués). Néanmoins, en intégrant une méthode d'indexation pour la recherche des plus proches voisins et en agrégeant trois autres descripteurs qu'ART, les auteurs ont montré que le taux de symboles non détectés pouvait être fortement diminué (passage de 57% à 9%). Malgré ces améliorations, la méthode proposée ne permet pas de détecter les symboles dont les chaînes ne sont pas connectées, les symboles qui ne possèdent pas de points de jonction dans le document et ne permet pas de faire face au cas où deux symboles partagent une même chaîne de points (Fig. 3.5) [Zuwala 06a].

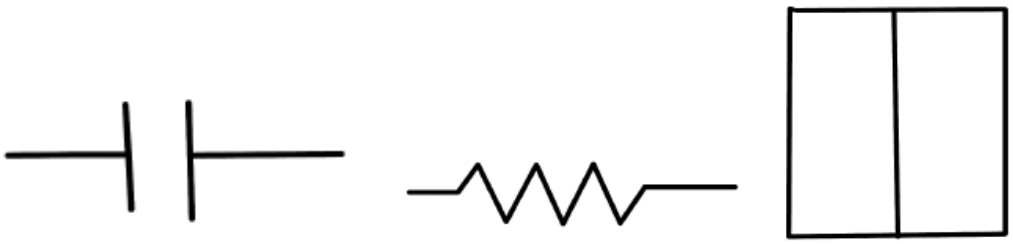
L'approche proposée par [Qureshi 08] suit le même principe : le document graphique est



(a) (b)

(c) (d)

FIG. 3.4 – Segmentation d'un document proposée par [Zuwala 06b]. (a) document. (b) décomposition du document avec points de jonction et points terminaux (points noirs). (c) une partie du graphe de jonction correspondant à la décomposition de (b). (d) un dendrogramme permettant de regrouper les chaînes de points.



(a) (b) (c)

FIG. 3.5 – Symboles non traités par la méthode proposée par [Zuwala 06b]. (a) symbole non connecté. (b) symbole n'ayant pas de points de jonction. (c) deux symboles partageant une même chaîne en admettant qu'un rectangle soit un symbole.

aussi décomposé en segments et représenté par un graphe dont les nœuds sont des *quadrilatères* (issus de l'étape de vectorisation, correspondant aux segments) et chaque arc relie deux quadrilatères adjacents (Fig. 3.6). Les arcs sont étiquetés par le type de connexion existant entre deux *quadrilatères* et pondérés par l'angle relatif et le rapport de longueurs entre les deux segments adjacents. Afin de localiser des régions d'intérêt qui contiennent des symboles, les auteurs proposent de définir un poids pour chaque nœud du graphe. Celui-ci est calculé en fonction de 6 hypothèses de la composition d'un symbole :

- H1 - Un symbole se compose de petits segments par rapport à d'autres segments du document ;
- H2 - Les segments appartenant à un symbole ont des longueurs comparables ;
- H3 - Deux segments adjacents formant un angle relatif dont la valeur est très différente de 90° ont une grande probabilité d'appartenance à un symbole ;
- H4 - Le symbole contient souvent des segments parallèles ;
- H5 - Un segment d'un symbole est rarement connecté à plus de 3 autres segments ;
- H6 - Les plus petites boucles fermées correspondent souvent à des symboles.

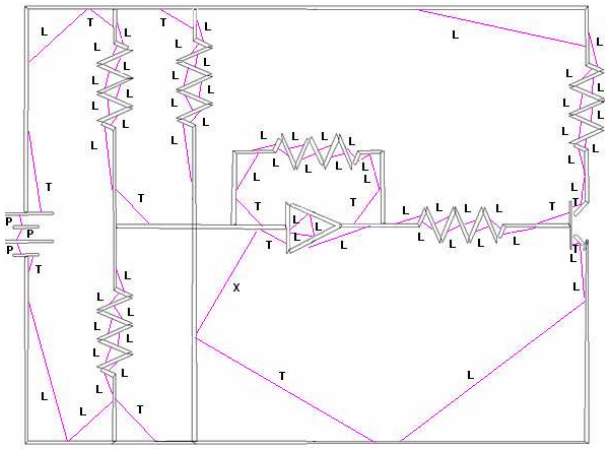
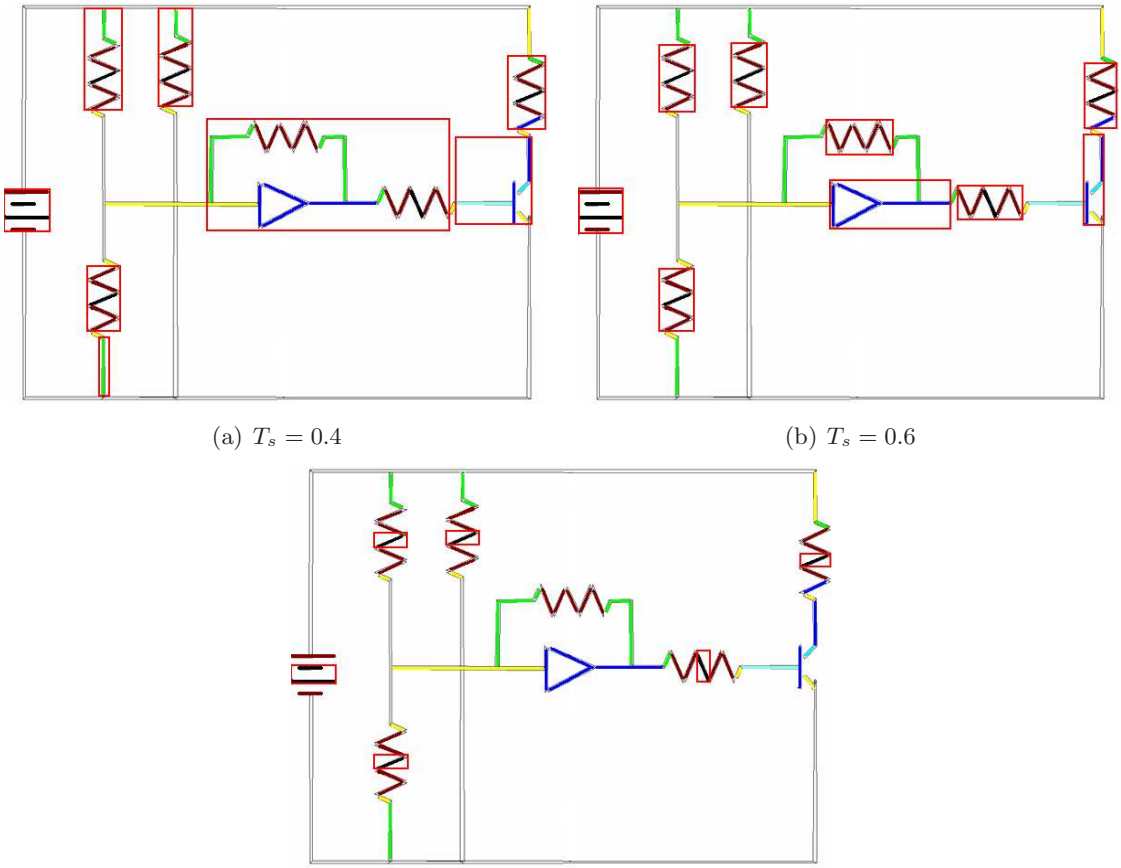


FIG. 3.6 – Graphe de représentation d'un document dont les arcs sont étiquetés par le type de connexion entre deux nœuds : jonctions "L", "S", "T", intersection "X" et parallélisme "P" (tirée de [Qureshi 08]).

Les régions qui intègrent des nœuds connectés possédant des poids élevés (supérieur à un seuil T_s) sont extraites afin de générer un ensemble de régions pouvant potentiellement contenir des symboles (Fig. 3.7). Cet ensemble de régions est utilisé pour la recherche de symboles. En effet, lorsqu'une requête d'un symbole est effectuée, la mise en correspondance entre les (sous-)graphes de ces régions et celui de la requête permet directement de trouver les régions contenant des occurrences de la requête dans le document.

Une remarque importante concernant cette approche est qu'il faut déterminer un seuil optimal T_s afin d'avoir des zones d'intérêt correctes. Une valeur trop grande ou trop petite entraîne une mauvaise segmentation (voir Fig. 3.7) conduisant à des erreurs pour la recherche de symboles.

Dans [Locteau 08], l'image est décrite par un graphe d'adjacence inexacte. Le squelette de l'image est utilisé pour construire un graphe d'adjacence dont les nœuds sont des régions (qui sont



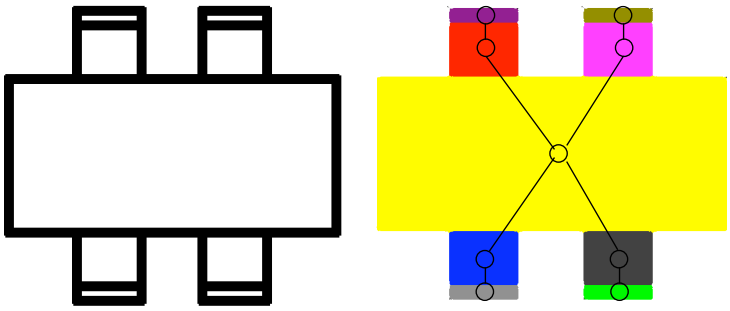
(a) $T_s = 0.4$ (b) $T_s = 0.6$

(c) $T_s = 0.8$

FIG. 3.7 – Régions extraites en fonction de la valeur du seuil T_s (tirée de [Qureshi 08]).

des composantes connexes, appelées *occlusions*) et dont les arcs relient les régions aux frontières limitrophes (Fig. 3.8). Ces régions sont caractérisées par les pseudo-invariants de Zernike. Les arcs quant à eux sont étiquetés par des informations liées aux deux régions qu'ils relient telles que la distance entre les deux centres de gravité, le rapport de surfaces, l'orientation de la droite liant les deux centres de gravité vis-à-vis de l'orientation principale d'une des deux régions. Chaque nœud du graphe peut posséder une étiquette nominale, déterminée grâce à une étape d'apprentissage, pour faciliter l'emploi d'algorithme opérant sur le graphe. À cause des dégradations pouvant exister dans les images, la topologie des graphes construits n'est pas toujours parfaite. Pour faire face à ce problème, des règles supplémentaires définies à partir d'une base d'apprentissage sont utilisées pour corriger les erreurs. Ces dernières permettent de construire un graphe d'adjacence inexacte pour chaque image. La localisation de symboles dans les documents revient à chercher des isomorphismes de sous-graphes. Ce problème, qui est un problème classique en théorie des graphes, est NP-complet. Pour le résoudre, l'auteur propose d'utiliser une approche d'extension d'isomorphisme partiel explorant un espace état-transition.

Bien que cette méthode donne de bonnes performances pour la reconnaissance et la détection de symboles, elle possède néanmoins certaines limites dûes à l'utilisation du graphe et à la squelettisation de l'image. Ainsi, le graphe défini ne permet pas de traiter les symboles qui ont des composantes déconnectées ou qui n'ont pas de composantes connexes (Fig. 3.9). De plus, cette méthode ne permet pas de distinguer certaines classes de symboles ayant des graphes isomorphes. Enfin, la performance de cette méthode repose largement sur la squelettisation, elle est donc sensible à la connexité des objets même sur des images non-bruitées et il peut rester toujours des problèmes de régions sur-segmentées malgré l'application de pré-traitements intégrés (Fig. 3.11).



(a) (b)

FIG. 3.8 – (a) Un symbole traité. (b) les régions (*occlusions*) extraites à partir du squelette du symbole et le graphe d'adjacence correspondant.

Avec la même idée de regroupement des primitives pour construire des symboles, [Locteau 07] propose une approche perceptuelle basée sur la théorie de Gestalt et sur un graphe de visibilité. Il s'agit de redéfinir les régions caractérisant l'objet en regroupant des segments dans un graphe de visibilité pour obtenir des polygones convexes pouvant couvrir le symbole et caractérisant le plus fidèlement possible la structure de celui-ci. La représentation par un graphe de regroupements perceptuels résout les cas non-représentables ou non-différenciables du graphe d'adjacence

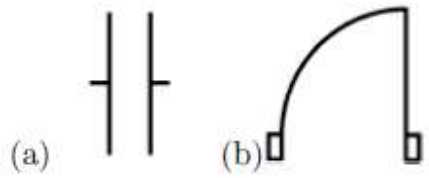


FIG. 3.9 – Exemples de symboles non-traités par la méthode de graphe d'adjacence de [Locteau 08].

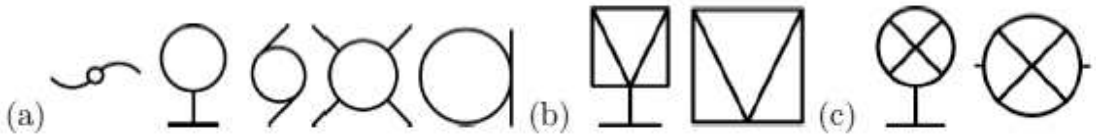


FIG. 3.10 – Groupes de symboles similaires et non différenciables, identifiés par la méthode de graphe d'adjacence de [Locteau 08].

FIG. 3.11 – Problèmes de régions sur-segmentées lors de la squelettisation. (a) Occlusions proches liées par une autre occlusion. (b) et (c) Occlusions fines dont la localisation d'une fragmentation peut être multiple [Locteau 08].

inexacte proposé par le même auteur. La Fig. 3.12 présente quelques exemples de graphes de visibilité et de régions segmentées correspondantes. Cette approche donne de bons résultats pour

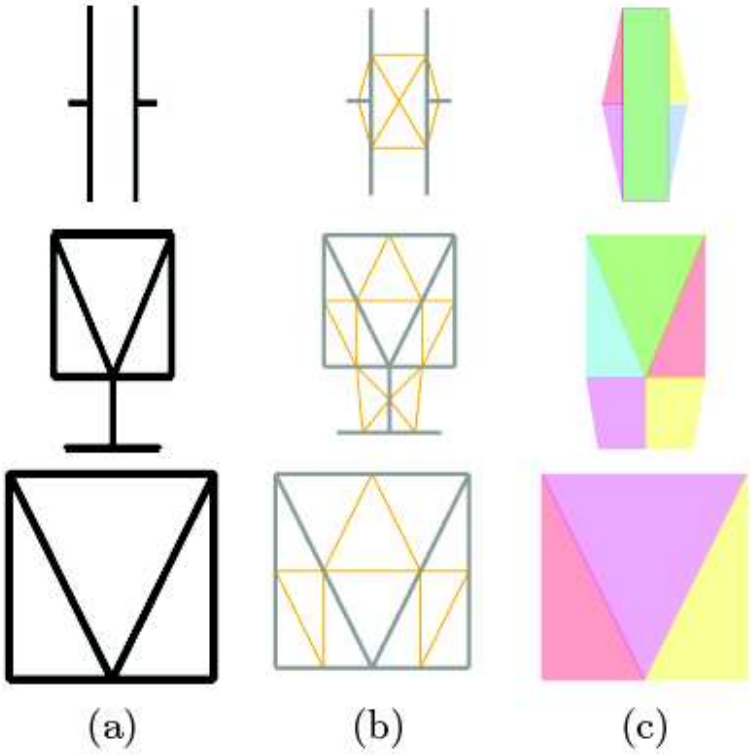


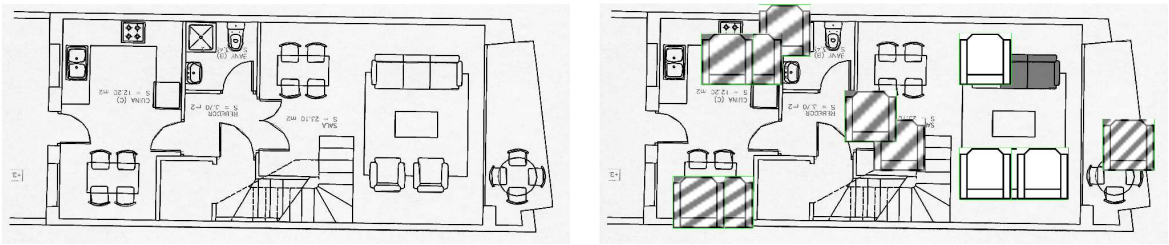
FIG. 3.12 – (a) Symboles non-traités avec le graphe d'adjacence inexacte. (b) Graphes de visibilité. (c) Régions (occlusions) segmentées à partir de graphes de visibilité (reprise de [Locteau 08]).

la reconnaissance de symboles, même pour les symboles n'ayant pas de contours fermés. Cependant, le problème de la localisation de symboles reste une tâche complexe en raison de la grande connexité du graphe. De plus, ce graphe est sensible à la qualité des données vectorielles.

Suivant un même principe, Rusinol et al. [Rusinol 06] déterminent sur un document vectorisé des régions d'intérêt à partir des coordonnées de l'ensemble des segments connectés. Les régions d'intérêt sont définies comme des fenêtres dynamiques déterminées à partir des coordonnées maximales et minimales des segments adjacents. Une signature vectorielle est proposée pour la représentation de chaque région. Dans un premier temps, les régions non-pertinentes (tels que le mur et le câblage) sont facilement éliminées. Un vote est ensuite effectué pour sélectionner parmi les fenêtres détectées celles pouvant contenir le symbole requête. Afin de diminuer l'effet des erreurs de vectorisation, le document est repassé en basse résolution lors de la détection de régions d'intérêt.

Les évaluations sur des documents réels montrent que les occurrences du symbole requête sont bien localisées malgré l'obtention de beaucoup de faux positifs (Fig. 3.13). Ces faux positifs sont causés par l'erreur de détection des régions d'intérêt et par le manque d'informations dans la base de signatures vectorielles. Un autre point négatif de cette approche, qui est tributaire de la qualité des résultats de la vectorisation, est que seuls les symboles connectés ont la possibilité

d'être localisés.



(a) Image originale

(b) Localisations avec une probabilité élevée de trouver le canapé

FIG. 3.13 – Exemple d'un résultat de la méthode proposée par [Rusinol 06].

Afin d'améliorer cette méthode, [Rusinol 07] propose de localiser rapidement des régions qui contiennent probablement un symbole similaire à une requête sans étape préalable de segmentation ou d'apprentissage. Une analyse en composantes connexes est effectuée sur le document entier pour trouver des régions fermées qui sont ensuite représentées par des polygones d'approximation. L'auteur propose d'utiliser l'ovale de Cassini pour construire une fonction de hachage de façon à indexer le document. Un polygone est représenté par un couple (a, b), où a, b sont deux paramètres géométriques de l'ovale de Cassini couvrant le polygone. Cet ovale possède deux foyers qui sont les deux points les plus éloignés du polygone et dont l'intersection passe par le centre de gravité du polygone (Fig. 3.14). Les couples (a, b) (discrétisés) sont utilisés comme entrées d'une table de hachage pour indexer tous les polygones ayant ces mêmes valeurs. Les régions dans le document contenant probablement des occurrences du symbole requête sont détectées par un système de vote qui se base sur l'identification de primitives (polygones) dans le document similaires à celles de la requête grâce à la table de hachage.

L'approche n'a pas besoin d'une étape d'apprentissage ou bien de segmentation préalable mais elle se base sur une analyse en composantes connexes. Un défaut de cette approche est qu'elle favorise les faux positifs (Fig. 3.15). En effet, les primitives dans une entrée de la table de hachage peuvent ne pas être toujours similaires. De plus, cette approche ne prend pas en compte l'organisation structurelle du symbole. D'ailleurs, l'approche ne s'applique qu'aux symboles contenant des composantes connexes dont les contours sont fermés.

Basée sur ce même principe d'utiliser des régions fermées comme primitives pour indexer et localiser des symboles dans le document, [Rusinol 09b] propose d'utiliser une chaîne de segments adjacents du contour de chaque région pour identifier et indexer les régions. La distance entre deux régions, représentées par la chaîne A et B, est mesurée par le coût de la transformation de la chaîne A en la chaîne B. La table de hachage est aussi utilisée pour augmenter la vitesse de recherche des primitives similaires. La chaîne médiane d'une entrée de la table est prise comme index d'accès. Ainsi, un nombre élevé de primitives similaires à celles du symbole requête indique l'existence d'une occurrence du symbole dans le document.

Bien que cette approche offre une faible précision de localisation dans les documents, le rappel est lui élevé et le classement des réponses obtenues est généralement correct. Un avantage de cette méthode est que, grâce à la fonction de distance définie pour la mise en correspondance

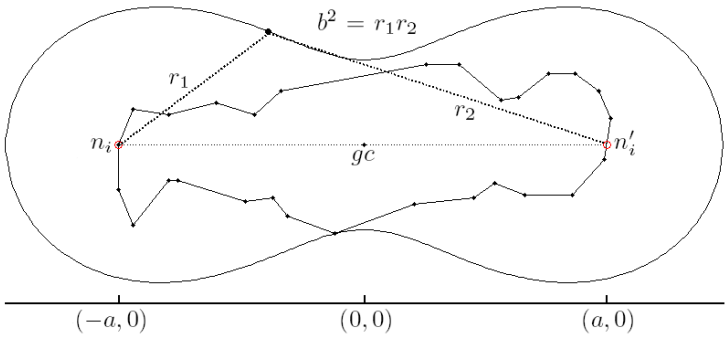
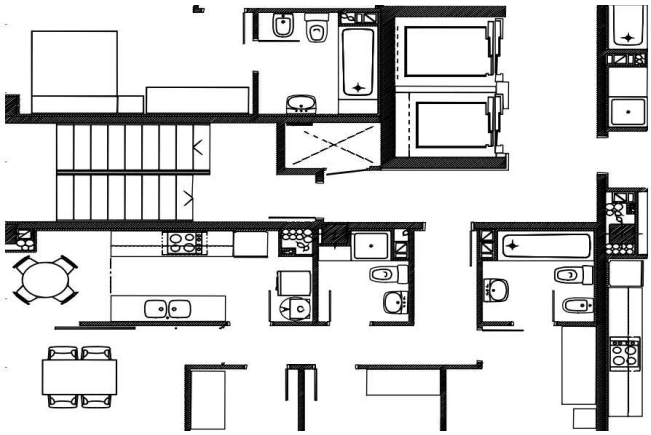
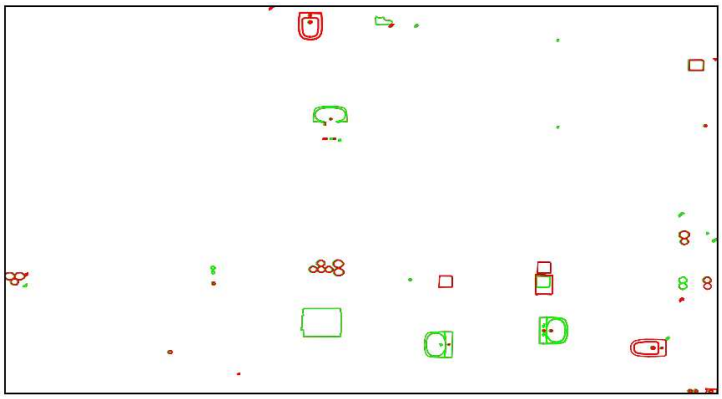


FIG. 3.14 – Polygone d'approximation d'une forme dont le centre de gravité est à gc et n_i, n'_i sont deux points les plus éloignés passant par gc et un ovale minimal de Cassini ($b^2 = r_1 r_2$) couvrant le polygone dont les foyers sont n_i, n'_i ayant normalisés à $(-a, 0)$ et $(a, 0)$.

(a) Modèles à localiser : le lavabo et le bidet



(b) Image originale



(c) Résultats de la localisation : les régions étiquetées comme le lavabo sont marquées en vert et les régions étiquetées comme le bidet sont marquées en rouge

FIG. 3.15 – Exemple des résultats obtenus par la méthode de [Rusinol 07].

entre deux chaînes, elle s'avère tolérante au problème de fragmentation de segments lors de la construction des polygones d'approximations de chaque région. Néanmoins, en raison du fait que seules des régions fermées soient prises comme primitives, la méthode ne permet pas de traiter les cas de symboles dont le contour est ouvert, ni les symboles partiellement invisibles.

Dans [Wenyin 07], la vectorisation est également utilisée pour extraire des primitives (lignes, arcs) d'un symbole ainsi que du document. Ces primitives sont utilisées pour construire un graphe attribué : les primitives représentent les nœuds du graphe et leurs relations (intersections, parallélisme, perpendicularité, ligne-arcs/cercles) les arcs du graphe. Pour chaque symbole requête, un *arbre-squelette* et un *chemin de traverse* sont extraits de son graphe. Ces derniers sont utilisés pour chercher des occurrences du symbole dans le document (Fig. 3.16). En effet, lors de la localisation du symbole dans un document, chaque primitive contenue dans le *chemin de traverse* du symbole est successivement utilisée pour chercher des primitives correspondantes dans le document en vérifiant les contraintes existantes vis-à-vis des primitives précédentes. Ainsi, les hypothétiques occurrences du symbole requête sélectionnées sont celles qui répondent, lors du parcours du *chemin de traverse*, à toutes les contraintes imposées par le symbole. Pour augmenter l'efficacité de la recherche dans le cas de symboles déformés, les auteurs ont proposé d'intégrer un processus de retour de pertinence qui aide à la mise à jour de taux de tolérance utilisés dans l'étape de reconnaissance. Les résultats expérimentaux sont prometteurs. Cependant, la performance du système dépend fortement de l'étape de vectorisation. De plus, cette méthode ne permet pas d'effectuer des recherches sur tous les types de symboles tels que les courbes libres[7] ou les symboles dont le graphe est déconnecté.

Similairement, Liu et al. [Liu 09] ont également utilisé des lignes et des courbes comme des primitives pour caractériser les images. Pour chaque primitive, la structure locale est définie par les voisins les plus proches. Elle est caractérisée par les distances et angles relatifs entre cette primitive et les primitives dans le voisinage. La détection de symboles dans l'image est effectuée par un processus de vérification des hypothèses. Tout d'abord, les k structures locales les plus proches de chaque primitive du modèle sont extraites et les paramètres de transformation correspondant sont estimés. En considérant que chaque estimation est un point dans l'espace de paramètres, une région dense dans cette espace signifie qu'il existe une occurrence du modèle dans l'image. De telles régions sont détectées et passées par une étape de vérification pour localiser les occurrences du modèle dans l'image. Les résultats obtenus sur quelques images de test montrent que la méthode est efficace (Fig. 3.17). Cependant, comme les autres méthodes structurelles, la vectorisation a un impact important sur la performance de la méthode. Les résultats obtenus sont intéressants à condition que l'image d'entrée est partfaite.

Pour conclure sur les approches structurelles, nous pouvons constater que ces approches permettent de localiser pertinemment des symboles même dans des cas complexes (cf. Fig. 3.17). Cependant, elles nécessitent généralement des étapes intermédiaires, telles que la vectorisation ou la segmentation, pour décomposer les images en primitives. La performance de ces approches dépend ainsi fortement des résultats des étapes intermédiaires. Une erreur dans ces étapes peut entraîner des altérations dans la représentation des documents (par exemple, le fait de se retrouver avec trop de nœuds et de relations dans les graphes) et ainsi nuire aux résultats de la

[7] Traduit de "free-curves"

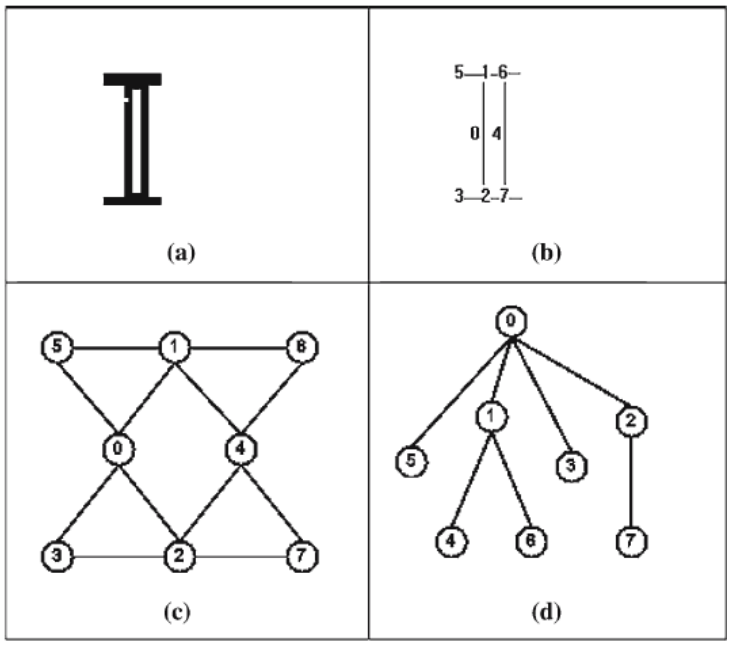


FIG. 3.16 – Représentation du symbole. (a) symbole. (b) primitives du symbole obtenus après une étape de vectorisation. (c) graphe attribué. (d) *arbre-squelette* du graphe en désignant primitive 0 comme la racine de l'arbre, son *chemin de traverse* est donc $E(0,5), E(0,1), E(0,3), E(0,2), E(1,4), E(1,6), E(2,7)$ où $E(i,j)$ désigne l'arc liant deux nœuds i et j (extrait de [Wenyin 07]).

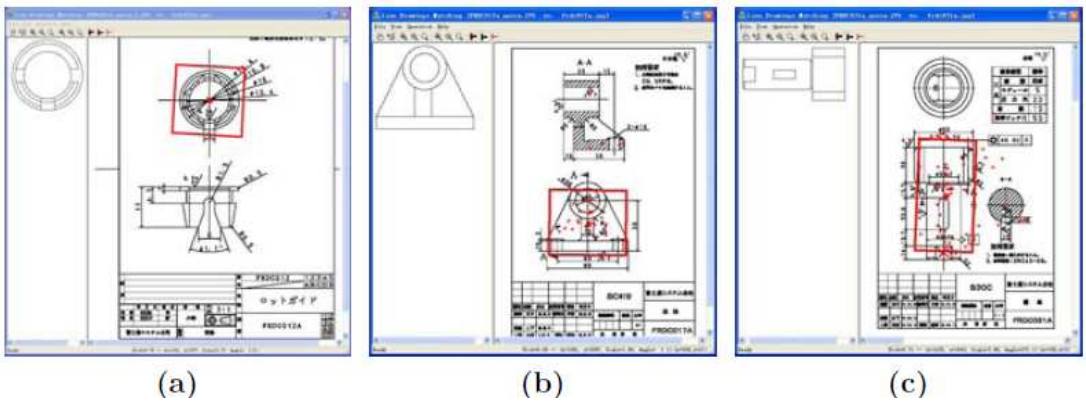
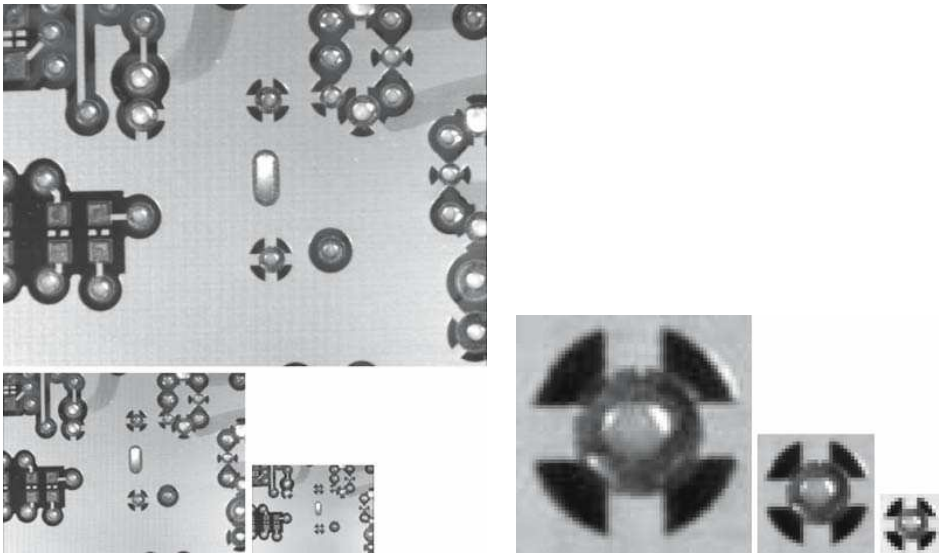


FIG. 3.17 – Illustration de résultats obtenus par la méthode proposé par [Liu 09].

localisation. De plus, elles nécessitent souvent de poser des contraintes et des hypothèses sur les symboles considérés telles que la connectivité, la convexité ou la fermeture du symbole. Enfin, elles nécessitent de faire face à la complexité de la mise en correspondance entre (sous-)graphes qui est un problème NP-complet. En effet, dans ces approches le symbole et les documents sont souvent représenté par des graphes. Nous présentons dans la suite des approches ne nécessitant pas de traitements préalables, et qui imposent peu ou pas du tout d'hypothèses sur les symboles.

3.2 Approches pixelaires

MacLean et Tsotsos proposent une méthode de localisation et de reconnaissance rapide de symboles dans le document qui ne nécessite aucune étape de pré-traitement ou de décomposition [MacLean 08]. Cette méthode est basée sur la représentation pyramidale de l'image. Ainsi, une représentation pyramidale correspondant au modèle requête et une autre correspondant à l'image sont construites (Fig. 3.18). La première étape de la recherche des positions potentielles des occurrences du symbole requête consiste à chercher des maxima sur la surface de corrélation normalisée et calculée à partir des sommets des deux pyramides (Fig. 3.19). Ces positions sont ensuite propagées, niveau par niveau, vers la base de la pyramide. Les positions exactes des occurrences sont trouvées au niveau le plus bas. Afin de déterminer le nombre maximal de niveaux de la pyramide correspondant à chaque modèle, les auteurs proposent d'utiliser une analyse du pire des cas. La méthode proposée localise rapidement et précisément les positions du symbole requête mais elle est sensible au changement de densité du fond de l'image et ne permet pas de vérifier les propriétés d'invariances à la rotation et au changement d'échelle.



(a) (b)

Fig. 3.18 – Représentation pyramidale (a) de l'image et (b) du modèle (reprise de [MacLean 08]).

Partant de la même idée de ne pas segmenter l'image, [Rusinol 08] propose, pour localiser des symboles ou des mots dans des documents techniques, d'effectuer directement un système de

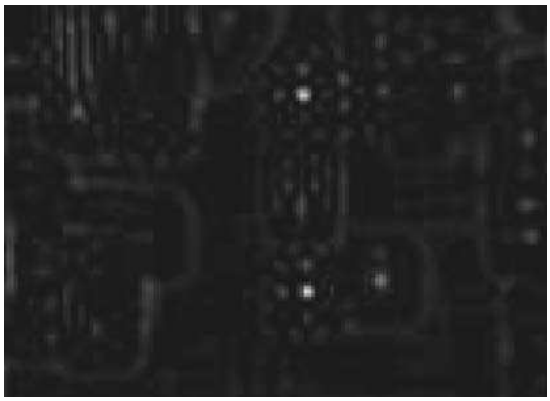
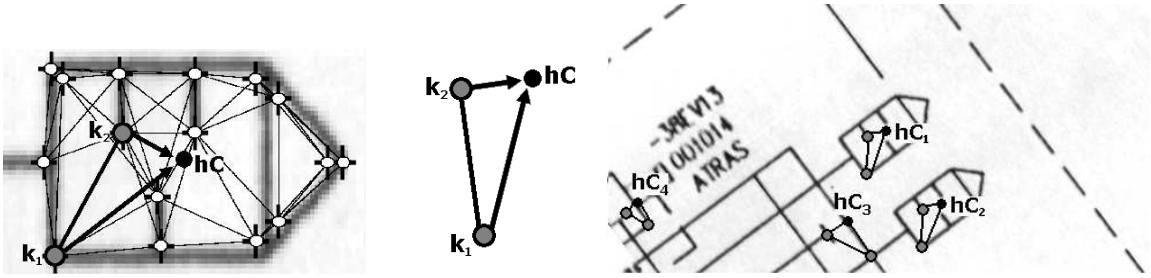


FIG. 3.19 – Surface de corrélation au niveau le plus haut de l'image, les deux extrema les plus forts (blancs) correspondent aux positions de deux occurrences du modèle (reprise de [MacLean 08]).

vote sur l'image entière en validant l'organisation spatiale des points d'intérêt appariés. Chaque paire de points d'intérêt du document est mise en correspondance avec une paire de points du modèle. Ces mises en correspondance sont utilisées pour définir, au travers d'un système de vote, le centre hypothétique du symbole dans le document. Ce centre hypothétique est déterminé de sorte que la relation spatiale existante entre ce centre et deux points considérés dans le document soit similaire à celle existante entre le centre et deux points d'intérêt appariés du modèle requête (Fig. 3.20). Afin d'éviter une explosion de traitements inutiles des paires de points, pour chaque point d'intérêt, seuls les cinq voisins les plus proches sont considérés pour former un couple (voir Fig. 3.20(a)).



(a) (b) (c)

FIG. 3.20 – (a) Graphe de proximité entre les points d'intérêt du modèle en considérant les 5 plus proches voisins. (b) Relation spatiale existante entre 2 points d'intérêt et le centre du modèle. (c) Quelques exemples de sous-configurations mises en correspondance avec celle de (b) dont les centres hypothétiques sont $hC1, hC2, hC3, hC4$ (extraite de [Rusinol 08]).

Bien que la méthode n'ait été testée que sur une petite base de documents et fournit des résultats non parfaits, la combinaison de descriptions locales et d'informations géométriques fournit une bonne discrimination pour la localisation et la reconnaissance de symboles.

Pour représenter le symbole dans les documents, [Escalera 09] propose le descripteur CBSM (*Circular Blurred Shape Model*). Celui-ci se base également sur la distribution des points de contour autour des centroïdes locaux. Une étape d'apprentissage est effectuée pour définir les

paramètres optimaux du descripteur. Ce descripteur est ensuite utilisé pour vérifier si des régions locales du document contiennent des occurrences du symbole requête. Ces régions ne sont pas segmentées a priori mais dynamiquement par une fenêtre glissante qui parcourt le document (avec un pas de déplacement de 5 pixels). La taille de la fenêtre varie selon un intervalle prédéfini. Cette technique classique de localisation est simple et permet d'utiliser n'importe quel descripteur de formes pour la reconnaissance de symbole. Quelsques résultats sont montrés en Fig. 3.21. Néanmoins, un point crucial de cette approche concerne le choix des valeurs pour les paramètres liés à la taille de la fenêtre et au pas de déplacement. Le symbole peut ne pas être trouvé si sa taille est éloignée des tailles choisies pour la fenêtre. De plus, le nombre de régions de test peut devenir très important pour une base de documents, en particulier dans le contexte d'images de grandes tailles.

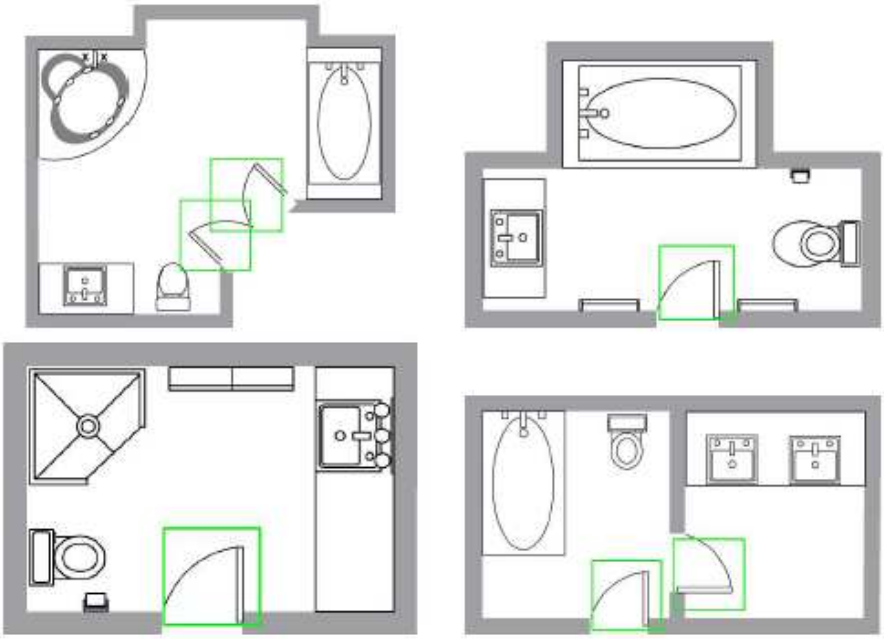


FIG. 3.21 – Quelques exemples de localisation de portes dans les documents de la méthode proposée par [Escalera 09].

Bien que peu de travaux se soient encore intéressés à cette catégorie d'approches, les approches pixelaires présentent néanmoins de nombreux avantages par rapport aux approches structurelles. En effet, les approches pixelaires n'ont pas besoin d'une étape préalable de vectorisation ou de segmentation. De plus, elles n'imposent pas de fortes contraintes sur les symboles considérés. Enfin, ces approches permettent d'éviter d'avoir à faire face au problème de l'appariement entre (sous-)graphes. Ces avantages nous orientent à exploiter cette catégorie d'approche pour notre problème de localisation de symboles dans les documents graphiques (chapitre 5).

3.3 Mesures d'évaluation

Considérées comme des systèmes de recherche d'information, les méthodes de localisation (de textes ou de symboles) sont souvent évaluées par des mesures classiques de ce domaine tels que la précision, le rappel et le *F-score* [Marcus 92, Rath 03, Tabbone 07, Valveny 07, Locteau 08]. Ces valeurs sont calculées par analyse des réponses obtenues pour une requête. Une réponse est soit pertinente soit non-pertinente. Soit une base de données contenant un ensemble d'éléments X et une requête i consistant à chercher les occurrences d'un symbole dans X, nous notons X_p l'ensemble des éléments pertinents et X_n l'ensemble des éléments non-pertinents par rapport à i, $X_p, X_n \subset X$. Soit *Ret* l'ensemble des éléments obtenus pour la requête i, les formules (3.1) et (3.2) définissent la précision et le rappel du résultat de la recherche. La précision P mesure la capacité du système à fournir des réponses pertinentes. Elle est définie par le rapport entre le nombre d'éléments pertinents trouvés et le nombre total d'éléments récupérés :

$$P = \frac{|Ret \cap X_p|}{|Y|} \tag{3.1}$$

Le rappel est défini par le rapport entre le nombre d'éléments pertinents trouvés et nombre total d'éléments pertinents dans la base. Il mesure la capacité du système à trouver exhaustivement les éléments pertinents existants.

$$R = \frac{|Ret \cap X_p|}{|X_p|} \tag{3.2}$$

La plupart des systèmes de recherche d'informations ne fournissent pas simplement un ensemble d'éléments a priori pertinents, mais fournissent aussi un classement de ces éléments. Afin de prendre en compte ce classement, une approche consiste à ne considérer la précision que pour les n éléments (P_n) définis comme les plus pertinents par le système. Pour fournir une vue d'ensemble sur la précision et le rappel, la courbe *précision/rappel* $P(r)$ est souvent utilisée. $P(r)$ désigne la précision du résultat lorsque le rappel atteint r. En pratique, la courbe est obtenue en calculant P_n pour plusieurs valeurs de $n(n = 1, 2, 3, ...)$ et en calculant le rappel associé.

Une autre mesure pour donner une vue d'ensemble sur les performances d'un système consiste à combiner la précision et le rappel en une valeur unique, tels que le *F-score* [Shaw 97] (Eq. (3.3)), la *E-mesure* [van Rijbergen 79] (Eq. (3.4)), ou la *précision moyenne*.

$$F = \frac{2}{\frac{1}{R} + \frac{1}{P}} \tag{3.3}$$

$$E = 1 - \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} \tag{3.4}$$

où b est un paramètre qui permet de définir l'importance relative de la précision par rapport au rappel. Si $b = 1$, la *E-mesure* est le complément du *F-score*. Une valeur élevée du *F-score* démontre que le système permet de fournir un bon compromis entre la précision et le rappel.

Il n'existait pas encore dans la littérature de mesures d'évaluation plus spécifiques au domaine de la localisation de symboles. Ainsi Rusiñol et Lladós [Rusinol 09a] ont proposé un ensemble

de mesures pour évaluer les performance des systèmes de localisation de symboles en termes de capacité de reconnaissance, de précision de localisation et d'adaptation à de grandes bases de données. Ces mesures sont des adaptations de mesures classiques du domaine de la recherche d'informations. Elles se basent sur l'aire de recouvrement entre la région d'intérêt du résultat et celle de la vérité terrain. La région d'intérêt d'un symbole est déterminée par son enveloppe convexe.

Étant donnée une collection de documents graphiques, P*tot* désigne l'ensemble des polygones contenus dans l'ensemble des documents dans cette collection, P*rel* l'ensemble des polygones de la vérité terrain où une occurrence du symbole S se trouve. Lors de la localisation du symbole S dans la collection, P*ret* désigne l'ensemble des polygones récupérés. La précision et le rappel sont donc définis par Eq. (3.5) et (3.6).

$$P_A = \frac{A(\mathrm{P}ret \bigoplus \mathrm{P}rel)}{A(\mathrm{P}ret)} \tag{3.5}$$

$$R_A = \frac{A(\mathrm{P}ret \bigoplus \mathrm{P}rel)}{A(\mathrm{P}rel)} \tag{3.6}$$

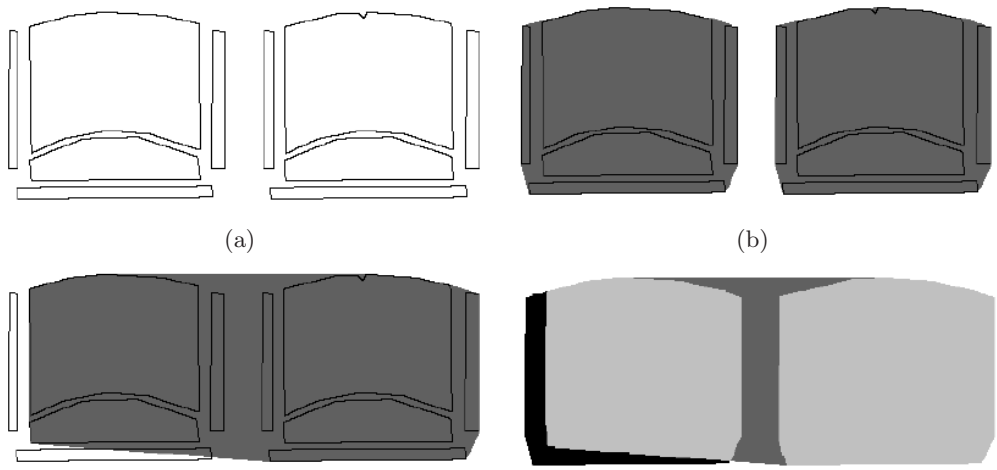
avec $A(\mathrm{P}_i)$ la somme des aires de tous les polygones de l'ensemble P_i. L'opération $\mathrm{P}_i \bigoplus \mathrm{P}_j$ fournit l'ensemble des polygones obtenus par l'intersection spatiale entre les polygones de l'ensemble P_i et ceux de l'ensemble P_j (Fig. 3.22). D'autres mesures d'évaluation sont également reformulées à partir de P_A et de R_A. Par exemple, le *F-score* est redéfini par (3.7).

$$F_A = \frac{2}{\frac{1}{R_A} + \frac{1}{P_A}} \tag{3.7}$$

Ces mesures permettent d'évaluer les performances d'un système de localisation de symboles. Cependant, elles tendent à sur-évaluer la précision des systèmes dans le cas où un résultat fourni par ces derniers couvrent plusieurs zones de vérité terrain (comme dans l'exemple présenté Fig. 3.22(d)). En effet, l'objectif de la localisation de symboles est de fournir des régions où le symbole existe. Ainsi, si une réponse couvre plus d'une région de vérité terrain, une seule région parmi ces régions doit être considérée comme pertinente à la fois. Or, dans les cas comme celui montré en Fig. 3.22(c) où un résultat couvre plusieurs régions en même temps, la précision calculée par la formule (3.5) est très proche de 1 lors que le symbole n'a pas été détecté correctement. Ceci soulève le problème de la pertinence de l'adéquation d'une requête aux résultats proposés par système. La communauté (cf. campagne d'évaluation ÉPEIRES[8]) se pose des questions dans ce sens sur la définition d'une vérité terrain. Dans quelle mesure peut-on dire qu'une réponse à une requête est bonne ?

Concernant la base de vérité terrain, les seules bases disponibles pour le problème de la localisation de symboles sont celles qui ont été définies au cours du projet SESYD[9] [Delalandre 09]. Malheureusement, ces bases de données ne peuvent être facilement utilisées pour calculer les mesures de performance proposées par [Rusinol 09a]. En effet, les mesures de performance proposées par [Rusinol 09a] doivent être calculées au niveau du pixel mais cette précision est peu

[8] Évaluation de PErformances à appliquées à la REconnaissance de Symboles, site : http ://epeires.loria.fr
[9] Systems Evaluation SYnthetic Documents



(a) (b)

(c) (d)

$A(P_i)$	Précision(%)	Rappel (%)
$A(\mathrm{P}rel) = 60560$		
$A(\mathrm{P}ret) = 62307$		
$A(\mathrm{P}ret \oplus \mathrm{P}rel) = 55449$	88.99	91.56
$A(\overline{\mathrm{P}ret} \oplus \mathrm{P}rel) = 5111$		
$A(\mathrm{P}ret \oplus \overline{\mathrm{P}rel}) = 6858$		

(e)

FIG. 3.22 – Re-définition de la précision et du rappel pour le problème de localisation. (a) document original. (b) vérité terrain (Prel). (c) résultat obtenu Pret. (d) polygones de chevauchement du résultat et de la vérité terrain : P$ret \bigoplus$ Prel sont équivalent à deux zones gris clair. (e) Calcul de la précision et du rappel de la localisation (extraite de [Rusinol 09a]).

compatible avec les bases du projet SYSED où les régions pertinentes pour un symbole sont définies à partir de son rectangle englobant.

Ainsi, le manque de bases de vérité terrain et de mesures d'évaluation communes expliquent que les méthodes de localisation n'aient pu être évaluées que sur des bases réduites d'images.

3.4 Conclusion

Pour les images de documents graphiques, il semble que les approches structurelles sont plus adaptées car les lignes et les formes de base sont les primitives principales pour construire le contenu de l'image en respectant des règles de combinaison, de connexion, etc. Intuitivement, ces approches permettent d'effectuer des localisations précises et correctes si les primitives sont correctement extraites. Néanmoins, l'extraction des primitives de l'image exigent souvent des étapes préalables tels que : la vectorisation de l'image, l'approximation de polygones, la décomposition en segments, qui sont sensibles aux bruits. Les erreurs dans ces étapes peuvent causer des erreurs importantes dans les étapes postérieures. Donc des traitements supplémentaires sont nécessaires. De plus, utiliser des graphes pour représenter les documents convertit le problème de localisation en problème d'appariement de (sous-)graphes[10], qui est connu comme un problème NP-complet. En outre, les méthodes se basant sur les graphes nécessitent des hypothèses préalables sur les symboles pour les construire en regroupant des primitives. Ces hypothèses limitent les types de symboles traités.

A l'opposé, les approches pixelaires ne sont pas confrontées au problème NP-complet d'appariement, ni au problème de vectorisation ou de segmentation en primitives. Généralement, elles n'ont pas de contraintes particulières sur les symboles traités. Cependant, les positions des symboles localisés par les approches pixelaires sont moins précises qu'avec les approches structurelles. Mais ce n'est pas une condition indispensable pour le problème de la localisation. Les avantages des approches pixelaires nous conduise vers une méthode pixelaire de localisation de symboles dans les documents graphiques au sein de cette thèse (chapitre 5).

[10]Traduit de "graph matching"

Chapitre 4

Contexte de forme pour les points d'intérêt

Sommaire

Les descripteurs structuraux semblent bien adaptés aux symboles graphiques car un symbole peut être considéré comme une composition de primitives de base telles que les segments, les courbes. Cependant, l'utilisation de ces descripteurs doit faire face aux problèmes de sensibilité aux bruits de la vectorisation et de la recherche des isomorphismes de (sous-)graphes qui est un problème NP-complet bien qu'il existe déjà de nombreux travaux visant à réduire cette complexité [Conte 04]. Les descripteurs se basant sur les pixels peuvent permettre d'éviter ces problèmes. Bien que certains d'entre eux soient très performants (comme *GFD* et les moments de Zernike) en termes de discrimination des formes, ils ne sont pas adaptés à la représentation des symboles dans les documents. En effet, il est très difficile de rendre ces descripteurs tolérants à la déformation ou à l'occlusion partielle. Or cette tolérance est l'un des critères les plus importants pour le problème de localisation car les symboles contenus dans un document sont souvent connectés à d'autres symboles du document ou à d'autres éléments externes. La segmentation d'un document peut donc fournir des symboles incomplets ou contenant des segments qui n'appartiennent pas au symbole.

La représentation du contenu des documents est une des étapes importantes pour la localisation de symboles dans les documents. Dans ce cadre, nous proposons dans ce chapitre un

descripteur basé sur le contexte de forme mais défini aux points d'intérêt. Il existe dans la littérature de nombreuses méthodes de représentation de formes. Nous avons proposé un état de l'art de ces méthodes dans le chapitre 2. Nous avons vu que le choix d'un descripteur dépend fortement de l'application considérée ainsi que des contraintes qui lui sont liées : robustesse aux bruits, stabilité par rapport aux distorsions, invariance aux transformations géométriques, tolérance aux occlusions, etc.

Les symboles graphiques contenant essentiellement des lignes et des courbes, nous posons l'hypothèse que les descripteurs basés sur la caractérisation des relations existantes entre pixels permettent de fournir une bonne description des symboles graphiques en termes de discrimination des symboles isolés et de représentation des informations locales. Dans ce chapitre, nous proposons une adaptation d'un descripteur pixels pour représenter des symboles graphiques. Notre objectif est de le rendre le plus pertinent possible dans le contexte de la localisation de symboles dans des documents. Nous commençons par rappeler la définition du contexte de forme (section 4.1). Ensuite, nous montrons comment ce descripteur peut être adapté uniquement aux points d'intérêt (section 4.2). Dans la partie relative à l'évaluation (4.3), nous montrons les résultats expérimentaux de notre descripteur en le comparant avec d'autres descripteurs. Nous verrons dans le chapitre suivant que ce descripteur s'adapte bien à notre méthode de localisation de symboles dans les documents graphiques.

4.1 Rappel sur le *contexte de forme*

Le *contexte de forme*[11] d'un point de contour p_i d'une forme est déterminé par la distribution des points de contour dans la région de voisinage de p_i [Belongie 02]. Ce descripteur prend la forme d'un histogramme de fréquence d'apparition des points de contour dans des sous-régions entourant le point p_i, le *point de référence*. Ces sous-régions, appelées *bins*, sont déterminées par division de l'espace autour de p_i en plusieurs plages uniformes dans l'espace log-polaire. Cet espace utilise p_i comme origine pour déterminer les coordonnées relatives d'autres points de contour et pour calculer leurs distributions dans les sous-régions.

Soit $\mathcal{C} = \{p_1, p_2, .., p_n\}, p_i \in \mathbb{R}^2$, l'ensemble des points échantillonnés des contours externes et internes d'une forme et n le nombre de points de contour, les coordonnées relatives d'un point q par rapport au point p_i sont données par (4.1).

$$q = (r_{qp_i}, \theta_{qp_i}), \forall q \neq p_i \wedge q \in \mathcal{C} \tag{4.1}$$

où r_{qp_i} est la distance entre q et p_i, θ_{qp_i} est l'angle entre le vecteur $\overrightarrow{p_i q}$ et l'axe horizontal. Le *contexte de forme* h_i du point p_i est donc défini par :

$$h_i(l) = \#\{q \neq p_i : (r_{qp_i}, \theta_{qp_i}) \in bin(l)\}, l = \overline{1, L} \tag{4.2}$$

où $h_i(l)$ est le nombre de points de contour appartenant à la $l^{ème}$ classe de l'histogramme et $bin(l) = \{(r_{*p_i}, \theta_{*p_i}) : r_{*p_i} \in [r_l, r_l + \Delta_{r_l}] \wedge \theta_{*p_i} \in [\theta_l, \theta_l + \Delta_{\theta_l}]\}$, $\Delta_{r_l}, \Delta_{\theta_l}$ désignent la taille du

[11] traduction du "shape context"

$bin(l)$. Un objet $\mathcal{O}$ est ainsi décrit à partir de l'ensemble des *contextes de forme* des points de contour :

$$\mathcal{O} \equiv \{h_i | p_i \in \mathcal{C}\} \tag{4.3}$$

Le *contexte de forme* décrit ci-dessus n'est pas invariant à la rotation et aux changements d'échelles. Pour obtenir l'invariance aux changements d'échelles, les distances radiales sont normalisées par la distance moyenne α des n^2 paires de points de la forme [Belongie 02]. Les auteurs ont également proposé d'utiliser le vecteur tangent associé à chaque point à la place de l'axe absolu horizontal pour rendre le *contexte de forme* invariant à la rotation.

La construction des *bins* dans l'espace log-polaire permet premièrement de représenter de façon relative (distance, angle) les points de contour formant l'objet. Deuxièmement, elle donne une plus grande importance aux points qui sont plus proches par rapport à ceux qui sont plus éloignés (grâce à la transformation *log*). Troisièmement, elle permet de diminuer la sensibilité au bruit grâce à la discrétisation de l'espace en *bins*.

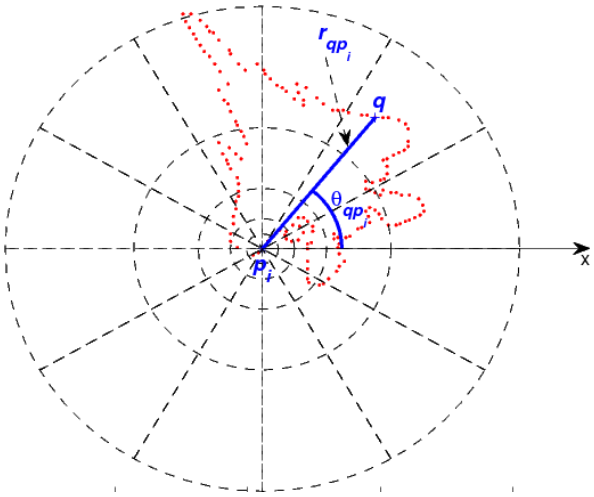


FIG. 4.1 – Distribution des points de contour dans l'espace log-polaire.

Comme le *contexte de forme* est un descripteur qui contient des informations sur la configuration locale associée à chaque point de contour, il semble bien adapté aux symboles graphiques auxquels nous nous intéressons. Cependant, construire une représentation d'un symbole à partir de l'ensemble des *contextes de forme* de tous les points de contour peut engendrer de la redondance d'information ainsi qu'une grande complexité de calcul lors de la représentation du symbole et de l'appariement entre symboles.

4.2 *Contexte de forme* pour les points d'intérêt

Nous présentons dans cette partie un descripteur adapté à la représentation de symboles graphiques. Notre descripteur est une adaptation du *"contexte de forme"* proposé par Belongie et al. [Belongie 02]. Il permet de donner une grande importance aux informations locales par rapport aux informations globales. Il permet également d'assurer l'invariance à la rotation, à

la translation, et au changement d'échelle. Notre motivation est aussi de réduire la complexité computationnelle (du calcul, de la représentation et de l'appariement) et la redondance des informations définies par le *contexte de forme* en attachant le contexte uniquement aux points d'intérêt. Nous précisons dans les sous-sections suivantes le mode d'obtention des points d'intérêt ainsi que notre adaptation du contexte de forme.

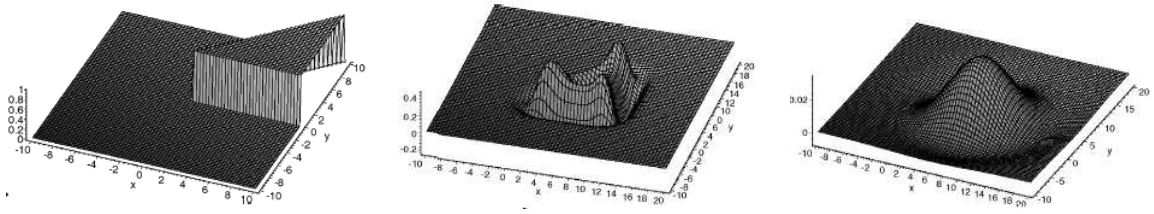
4.2.1 Détection des points d'intérêt

L'identification de points d'intérêt a été au coeur de nombreux travaux de recherche depuis le début des années 80 [Crowley 81, Crowley 84]. De nombreuses méthodes pour détecter ces points ont été proposées [Schmid 00, Lowe 04, Mikolajczyk 04]. La procédure d'extraction des points d'intérêt doit être robuste au changement d'échelle et/ou aux transformations affines. Le principe général de la plupart de ces méthodes consiste à effectuer la recherche de ces points soit à partir de différentes résolutions de l'image, soit à partir d'une résolution raisonnable sélectionnée selon un ensemble de critères [Kadir 01]. La première approche (utilisation de différentes résolutions) est généralement préférée. Parmi ces détecteurs, les détecteurs *SIFT* (*Scale-Invariant Feature Transform*) [Lowe 04], *Harris-Laplace* et *Harris-Affine* [Mikolajczyk 04] font partie des plus populaires et des plus performants au regard de la répétition des points d'intérêt[12] et de l'erreur de localisation lors de changement d'échelle. Les deux premiers sont seulement invariants au changement d'échelle, le dernier est invariant en plus aux transformations plus complexes telles que le changement de point de vue. Dans le cadre de notre application (recherche de symboles dans des documents graphiques), nous ne nous intéressons pas au changement de point de vue. Ainsi, il est plus intéressant pour nous d'utiliser un détecteur invariant au changement d'échelle comme SIFT et *Harris-Laplace* plutôt qu'un détecteur invariant aux transformations affines comme *Harris-Affine*. Le détecteur *Harris-Laplace* est légèrement plus performant que le détecteur SIFT [Mikolajczyk 04]. Néanmoins, le détecteur SIFT est plus avantageux en termes de complexité computationnelle car il se base sur un DoG (*Difference of Gaussian*), une approximation proche de LoG (*Laplacian of Gaussian*) [Mikolajczyk 04].

Nous avons sélectionné dans le cadre de cette thèse le détecteur SIFT même si d'autres méthodes peuvent être des choix valables. Pour celui-ci, un point d'intérêt correspond à un extrema existant dans une pyramide d'échelles, construite par la convolution d'une image I avec des filtres de DoG de tailles différentes. D'après l'évaluation de Mikolajczyk et al. [Mikolajczyk 05], le descripteur SIFT [Lowe 04] calculé aux points d'intérêt détectés par le détecteur SIFT, donne de très bons résultats pour la recherche d'images par le contenu. De plus, les comparaisons expérimentales détaillées dans [Mikolajczyk 02] montrent que les maxima et minima de la fonction LoG fournissent une représentation plus stable que celles obtenues avec d'autres fonctions telles que le gradient, Harris ou Hessian. Dans [Tabbone 05a], l'analyse du comportement de LoG sur les jonctions des modèles a montré que le LoG fournit un ou plusieurs extrema à proximité des jonctions qui jouent un rôle important pour distinguer un modèle d'un autre, surtout pour les symboles graphiques. Les positions de ces extrema, plus ou moins proche des jonctions, dépendent de la résolution à laquelle le LoG est calculé (valeur de σ de LoG, voir Fig. 4.2). En tant

[12]"repeatability score" : le taux moyen de points correspondants détectés dans les images sous des transformations différentes.

qu'approximation du LoG, le DoG possède un comportement similaire. Nous proposons d'étendre ce raisonnement aux symboles puis aux documents graphiques (chapitre 5), en utilisant le détecteur SIFT. Ce détecteur fournit des points d'intérêt aux différentes résolutions. L'utilisation de ce détecteur permet d'éviter le choix arbitraire d'une résolution à traiter et d'avoir des extrema pour caractériser les éléments du document à plusieurs résolutions. Les résolutions les plus grandes sont caractérisées par les points les plus éloignés des jonctions.



(a) (b) (c)

FIG. 4.2 – Le comportement du LoG sur les jonctions du modèle. (a) modèle. (b) LoG calculé du modèle avec $\sigma = 1$. (c) (b) LoG du modèle avec $\sigma = 5$ (extraite de [Tabbone 05a]).

Nous rappelons ici une brève description sur le détecteur SIFT. Les points d'intérêt d'une image obtenus par ce descripteur sont des extrema à chaque niveau de la pyramide d'échelles de l'image. La Fig. 4.3(a) montre le processus de construction de cette pyramide où chaque niveau contient s images DoG. A chaque niveau de la pyramide, l'image initiale est lissée par des filtres gaussiens qui différent par un facteur d'échelle constant k pour avoir un ensemble d'images lissées. La différence entre deux images adjacentes crée une image DoG (Eq. (4.4)). Pour avoir s images DoG à chaque niveau de la pyramide, on fixe $k = 2^{1/s}$. L'image initiale du niveau suivant est obtenue par un rééchantillonnage d'un facteur 2 de la dernière image gaussienne du niveau précédent. Le processus est répété pour avoir une pyramide complète.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (4.4)$$

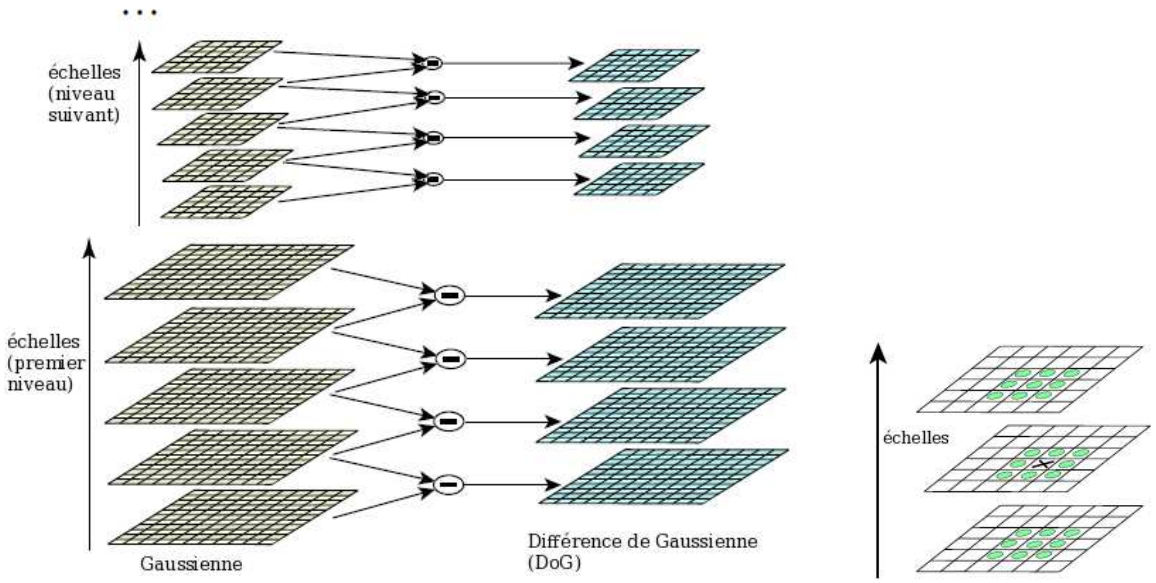
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\pi\sigma^2} \quad (4.5)$$

Afin de détecter les maxima et minima de $D(x, y, \sigma)$, chaque point est comparé avec ses 8 voisins de l'image actuelle et 18 voisins de deux images adjacentes (voir Fig.4.3(b)). Ce point est sélectionné s'il est plus grand ou plus petit que tous ses points de voisinage. Les points détectés sont ensuite filtrés pour les rendre plus stables en utilisant le déterminant et la trace de la matrice du Hessian calculée à la position et l'échelle du point considéré. L'orientation d'un point d'intérêt est déterminée par le gradient local dominant dans la région autour du point.

Chaque point d'intérêt p_i est donc localisé par un quadruplet $(x_i, y_i, \delta_i, \theta_i)$ où (x_i, y_i) sont les coordonnées de p_i, δ_i la résolution où il est détecté (correspond à σ dans la formule (4.4)) et θ_i son orientation.

4.2.2 CFPI : Contexte de Forme pour les Points d'intérêt

Soit $\mathcal{IP} = \{p_1, p_2, .., p_N\}$ l'ensemble des points d'intérêt et $\mathcal{C} = \{q_1, q_2, .., q_n\}$ l'ensemble des points de contour de l'objet. Chaque point dans $\mathcal{IP}$ est considéré comme un point de



(a) (b)

FIG. 4.3 – (a) Construction de la pyramide d'échelles. (b) Détection des extrema des images de DoG : les maxima et les 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 (extraite de [Lowe 04]).

référence pour calculer le *contexte de forme* correspondant. Pour que l'objet soit bien représenté, le descripteur doit être invariant à la rotation et aux changements d'échelle au regard de notre problématique. Les coordonnées relatives des points de contour doivent ainsi être normalisées. De par la restriction du calcul du contexte de forme aux seuls points d'intérêt, il devient nécessaire d'adapter l'étape de normalisation liée à l'orientation car les points d'intérêt ne correspondent pas aux points de contour [Tabbone 05a], c'est-à-dire $\mathcal{IP} \nsubseteq \mathcal{C}$. L'utilisation du vecteur tangent [Belongie 02] n'est donc plus applicable ici. Nous proposons d'utiliser l'orientation (θ_i) du point d'intérêt comme étant l'axe des abscisses lors du calcul des coordonnées relatives. Chaque point d'intérêt p_i est localisé par ses coordonnées, la résolution où il est détecté (δ_i) et son orientation $(\vec{e_i})$ qui forme avec l'axe horizontal un angle θ_i :

$$p_i = \{x_i, y_i, \delta_i, \theta_i\} \tag{4.6}$$

Les coordonnées log-polaires relatives d'un point de contour $q_j \in \mathcal{C}$ dans l'équation (4.1) sont réécrites comme suit :

$$q_j^{p_i} = (r_{ij}, \theta_{ij}) \tag{4.7}$$

où p_i est le point de référence, r_{ij} la distance normalisée de q_j à p_i et $\theta_{ij} =< \overrightarrow{p_i q_j}, \vec{e_i} >$ (voir Fig. 4.4).

Le *contexte de forme* associé au point p_i est défini suivant l'équation (4.2). Il prend la forme d'un histogramme de L classes (L *bins*). Si par exemple, comme dans [Belongie 02], 5

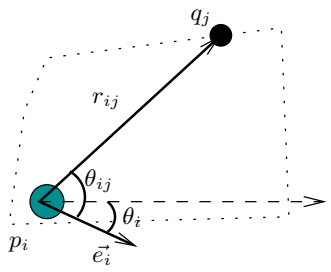


FIG. 4.4 – Coordonnées relatives de q_j par rapport à p_i.

intervalles sont choisis pour la décomposition radiale $log(r)$ avec r compris entre 0.125α et 2α, et 12 intervalles pour la décomposition angulaire θ, alors L vaut 60. Un objet $\mathcal{O}$ est décrit par l'ensemble des *contextes de forme* des points d'intérêt p_i.

$$\mathcal{O} \equiv \{h_i | p_i \in \mathcal{IP}\} \tag{4.8}$$

Afin d'illustrer le comportement du CFPI à la rotation et au changement d'échelle, nous montrons dans la Fig. 4.5 les CFPIs associés à un point d'intérêt avant et après avoir effectué des opérations de rotation et de zoom. Nous pouvons remarquer que les CFPIs en P_1 et en P_2 sont similaires. Dans la section suivante, nous évaluons quantitativement les performances obtenues par le biais de notre proposition.

4.3 Évaluation du descripteur

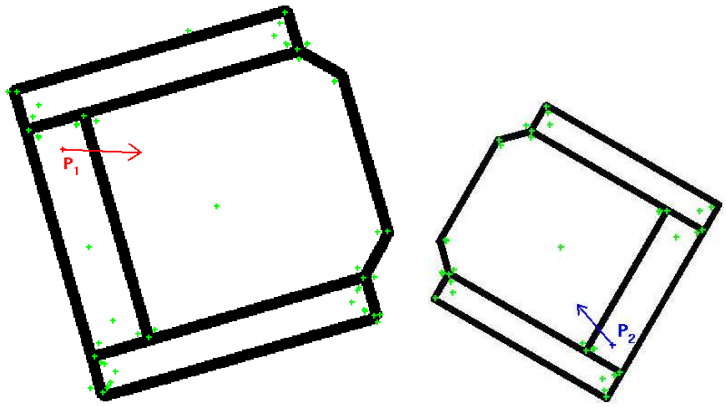
Afin de valider l'adéquation du descripteur CFPI aux symboles graphiques et aussi la performance du descripteur en termes de discrimination des symboles, de tolérance à la déformation et à l'occlusion, nous avons effectué des expérimentations sur différentes bases de symboles (graphiques) isolés. Pour ce faire, nous avons retenu les deux critères suivants : capacité du descripteur à discriminer les symboles et robustesse aux déformations ainsi qu'aux occlusions.

4.3.1 Mesure de similarité

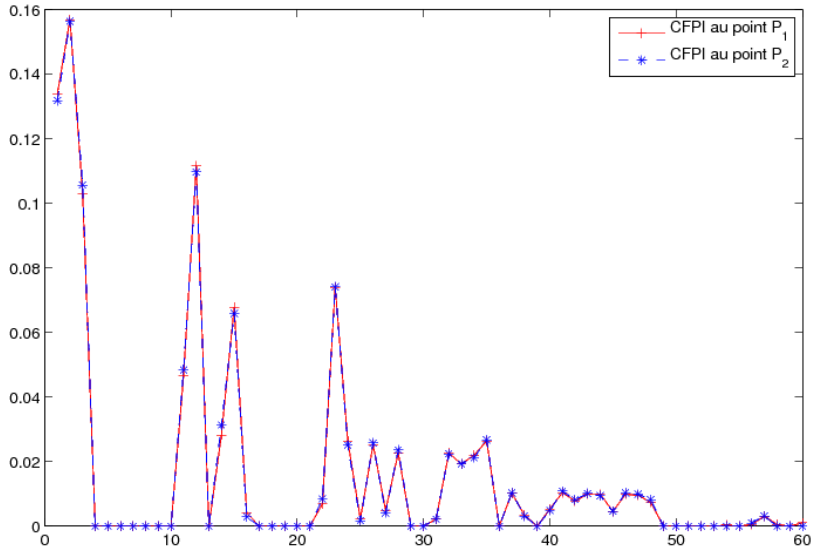
Dans cette évaluation, nous utilisons la distance cosinus pour mesurer la similarité entre deux vecteurs. Pour les symboles représentés par un vecteur caractéristique, la similarité entre deux symboles est déterminée par la similarité entre les deux vecteurs.

Concernant notre descripteur CFPI, un symbole est représenté par l'ensemble des vecteurs CFPI aux points d'intérêt. Nous déterminons donc la similarité entre deux symboles en nous basant sur les similarités des vecteurs CFPI. Ainsi, pour comparer deux symboles, nous commençons par apparier chaque point d'intérêt du premier symbole à un point d'intérêt du second symbole, puis nous déduisons des résultats de ces appariements la similarité entre les deux symboles.

Soit $\mathbb{P} \equiv \{p_1, p_2, .., p_N\}$ et $\mathbb{Q} \equiv \{q_1, q_2, ..., q_M\}$ deux ensembles de points d'intérêt correspondant respectivement aux symboles $\mathcal{P}$ et $\mathcal{Q}$. Nous notons c_{p_i} le vecteur CFPI calculé au point



(a) (b)



(c)

FIG. 4.5 – Un point d'intérêt du symbole avant ((a) - noté P_1) et après avoir subit des opération de rotation et de zoom ((b) - noté P_2), les flèches indiquant l'orientation à chaque point. (c) CFPIs associés au point P_1 et au point P_2.

p_i. Le symbole $\mathcal{P}$ (respectivement $\mathcal{Q}$) est donc représenté par $\{c_{p_i}, i = \overline{1,N}\}$ (respectivement $\{c_{q_j}, j = \overline{1,M}\}$). Deux points p_l et q_k sont appariés si :

$$q_k = arg\min_{q_j \in \mathbb{Q}} d(c_{p_l}, c_{q_j}) \tag{4.9}$$

$$p_l = arg\min_{p_i \in \mathbb{P}} d(c_{p_i}, c_{q_k}) \tag{4.10}$$

où $d(.,.)$ désigne la distance entre deux vecteurs. Dans la Fig. 4.6, nous montrons un exemple des paires de points qui sont appariées. Supposons que $\mathbb{M}$ est l'ensemble des paires de points appariés entre deux symboles $\mathcal{P}$ et $\mathcal{Q}$, $\mathbb{M} \equiv \{(p_1^{\mathcal{P}}, p_1^{\mathcal{Q}}), (p_2^{\mathcal{P}}, p_2^{\mathcal{Q}}), ..., (p_K^{\mathcal{P}}, p_K^{\mathcal{Q}})\}$, $K \leq \min(N, M)$. Nous définissons la distance entre les deux symboles par l'équation (4.11). Cette distance est symétrique, *i.e.* $D_{\mathcal{P},\mathcal{Q}} = D_{\mathcal{Q},\mathcal{P}}$. Elle prend en compte la distance moyenne des appariements (premier terme) ainsi que le taux de points appariés (second terme). Plus cette valeur est petite, plus les deux symboles sont similaires.

$$D_{\mathcal{P},\mathcal{Q}} = \frac{\sum_{i=1}^{K} d(c_{p_i^{\mathcal{P}}}, c_{p_i^{\mathcal{Q}}})}{K} * \frac{1}{2K/(N+M)} \tag{4.11}$$

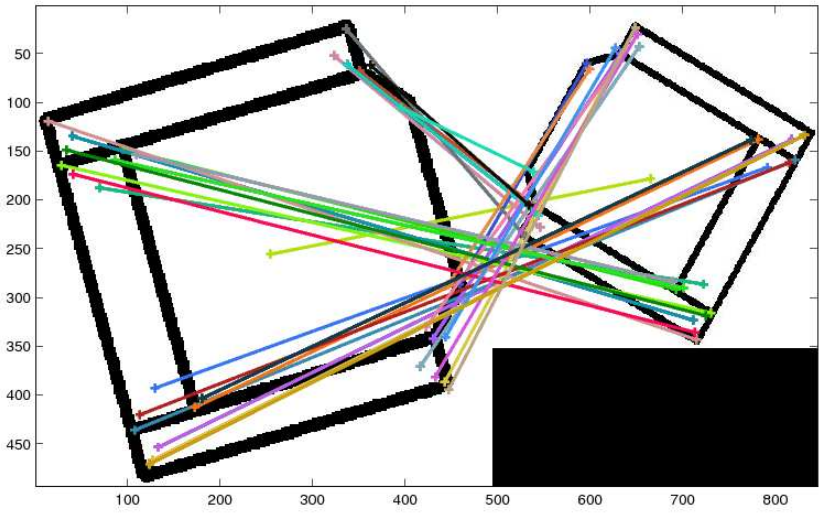


FIG. 4.6 – Appariement entre les CFPIs aux points d'intérêt. Les lignes relient les paires de points appariés.

4.3.2 Bases de symboles isolés

Dans le cadre de cette expérimentation, nous nous proposons d'utiliser essentiellement comme bases de test des bases issues de la compétition de reconnaissance de symboles qui s'est déroulée durant le worshop GREC'03[13]. Ces bases contiennent des occurrences de 50 modèles de symboles (l'ensemble **A**, Fig. 4.7), obtenues par des transformations linéaires (changement d'échelle et rotation) et des déformations et dégradations de chaque modèle de **A**. Nous avons utilisé les

[13] http ://www.cvc.uab.es/grec2003/SymRecContest/index.htm

éléments de A comme requête pour chercher leur occurrences (symboles similaires) dans les bases suivantes :

- *Base **SR1** (scale-set3)* : Cette base contient 250 symboles de tailles différentes de 50 modèles de **A** (voir Fig.4.8(a)). Le facteur d'échelle varie entre 0.5 et 1.5. Le nombre d'occurrences de chaque modèle varie entre 0 et 13.
- *Base **SR2** (rotation-set3)* : contient 250 symboles obtenus par des rotations avec un angle $\alpha \in (0, \pi]$ sur chaque modèle de **A** (voir Fig.4.8(b)). Le nombre d'occurrences de chaque modèle varie entre 0 et 11.
- *Base **SR3** (rotation-scale-set3)* : contient 250 symboles obtenus à la fois par rotation ($\alpha \in (0, \pi]$) et par changement d'échelle (par un facteur compris dans l'intervalle $[0.5, 1.5]$) des modèles de **A** (voir Fig.4.8(c)). Le nombre d'occurrences de chaque modèle varie entre 1 et 10.
- *Base **DD1** (degrad-level3-m3)* : contient 250 symboles dégradés des modèles de **A** (voir Fig.4.8(d)). Cette base contient 5 occurrences de chaque modèle.
- *Base **DD2** (distortion3-set2)* : contient 75 symboles construits par forte déformation (niveau 3) de 15 modèles de **A** (voir Fig. 4.8(e)). Le nombre d'occurrences de chaque modèle varie entre 1 et 11.
- *Base **DD3** (deform-degrad-leve3-m3)* : contient 75 symboles construits par forte déformation (niveau 3) et ajout de bruits de 15 modèles de **A** (voir Fig. 4.8(f)). Le nombre d'occurrences de chaque modèle varie entre 0 et 11.

Afin de vérifier l'adaptabilité des descripteurs pour la recherche de symboles incomplets avec/sans déformation, nous avons utilisé deux autres bases :

- *Base **OC1** (occlusion)* se compose de 77 symboles incomplets créés manuellement à partir des symboles complets de l'ensemble **A** et de ceux obtenus après rotation et changement d'échelle (Fig. 4.8(g)).
- *Base **OC2** (shapes99)* se compose de 9 catégories avec 11 occurrences de chaque catégorie. Chaque occurrence subit des variations de forme ainsi que d'occlusions (Fig. 4.8(h)). Cette base a été définie par Sharvit et al. [Sebastian 01][14].

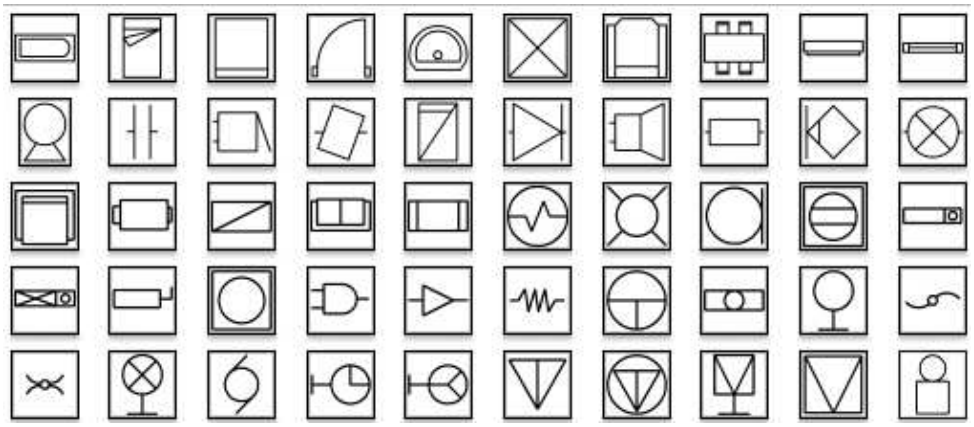
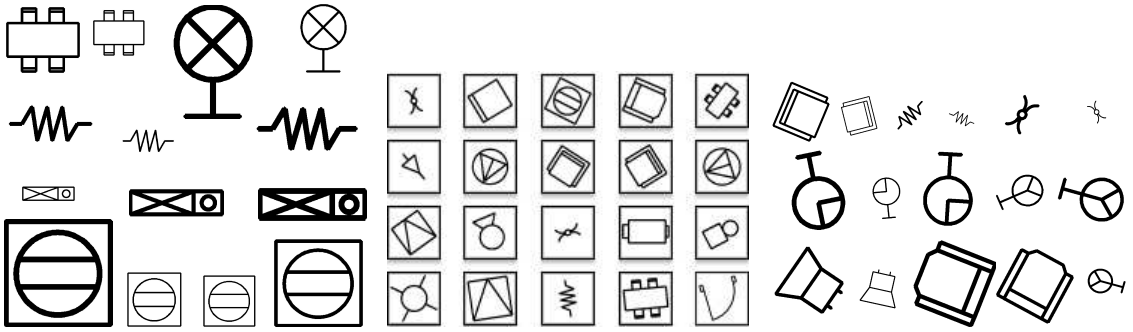


FIG. 4.7 – Modèles dans l'ensemble ***A***.

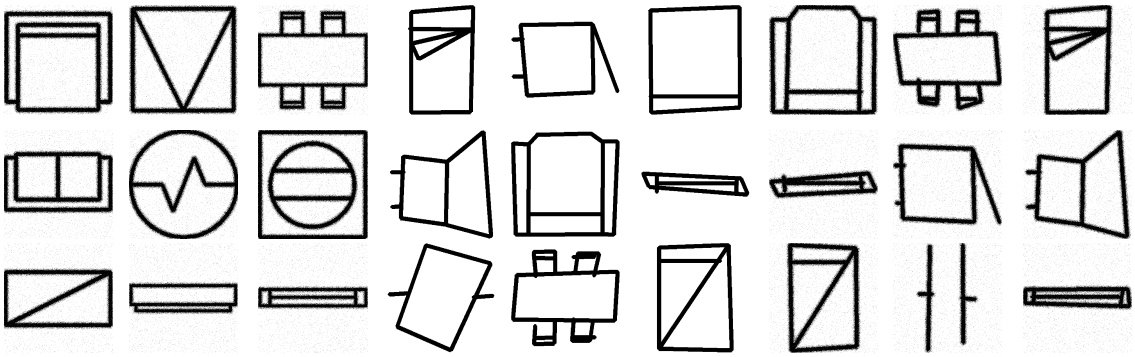
[14]http ://www.lems.brown.edu/vision/researchAreas/SIID/



(a) symboles de tailles différentes dans la base ***SR1*** (scale-set3)

(b) symboles obtenus par des rotations dans la base ***SR2*** (rotation-set3)

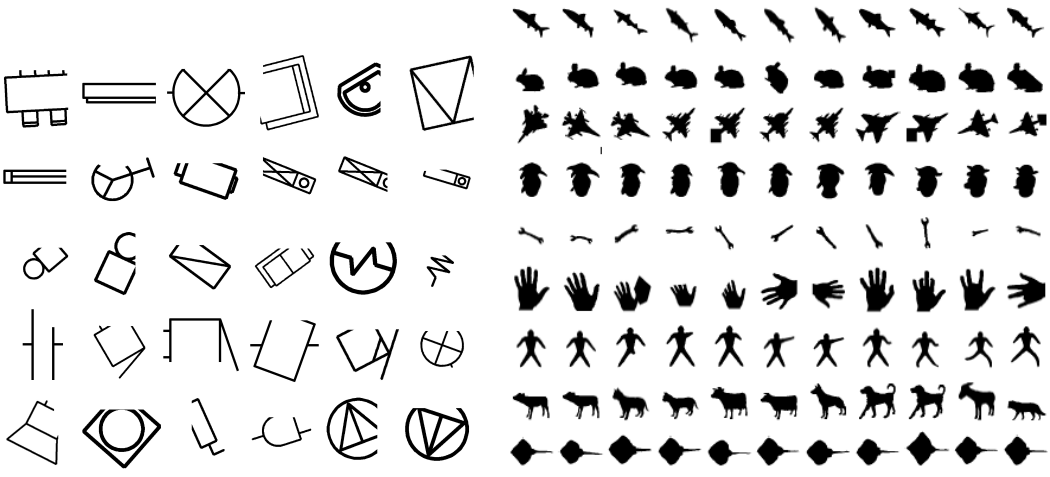
(c) symboles obtenus par rotation et changement d'échelle dans la base ***SR3*** (rotation-scale-set3)



(d) symboles dégradés dans la base ***DD1*** (degrad-level3-m3)

(e) symboles déformés dans la base ***DD2*** (distortion3-set2)

(f) symboles dégradés et déformés dans la base ***DD3*** (deform-degrad-level3-m3)



(g) symboles dans la base ***OC1*** (occlusion)

(h) base ***OC2*** (shapes99)

FIG. 4.8 – Exemples de symboles dans les bases de test.

4.3.3 Mesure de performance

Rappelons que, dans le domaine de la recherche d'informations, il existe de nombreuses méthodes pour évaluer un système de recherche. La plupart de ces méthodes se basent essentiellement sur deux mesures : la *précision* et le *rappel*. La *précision* mesure la qualité des résultats fournit par le système. Elle est calculée par le taux de réponses pertinentes parmi les réponses obtenues. Une précision élevée signifie que le système fournit peu de réponses incorrectes. Une précision parfaite est égale à 1. Le *rappel* mesure la quantité de réponses pertinentes obtenues par rapport à l'ensemble des réponses pertinentes existantes dans la base. Un rappel de 1 signifie que le système a fourni toutes les réponses qui répondent au besoin de la recherche. Un système parfait de recherche doit donc fournir des résultats dont la précision et le rappel sont égaux à 1. Néanmoins, en pratique, la précision et le rappel ont souvent une relation inverse au sens où l'augmentation de la valeur de l'un se produit au détriment de l'autre. Il est important de prendre en compte simultanément ces deux mesures pour fournir une évaluation exacte du système. Ces deux mesures sont ainsi souvent agrégées en une mesure unique (*F-score*) ou représentées sous la forme d'une courbe de *précision/rappel* [Baeza-Yates 99, Smith 98, Gevers 04, Mikolajczyk 05].

Dans l'objectif de rechercher des symboles similaires à une requête, classés selon leur degré de similarité, nous proposons d'utiliser la courbe de *précision/rappel* pour évaluer les performances de notre descripteur. Dans notre contexte, le **rappel** est défini comme le nombre de symboles pertinents retrouvés (Ra) au regard du nombre de symboles pertinents présents dans toute la base (R).

$$Rappel = \frac{|Ra|}{|R|} \tag{4.12}$$

La **précision** est le rapport entre le nombre de symboles trouvés pertinents et le nombre total de symboles trouvés (X).

$$Precision = \frac{|Ra|}{|X|} \tag{4.13}$$

Afin de tracer la courbe de *précison/rappel* pour un jeu de tests, nous calculons les valeurs de la précision et du rappel en faisant varier la valeur des r symboles les plus proches de la requête. Nous cherchons ici à évaluer la qualité de ce classement en allant de plus en plus loin dans le classement proposé. Nous testons donc les valeurs de r comprises entre 1 et K, avec K, le nombre d'occurrences du symbole requête dans la base. Pour chaque valeur de r, on se propose de calculer la précision et le rappel obtenus pour chaque symbole requête et d'en définir les valeurs moyennes qui seront utilisées comme données pour tracer la courbe de performance du descripteur. Plus la courbe est élevée, meilleure est la performance obtenue.

4.3.4 Résultats expérimentaux

4.3.4.1 Contexte de forme vs CFPI

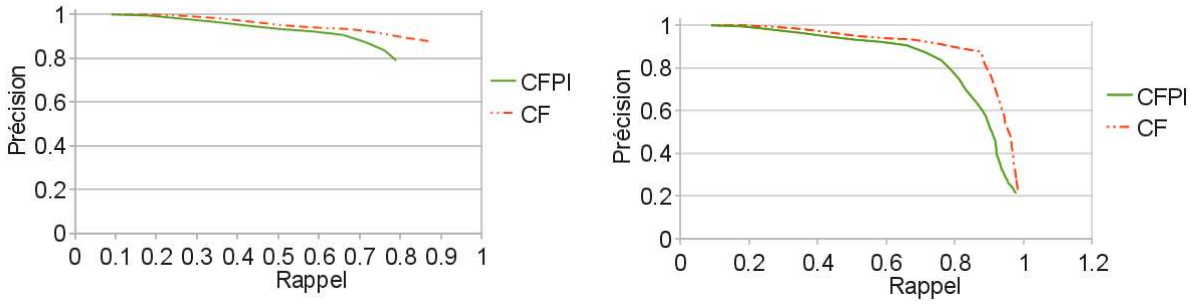
Comme notre descripteur se base sur le *contexte de forme*, nous faisons tout d'abord des analyses sur le CFPI et le *contexte de forme*. L'adaptation du *contexte de forme* pour les points d'intérêt présente un avantage sur la complexité de la représentation du symbole. Soit N_{pc}

le nombre de points de contour d'un symbole, N_{pi} le nombre de points d'intérêt extraits, L le nombre de bins du $CFPI$, le symbole est donc décrit par N_{pc} vecteurs de taille L avec le *contexte de forme* et par N_{pi} vecteurs de même taille avec $CFPI$. N_{pi} est généralement beaucoup plus petit que N_{pc}. Quelques exemples de ces valeurs ainsi que des précisions des résultats obtenus (P_{CF}, P_{CFPI}) pour chaque descripteur sont introduits dans le Tab. 4.1. Nous remarquons de grandes différences entre le nombre de points de contour et le nombre de points d'intérêt retenus pour chaque symbole. Cela produit de grandes différences en terme de complexité de calcul des descripteurs.

						
N_{pc}	5415	4529	3267	438	659	565
N_{pi}	149	29	39	18	30	37
$T_{CFPI}(s)$	1.84	1.39	1.34	0.09	0.1	0.1
$T_{CF}(s)$	54.87	22.87	8.29	0.76	0.44	0.36
P_{CFPI}	5/5	4/5	4/5	9/11	8/11	11/11
P_{CF}	5/5	4/5	3/5	10/11	11/11	11/11

TAB. 4.1 – Complexité du CFPI et du *contexte de forme*. N_{pc} : le nombre des points de contour, N_{pi} : le nombre des points d'intérêt détectés pour un symbole, $T_{CFPI}(s)$ et $T_{CF}(s)$: le temps de calculs (en seconde) pour CFPIs et *contextes de forme* d'un symbole, P_{CFPI} et P_{CF} : les précisions obtenues pour chaque descripteur .

De plus, dans la Fig. 4.9, nous présentons les résultats de tests en utilisant respectivement le CFPI et le *contexte de forme* comme descripteur. La mesure de similarité est identique pour les deux descripteurs. Cette expérimentation est effectuée sur la base ***OC2*** en prenant chaque élément de la base comme requête pour chercher les symboles similaires dans la base. Le Tab. 4.2 montre le temps nécessaire de ce test lors de recherches de symboles similaires correspondant à chaque descripteur. Ces valeurs ne tiennent pas compte du temps de calcul des descripteurs pour chaque symbole dans la base. Nous constatons qu'il n'y a pas de différence significative entre la



(a) les 11 plus proches

(b) les 50 plus proches

FIG. 4.9 – Courbes de précision/rappel moyennes pour la recherche de symboles similaires à partir de 99 requêtes dans la base ***OC2*** en utilisant CFPI et *contexte de forme*.

	Nombre des requêtes	Temps total (s)	Temps moyen par une requête (s)
$CFPI$	99	58.8	0.59
CF	99	3163.8	31.96

TAB. 4.2 – Temps de calculs pour effectuer des recherches dans la base ***OC2*** en utilisant CFPI et *contexte de forme* (installation en matlab).

performance en termes de taux de reconnaissance du CFPI et celle du *contexte de forme* lors de la recherche de symboles similaires. En effet, malgré un nombre réduit de vecteurs représentant un symbole, le CFPI fournit les résultats presque similaires à ceux du *contexte de forme*, surtout pour les premiers les plus proches (Fig. 4.9(a)). Ces courbes tracent la précision / rappel moyen jusqu'aux 11 plus similaires. Si on prend les 50 symboles les plus proches, la précision atteint 22% avec les deux descripteurs, tandis que le rappel atteint 98% et 99% respectivement pour le CFPI et le *contexte de forme*. En utilisant les points d'intérêt, nous pouvons raisonnablement réduire la complexité de la représentation d'un symbole et aussi l'appariement entre symboles.

En fait, en utilisant les points d'intérêt, nous gardons les points importants pour représenter un symbole. Avec le *contexte de forme*, le nombre de vecteurs caractéristiques pour chaque symbole peut être diminué en utilisant seulement un échantillonnage uniforme du contour. Toutefois, cet échantillonnage ne garantit pas la préservation des points discriminants du symbole. Nous présentons cet aspect en Fig. 4.10. Dans cette figure, CF-1 désigne les *contextes de forme* calculés sur tous les points de contour, CF-2 désigne les *contextes de forme* calculés aux points de contours échantillonés pour que le nombre de points considérés soit égal au nombre de points d'intérêt utilisés au niveau du CFPI. Nous remarquons qu'avec le même nombre de points pour représenter un symbole, le résultat obtenu par le CFPI est nettement meilleur que celui obtenu par le CF-2. Cela signifie que les points d'intérêt sont plus discriminants que des points échantillonnés sur les contours. Nous présentons dans le Tab. 4.3 un exemple permettant d'illustrer cet impact visuellement.

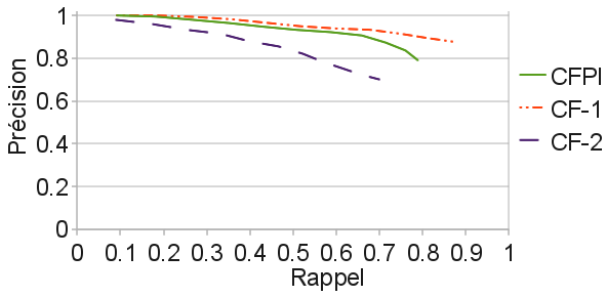
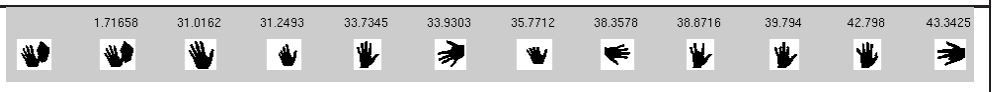
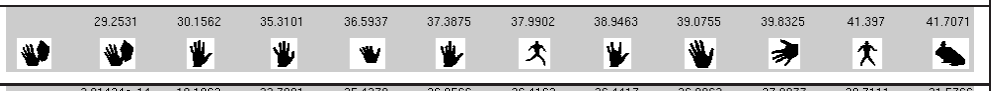


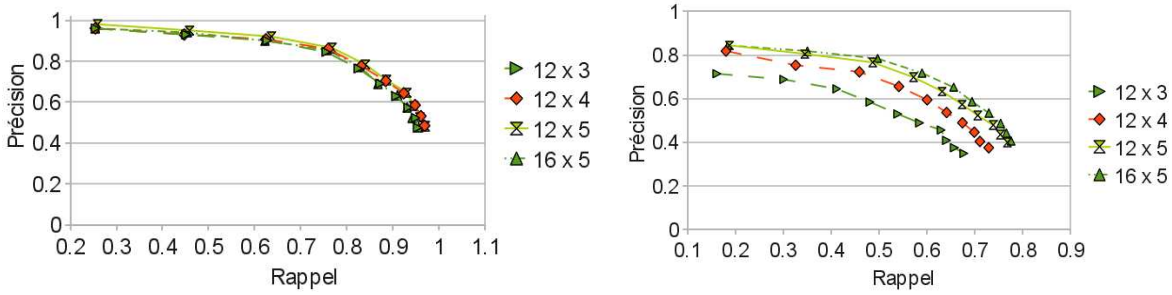
FIG. 4.10 – Échantillonnage uniforme des points de contour vs Point d'intérêt. Les courbes présentent les résultats obtenus lors de la recherche de symboles similaires à partir de 99 requêtes dans la base ***OC2*** en utilisant CFPI, *contexte de forme* avec tous les points de contour (CF-1) et *contexte de forme* avec un ensemble échantillonné (CF-2) dont le nombre de points retenus est égal au nombre de points d'intérêt utilisés pour calculer le CFPI.

Concernant la taille du CFPI, nous avons testé différentes tailles en jouant sur le nombre

	N°	La requête et ses plus proches
CF-1	899	
CF-2	32	
CFPI	32	

TAB. 4.3 – Exemple visuel de résultat obtenu avec CF-1 (la première ligne), CF-2 (la deuxième ligne), CFPI (la troisième ligne) pour la même requête. N° : le nombre de contexte de formes pour la représentation des symboles.

d'intervalles pour les décompositions angulaires et radiales : 12×3, 12×4, 12×5, 16×5. Notre premier test a consisté à chercher des symboles similaires dans la base ***SR3*** pour des symboles parfaits de l'ensemble **A**. Nous pouvons constater qu'il n'y a pas de différences significatives en termes de précision/rappel pour les différentes tailles testées (Fig. 4.11(a)). Ainsi, nous pouvons conclure que pour la représentation de symboles parfaitement segmentés, un descripteur de taille 36 est suffisant. Dans le cadre de documents graphiques, les symboles sont rarement parfaitement segmentés et il est donc important de procéder à des tests sur des symboles incomplets. Nous proposons donc de tester les différentes tailles définies du CFPI pour la recherche de symboles similaires dans la base ***SR3*** en utilisant les symboles occultés de la base ***OC1*** comme requêtes. Nous pouvons constater que dans ce cas, les CFPI de petites tailles sont moins discriminants que ceux de tailles plus élevées (Fig. 4.11(b)). En revanche, l'écart de performance entre le CFPI de taille 60(12×5) et celui de taille 80(16×5) est très faible. Pour conclure sur la taille du CFPI, nous proposons, afin d'avoir un compromis raisonnable entre la dimension du descripteur et sa performance, de choisir une taille de 60 pour nos expérimentations dans cette thèse.



(a) pour les symboles parfaits (b) pour les symboles occultés

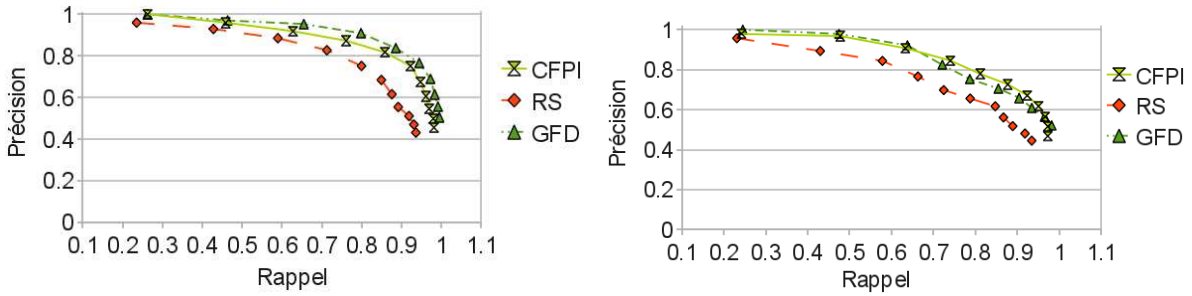
FIG. 4.11 – Effet de la taille du descripteur CFPI sur la performance de recherche.

4.3.4.2 CFPI vs $\mathcal{R}$-signature et GFD (*Generic Fourier Descriptor*)

Dans la suite, nous évaluons la performance de notre descripteur sur différentes bases en comparant nos résultats avec ceux d'autres descripteurs de formes : *GFD* [Zhang 02c] et $\mathcal{R}$-*signature* (nommée *RS* sur les figures) [Tabbone 06b]. À cause de la complexité du *contexte de*

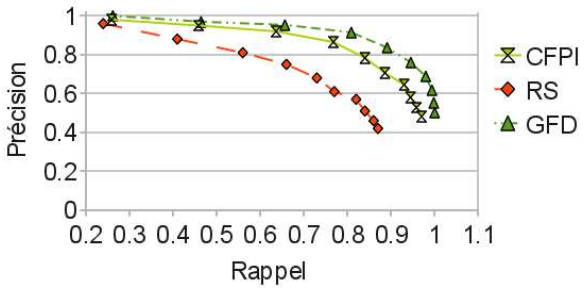
forme, nous ne présentons pas d'expérimentations supplémentaires avec le *contexte de forme* sur d'autres bases de test.

Notre première expérimentation est dédiée à l'étude de l'invariance aux transformations des descripteurs. Nous utilisons les modèles de l'ensemble **A** comme requêtes pour chercher des symboles similaires dans les bases ***SR1, SR2, SR3***. Les résultats moyens calculés sur 50 requêtes de **A** sont présentés en Fig. 4.12. Nous constatons que, quelles que soient les transformations appliquées, rotation, changement d'échelle ou combinaison des deux, la $\mathcal{R}$-*signature* donne des résultats moins bons que les deux autres. Le descripteur CFPI fournit des résultats assez similaires que GFD. Le résultats obtenus avec le CFPI sont un peu moins bons qu'avec le GFD sur la base ***SR1*** et ***SR3***. Cependant, le GFD est un descripteur global donc il est difficile de l'adapter pour représenter les informations locales dans un document.



(a) sur la base ***SR1*** avec des symboles obtenus par changement d'échelle

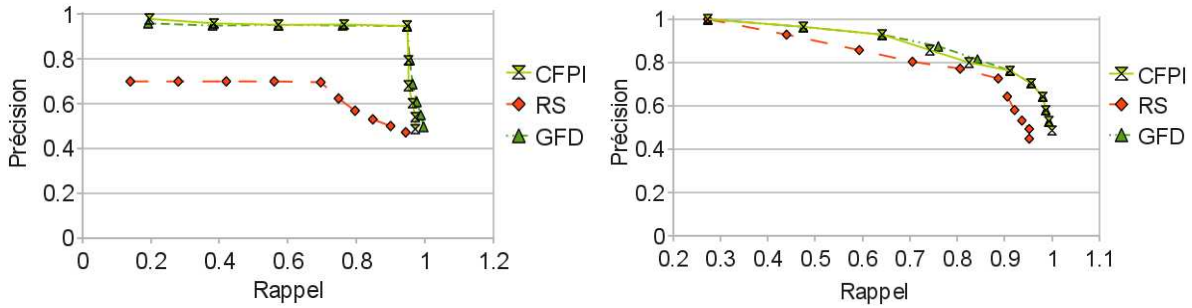
(b) sur la base ***SR2*** avec des symboles obtenus par rotation



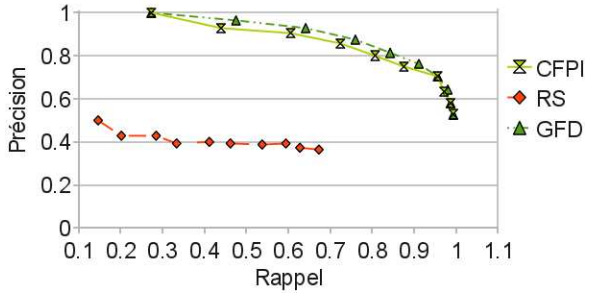
(c) sur la base ***SR3*** avec des symboles obtenus par rotation et changement d'échelle

FIG. 4.12 – Courbes de précision/rappel moyennes pour la recherche de symboles similaires à partir de 50 symboles requêtes dans les trois bases ***SR1***, ***SR2*** et ***SR3***.

Nous proposons comme seconde expérimentation d'examiner le comportement des descripteurs pour les symboles dégradés et/ou déformés. Comme pour la première expérimentation, nous utilisons des éléments de **A** comme symboles requêtes pour chercher leurs occurrences dans les bases ***DD1, DD2, DD3***. Les résultats sont présentés en Fig. 4.13. Nous avons obtenu des résultats similaires pour les descripteurs *GFD* et CFPI sur les trois bases ***DD1, DD2, DD3***. La $\mathcal{R}$-*signature* donne des résultats moins bons pour des symboles dégradés ou avec déformations. Sa performance est diminuée davantage sur la base qui contient des symboles à la fois déformés et dégradés (***DD3***).



(a) sur la base ***DD1*** contenant des symboles dégradés (b) sur la base ***DD2*** contenant des symboles déformés

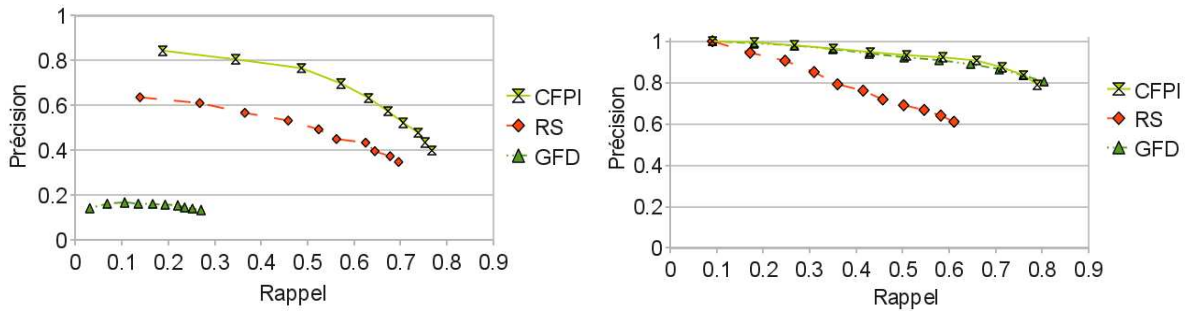


(c) sur la base ***DD3*** contenant des symboles dégradés et déformés

FIG. 4.13 – Courbes de précision/rappel moyennes pour la recherche de symboles similaires à partir de 15 (50 pour la base ***DD1***) symboles requêtes dans les trois bases ***DD1***, ***DD2*** et ***DD3***.

Finalement, afin d'examiner la robustesse du descripteur CFPI aux occlusions et aux variations de forme, nous proposons de le tester sur les bases ***OC1*** et ***OC2***. Pour la base ***OC1*** que nous avons créée manuellement, nous prenons comme symboles requêtes les éléments de cette base et nous essayons de chercher leurs occurrences dans l'ensemble **SR3**. La Fig. 4.14(a) présente les résultats moyens obtenus à partir de 77 requêtes pour les trois descripteurs. Pour la base ***OC2***, nous utilisons comme requêtes des symboles de cette base et nous cherchons leurs occurrences dans le reste de la base. Les résultats moyens obtenus sont présentés en Fig. 4.14(b). Nous remarquons que le CFPI et le GFD ont des résultats presque identiques sur la base ***OC2***. Par contre, sur la base ***OC1*** contennant des symboles graphiques, le GFD donne de moins bons résultats. En effet, la partie occultée du symbole influence fortement le calcul de la position du centre de gravité de l'objet et aussi le calcul du descripteur global GFD du symbole. La $\mathcal{R}$-*signature* s'avère meilleure que le GFD dans le cas des symboles graphiques. Ses performances demeurent néanmoins en deça de celles du CFPI. Les résultats obtenus sur ces deux bases montrent que le CFPI est plus robuste aux occlusions et aux déformations légères que les deux autres descripteurs.

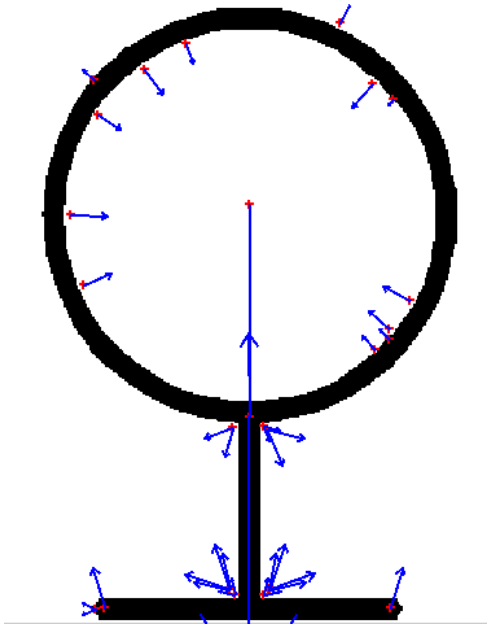
Bien que notre descripteur ait de bonnes performances sur ces expérimentations, nous pouvons observer certains inconvénients. En effet, comme celui-ci se base sur les points d'intérêt détectés par le détecteur *SIFT*, il présente quelques instabilités au sens de la répétition (cf. section 4.2.1) au niveau des courbes. Pour le symbole requête composant de courbes simples, avec peu de points d'intérêt stables, la représentation du symbole avec CFPI est donc moins discrim-



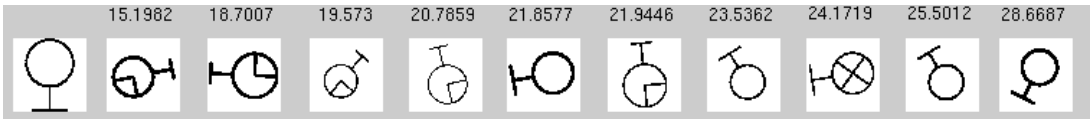
(a) sur la base ***OC1*** (b) sur la base ***OC2***

FIG. 4.14 – Résultats des tests de comportement des descripteurs sur des symboles ayant des occlusions et/ou des déformations.

inante qu'elle ne l'est généralement pour d'autres symboles. Ceci se traduit par une baisse de la performance de recherche de symboles similaires (voir Fig. 4.15).



(a) Les points d'intérêt détectés



(b) La requête (à gauche) et les dix résultats plus proches

FIG. 4.15 – Instabilité des points d'intérêt sur la performance de la recherche des symboles similaires. (a) Les points d'intérêt détectés (marqués par un '+' rouge) avec leurs orientations (l'orientation des flèches) et leurs échelles (les grandeurs des flèches indiquent relativement les échelles où les points sont détectés). (b) La requête et les dix résultats plus proches.

4.4 Conclusion

Dans ce chapitre, nous avons défini un descripteur de symboles adapté aux symboles graphiques, le *CFPI (Contexte de Forme pour les Points d'Intérêt)*. Ainsi, nous avons proposé une adaptation du contexte de forme pour décrire des symboles graphiques en prenant compte seulement des informations associées aux points d'intérêt. Le descripteur fournit une bonne représentation de la configuration locale correspondant à chaque point d'intérêt et nous permet de réduire la complexité de description d'un symbole par rapport aux contextes de forme (*shape contexts*). Ce descripteur est simple et invariant à la rotation et au changement d'échelles. Il est aussi robuste à la dégradation, à la déformation et aux occlusions. Afin d'évaluer ces caractéristiques, nous avons procédé à des tests expérimentaux à partir de différentes bases de symboles isolés. Nous avons montré que notre descripteur donne de bons résultats par rapport à la $\mathcal{R}$-*signature* et au *GFD*. De plus, comme notre descripteur prend en compte les informations locales aux points d'intérêt, il s'avère particulièrement robuste aux occlusions, ce qui est un critère important dans le cadre d'un système de localisation de symboles dans des documents. Nous reviendrons dans le chapitre 5 sur l'utilisation de ce descripteur pour la représentation des documents graphiques pour résoudre le problème de la localisation de symboles.

Chapitre 5

Localisation de symboles dans les documents graphiques

Sommaire

Dans ce chapitre, nous présentons notre approche pour résoudre le problème de la localisation de symboles dans les documents graphiques. Notre approche se place à mi-chemin entre le domaine de la recherche d'informations textuelles et celui de la recherche d'images par le contenu. Elle se base sur des techniques classiques d'indexation et de recherche de documents textuels par l'utilisation de la notion de mots visuels. Cette idée est inspirée de travaux précédents en recherche de vidéos [Sivic 06].

Comme précisé dans le chapitre 3, les quelques méthodes de localisation de symboles apparaissant dans la littérature s'appuient sur des pré-traitements tels qu'une étape de vectorisation, l'ajout d'hypothèses sur les symboles, et/ou une étape d'apprentissage. À l'inverse notre approche de localisation de symboles ne nécessite aucune hypothèse sur le symbole, ni d'étape "lourde" de segmentation[15] des documents.

[15]Nous considérons comme segmentation toutes techniques préalables de décomposition du document en primitives quelconques.

La Fig. 5.1 montre le schéma général de notre approche. Celle-ci se compose de deux étapes. La première étape consiste à caractériser et à indexer les documents graphiques. Cette étape est effectuée hors-ligne. La deuxième étape est dédiée à la localisation de symboles dans ces documents. Nous proposons pour cela d'utiliser une extension du descripteur CFPI (décrit dans le chapitre 4) pour décrire le contenu des documents. En effet, ce descripteur dispose de propriétés qui nous intéressent particulièrement dans le cadre de notre problème de localisation de symboles : il est invariant aux transformations géométriques, robuste au bruit, stable aux distorsions et surtout tolérant aux occlusions. Un ensemble de mots visuels est construit par regroupement de descriptions (CFPIs) similaires extraits des documents. Ces documents sont ensuite indexés grâce à ces mots visuels. Lorsqu'un symbole requête est recherché dans ces documents, les CFPIs de ce symbole sont calculés puis mis en correspondance avec les mots visuels. Des régions candidates (*ROI*) pouvant contenir des occurrences du symbole requête sont ensuite identifiées par analyse des documents contenant les mêmes mots visuels que ceux de la requête. La détermination des régions contenant les occurrences du symbole requête est opérée à l'aide d'un système de vote en appliquant le modèle vectoriel de la recherche d'information textuelle. Nous détaillons plus précisément ces différents aspects au fil des sections suivantes.

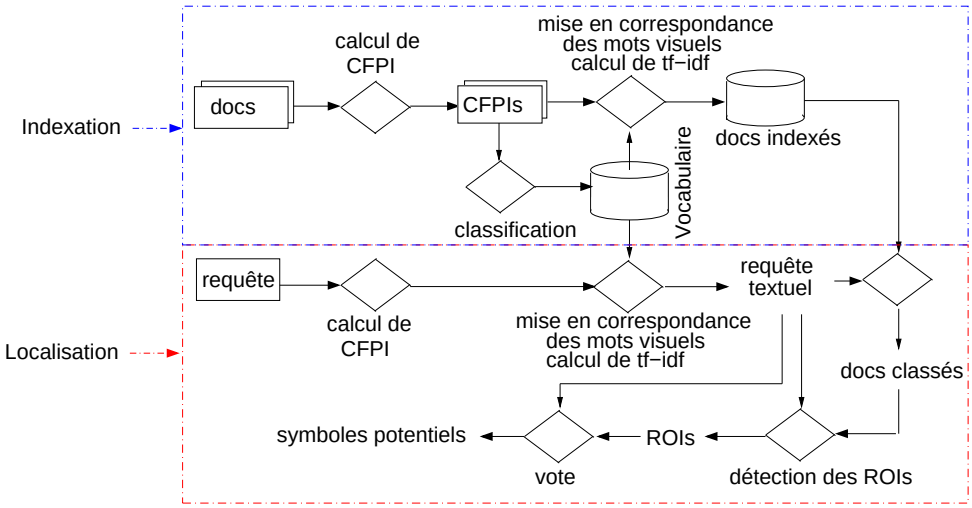


FIG. 5.1 – Schéma de l'approche proposée.

5.1 De l'image au document *"textuel"*

La recherche d'images est un sous-domaine de la recherche d'information (*Information Retrieval - IR*). Les méthodes de recherche d'images peuvent être divisées en deux catégories. La première regroupe les méthodes se basant sur l'annotation textuelle (*text-based image retrieval*), la seconde, les méthodes se basant sur le contenu de l'image (*Content-Based Image Retrieval - CBIR*) [Gevers 04, Liu 07]. Dans la première catégorie de méthodes, les images sont récupérées soit grâce à la description contextuelle des images (descriptions textuelles localisées autour de l'image), soit grâce à des mots clés associés directement à l'image. L'utilisation de la description contextuelle de l'image permet d'éviter le problème d'annotation manuelle des images, qui est

un grand problème surtout pour des grandes bases, et elle est bien appropriée à la recherche d'images sur le web, par exemple avec Google ou Alta Vista. Les approches se basant sur les mots clés ont pour avantage de véhiculer des concepts sémantiques au travers de ces mots, mais, en contrepartie, elles sont fortement dépendantes de la qualité de la description et de son caractère subjectif. A l'inverse, dans la deuxième catégorie de méthodes (*CBIR*), chaque image est représentée par ses caractéristiques telles que sa texture, sa couleur, sa forme, etc. Ainsi, la description d'une image est uniquement basée sur des caractéristiques visuelles et ne dépend pas d'agents externes. Ce type de description ne fournit pas d'informations sémantiques sur l'image et pose le problème du fossé sémantique[16]. Pour profiter à la fois des avantages de l'information textuelle et visuelle de l'image, les deux types d'informations peuvent être combinés [Barrat 08]. Une telle combinaison a été utilisée dans différents systèmes tels que ImageRover, Diogenes, WebSeer, WebSeek ou atlas WISE [Shapiro 01, Long 02, Kherfi 04, Gevers 04].

Mettre en place un système de recherche d'images par le contenu nécessite de choisir un descripteur pour représenter les images ainsi qu'une méthode d'appariement entre images. Une première approche pour la représentation des images consiste à utiliser les propriétés physiques de l'image pour construire un descripteur global (e.g. *couleurs, textures, ...*). Une image sera ainsi représentée par un vecteur numérique. Définir la similarité entre deux images consistera donc à calculer "l'écart" entre leurs deux vecteurs correspondants. Ces méthodes permettent de fournir rapidement des résultats mais avec une faible précision. En effet, un descripteur global ne prend pas en compte les différentes propriétés des objets présents dans l'image ni leurs relations. Une autre approche pour représenter les images consiste à utiliser des descripteurs locaux afin de caractériser des régions locales de l'image. La similarité entre images est déterminée par le nombre de sous-régions similaires entre elles. Un des problèmes de cette approche est sa complexité lors de la recherche dans de grandes bases de données en raison du grand nombre de régions pouvant caractériser une image [Tirilly 09].

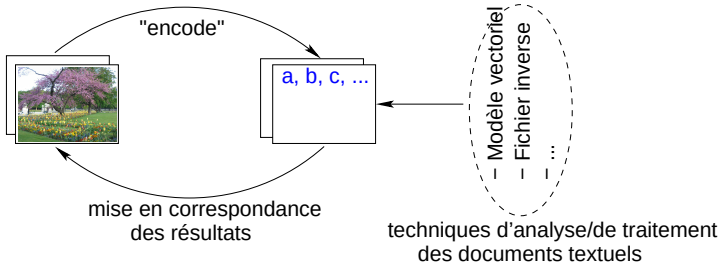


FIG. 5.2 – Utilisation de techniques de recherche d'informations textuelles pour la recherche d'images par le contenu.

Les travaux actuels sur la recherche d'images par le contenu se basent essentiellement sur la notion de *mots visuels* ou *sac de primitives*. Il s'agit d'adapter des techniques bien établies dans le domaine de la recherche d'informations textuelles aux systèmes de recherche d'images par le contenu (Fig. 5.2). Sur cette figure, l'idée est de textualiser l'image pour y appliquer des techniques d'analyse/de traitement des documents textuels.

Les premiers travaux proposant d'exploiter des techniques de la recherche d'informations

[16]Traduit de "semantic gap"

textuelles pour la recherche d'images furent ceux de [Squire 00, Zhu 00]. Squire et al. ont ainsi proposé d'utiliser les fréquences pondérées d'apparition de certaines caractéristiques dans chacune des images de la base pour calculer la similarité entre deux images et mettre à jour le score dans le retour de pertinence. [Zhu 00] a introduit pour la première fois, à notre connaissance, la notion de mots clés (nommée *"keyblock"*) dans un système de recherche d'images par le contenu. Le vocabulaire visuel (le *"codebook"*) est construit par une méthode de clustering utilisée sur des blocs de tailles différentes (2x2, 4x4, 8x8, 16x16) des images. Cette technique est inspirée d'une technique provenant de la compression d'images, la quantification vectorielle [Idris 96, Lu 99]. Chaque bloc d'une image est remplacé par l'entrée la plus proche du vocabulaire. Il est ensuite possible d'appliquer sur ces images différents modèles de recherche de textes tels que le modèle binaire ou le modèle vectoriel. Bien que n'ayant été défini que récemment, la notion de mots visuels a connu de nombreuses applications dans les domaines de la vision par ordinateur tels que la recherche d'images, de vidéos [Sivic 03], la détection d'objets [Agarwal 04], la classification ou la reconnaissance de scènes [Csurka 04, Jurie 05, Li 05, Bosch 06, Nilsback 06, Lazebnik 06, Pham 09].

La construction des mots visuels demande de faire différents choix. Elle nécessite en effet de choisir un descripteur pour décrire les images. Elle demande également de définir les parties des images qui seront utilisées pour cette construction. Enfin, elle demande de choisir un algorithme de classification. Concernant le choix d'un descripteur pour décrire les images, tous les descripteurs de bas niveaux (couleurs, texture, etc.) peuvent être utilisés. Des descripteurs plus complexes peuvent également être utilisés. Dans ce cadre, un descripteur particulièrement performant est le descripteur SIFT. Il existe de nombreuses variations de ce descripteur telles que GLOH [Mikolajczyk 05], PCA-SIFT [Ke 04], SURF [Bay 08], HoG [Dalal 05]. Pour augmenter la qualité de la discrimination entre objets, le vocabulaire peut être construit à partir de plusieurs propriétés telles que la couleur, la forme ou la texture [Nilsback 06]. Le descripteur choisi sera utilisé pour caractériser différentes parties de l'image considérée. Ces parties peuvent être des régions obtenues par un découpage régulier de l'image ([Zhu 00, Pham 09]) ou des régions d'intérêt (ex : régions associées aux points d'intérêt) [Csurka 04, Agarwal 04]. Pour la classification d'images naturelles, l'utilisation des régions par découpage régulier est généralement plus performante que l'utilisation seule de régions d'intérêt [Li 05, Jurie 05, Nowak 06]. Le dernier point à définir concerne le choix d'une méthode de clustering pour la construction d'un vocabulaire visuel. Cette méthode est appliquée sur les valeurs des descripteurs obtenues pour les différentes parties des images. Il existe dans la littérature de nombreuses méthodes de clustering. Ces dernières peuvent être regroupées en deux catégories : celles basées sur un clustering de type *k-means* et celles basées sur un clustering de type *agglomératif* [Larlus 06]. Malgré le problème du choix du nombre de clusters, les méthodes basées sur un clustering de type *k-means* sont souvent utilisées en raison de leur simplicité. Les méthodes s'appuyant sur un clustering de type *agglomératif* nécessitent, elles, de faire face au problème du nombre de vecteurs de description. Une solution pour bien s'adapter au traitement de grosses quantités de vecteurs, de permettre une grande souplesse dans le choix du nombre de clusters, consiste à combiner plusieurs méthodes [Jurie 05].

Une fois les images *textualisées*, des techniques de traitement automatique des textes ou des langues peuvent être utilisées sur ces dernières pour chercher des images. Il s'agit de techniques

bien établies dans les domaines de la recherche d'informations textuelles ([Zhu 00, Sivic 03]), de la classification de textes [Csurka 04] et du traitement automatique des langues [Bosch 06, Hörster 08, Pham 09, Li 05].

L'utilisation de mots visuels permet de recourir à des techniques d'analyse de textes. Cependant, il existe des différences majeures entre les documents visuels et les documents textuels [Tirilly 09]. Dans les documents textuels, le vocabulaire se compose de mots apparaissant dans les documents et contient des propriétés particulières propres à la langue des documents. En revanche, la richesse et la qualité d'un vocabulaire visuel dépendent, elles, de plusieurs facteurs tels que le détecteur de régions, le descripteur et la méthode de clustering utilisée. De plus, l'apport sémantique d'un mot visuel est plus faible que celui d'un mot textuel. Un mot textuel peut désigner un concept concret lorsqu'il faut plusieurs mots visuels pour signifier la même chose car un mot visuel ne représente qu'une partie de l'objet. Cet apport sémantique fort permet de construire des requêtes textuelles très courtes, alors qu'il est souvent nécessaire d'utiliser plusieurs mots visuels pour formuler une requête.

Construire un vocabulaire visuel requiert de faire un certain nombre de choix. Ces choix peuvent avoir un impact important dans les mots générés. Il est possible d'atténuer l'impact de ces choix en combinant plusieurs détecteurs ou/et descripteurs [Sivic 03, Nilsback 06]. Néanmoins, se pose toujours le problème du clustering. Il n'existe à l'heure actuelle pas de méthodes permettant de définir un clustering optimal. Cependant, certaines techniques peuvent permettre d'améliorer celui-ci. Ainsi, Gemert et al. [van Gemert 09] montrent que l'exploitation des ambiguïtés présentes dans les mots visuels permet d'améliorer la pertinence de la classification des images. Elle permet également d'éviter une dégradation des performances du système lors de l'utilisation d'un vocabulaire trop large.

Nous proposons dans ce travail de thèse d'adapter l'approche des mots visuels aux documents graphiques. Pour la représentation des documents, nous avons choisi la représentation éparse (e.g. informations aux points d'intérêt) en regard à la nature des images considérées. Concernant la technique de clustering utilisée pour construire les mots visuels, nous avons opté pour une technique simple mais bien établie, le *k-means*. Enfin, pour la mise en correspondance des caractéristiques des images avec des mots visuels, nous avons choisi d'utiliser un appariement flou. Cette méthode a en effet l'avantage d'atténuer la dépendance vis-à-vis de la taille du vocabulaire [van Gemert 09] et de réduire les faux appariements de points qui se trouvent aux frontières des clusters (des mots visuels).

5.2 Indexation d'images par un vocabulaire visuel

5.2.1 Descripteur local : CFPI au niveau du document

Dans le chapitre 4, nous avons défini le CFPI comme la distribution des points de contour du symbole par rapport aux points d'intérêt. Dans les documents, les symboles ne sont pas segmentés, il est donc nécessaire de définir une région de voisinage pour chaque point d'intérêt afin d'en calculer son CFPI. Il est cependant complexe de définir au préalable une taille pour les régions. En effet, la résolution des symboles contenus dans les documents peut-être différente

d'un symbole à un autre. Nous proposons donc, afin de capturer avec pertinence les informations aux différentes résolutions et de définir la région de voisinage d'un point d'intérêt en fonction de la résolution où le point est détecté. La région de voisinage $\mathcal{N}_i$ d'un point $p_i = (x_i, y_i, \delta_i, \theta_i)$ pour calculer le CFPI est déterminée par une ellipse dont le centre est p_i et la direction du grand axe est l'orientation du point d'intérêt. La longueur du demi-grand axe est déterminée par $a = \beta\delta_i$ et la longueur du demi-petit axe par $b = \gamma\delta_i$. En pratique, nous avons choisi $\gamma = 2*\beta$ et b est égal à la taille du filtre *DoG* utilisé pour déterminer ce point. La raison de ces choix s'explique par le fait que nous souhaitons que chaque région contienne le plus d'information possible sur le tracé auquel le point d'intérêt est censé être rattaché. Ainsi, un point d'intérêt ne caractérise l'élément du document qu'à une seule unique résolution. Il est néanmoins possible de caractériser un élément à différentes résolutions grâce aux points d'intérêt obtenus à différentes résolutions. Nous montrons en Fig. 5.3 un exemple de points d'intérêt et des régions de voisinage correspondantes utilisées pour le calcul des CFPIs pour un sous-document. Les points ('+') bleus désignent les points d'intérêt, les ellipses délimitent les régions de voisinage déterminées par rapport à la résolution où les points d'intérêt ont été détectés.

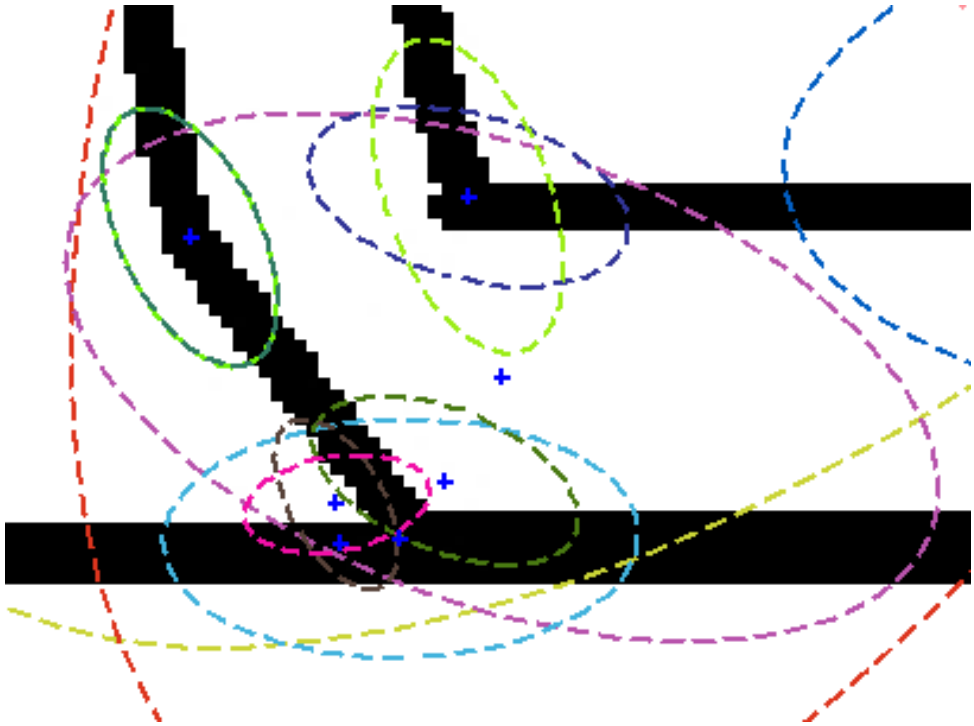


FIG. 5.3 – Détermination des régions de voisinage pour le calcul des CFPI dans le document. Les points ('+') bleus désignent les points d'intérêt, les ellipses délimitent les régions de voisinage déterminées par rapport à la résolution où les points d'intérêt ont été détectés.

Le descripteur CFPI h_i, au point p_i, calculé dans sa région est défini par :

$$h_i(l) = \#\{q_j \neq p_i, q_j \in \mathcal{C} \cap \mathcal{N}_i : (q_j - p_i) \in bin(l)\}, l = \overline{1, L} \tag{5.1}$$

et un document $\mathcal{D}$ est donc représenté par l'ensemble des h_i tel que :

$$\mathcal{D} \equiv \{h_i | p_i \in \mathcal{IP}\} \tag{5.2}$$

Comme dans un document les symboles sont souvent proches ou connectés, tous les points de contour présents dans le voisinage d'un point d'intérêt n'appartiennent pas nécessairement au symbole, perturbant le calcul du CFPI. Cependant, comme chaque symbole est représenté par plusieurs CFPI basés sur des points d'intérêt à des résolutions différentes, la description du symbole sera affectée de façon moindre par ces points de contour extérieurs.

5.2.2 Construction du vocabulaire visuel

La première étape de la construction du vocabulaire visuel consiste à calculer les CFPIs pour l'ensemble des documents de la base considérée (cf. section 5.2.1). Ensuite, une technique de classification non-supervisée (clustering) est utilisée pour regrouper les descripteurs similaires en classes. Chaque classe est considérée comme un mot visuel identifié par le centre de la classe. Les descripteurs appartenant à cette classe sont considérés comme des occurrences d'un mot visuel dans la base. Un document graphique est ainsi représenté par un ensemble de mots visuels et peut être traité comme un document textuel. La figure 5.4 présente un exemple de structures divisées selon trois classes qui ont été définies à partir d'une base de documents. En principe, n'importe quelle méthode de classification peut être utilisée. Le choix d'une méthode pourra dépendre de la distribution des vecteurs. Ici nous avons utilisé la méthode des *k-means*.

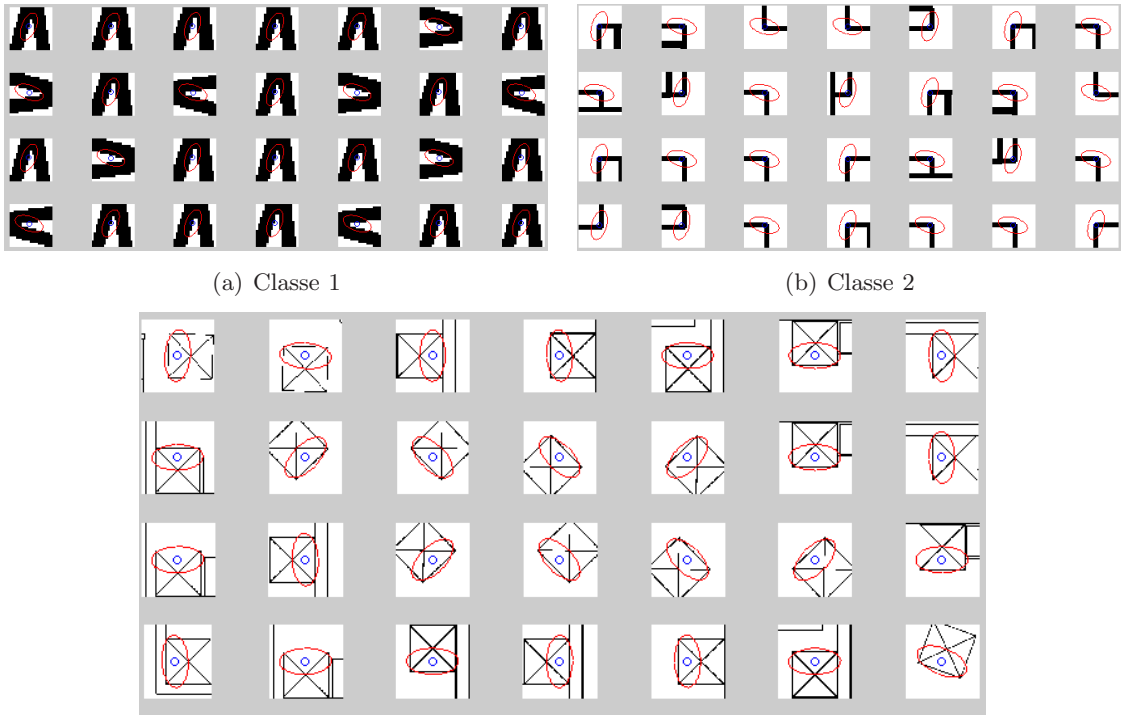
5.2.3 Mise en correspondance des descriptions locales aux mots visuels

Le vocabulaire visuel est construit pour que le CFPI associé à chaque point soit représenté par un mot visuel où chaque point d'intérêt est apparié à un mot visuel. L'objectif est de diminuer le nombre d'appariements redondants dans les étapes postérieures. Une difficulté de cet appariement est qu'un CFPI peut se trouver à la frontière entre deux classes (c'est-à-dire entre deux mots visuels). Un choix arbitraire de mot apparié avec ce point peut entraîner des erreurs. Afin de réduire ce problème, nous proposons d'effectuer des appariements multiples, ce qui signifie que le CFPI d'un point peut être apparié avec plusieurs mots visuels. Ainsi, pour un CFPI au point p_i donné, les mots dont la similarité est très proche sont choisis (V_i). Par soucis de clarté, nous parlerons de *mots correspondants de* p_i pour les désigner.

$$V_i = \left\{ w_j \in \mathcal{V} | \frac{sim_{p_i,w_j}}{sim_{p_i,w_0}} > \epsilon \right\} \tag{5.3}$$

$\mathcal{V}$ est le vocabulaire visuel, w_0 le mot le plus proche de p_i et ϵ est un seuil prédéfini (fixé à 0.96 dans nos résultats expérimentaux).

Nous définissons un degré de confiance pour chaque appariement. Ce dernier dépend du nombre de mots appariés avec le CFPI au point p_i et de leur similarité correspondante. Il est défini comme suit :



(a) Classe 1

(b) Classe 2

(c) Classe 3

FIG. 5.4 – Exemple de quelques structures (délimitées par les ellipses rouges) classées selon trois classes caractérisant trois mots visuels.

$$dConf_{p_i,w_j} = \frac{sim_{p_i,w_j}}{\sum_{w_k \in V_i} sim_{p_i,w_k}} \tag{5.4}$$

$$\sum_{w_j \in V_i} dConf_{p_i,w_j} = 1 \tag{5.5}$$

où V_i est l'ensemble des mots appariés avec p_i et sim_{p_i,w_j} la similarité entre le CFPI au point p_i et le mot w_j.

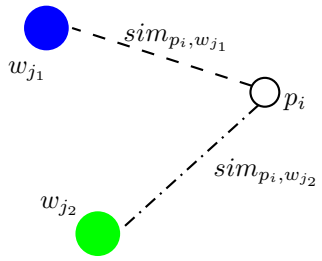


FIG. 5.5 – Mise en correspondance d'un vecteur CFPI avec plusieurs mots visuels.

5.2.4 Représentation des documents

Nous avons défini dans l'étape précédente un ensemble de mots visuels et l'appariement entre ces mots visuels et les CFPIs dans les documents. Ces documents peuvent être traités de façon similaire avec des documents textuels. Dans cette section, nous présentons deux techniques de la recherche d'informations textuelles que nous proposons d'utiliser pour les documents graphiques.

5.2.4.1 Modèle vectoriel

Le modèle vectoriel est une technique couramment utilisée en recherche d'informations car elle est plus performante que d'autres techniques classiques dans le cas général [Baeza-Yates 99]. Avec ce modèle, un document est représenté par un vecteur de fréquences d'apparition des mots. Ce vecteur de fréquences est souvent pondéré par l'importance relative des mots afin d'apporter un compromis entre deux facteurs : la fréquence d'apparition des termes (mots) dans le document (facteur *tf*) et l'estimation du degré d'importance de ce mot pour distinguer un document pertinent d'un document non pertinent (facteur *idf*).

Nous rappelons que les points d'intérêt pour un document j sont appariés avec des mots visuels. Un document j est représenté par un vecteur de *tf-idf* $\vec{s_j}$:

$$\vec{s_j} = \{f_{1,j}, f_{2,j}, .., f_{K,j}\} \tag{5.6}$$

où K est la taille du vocabulaire et $f_{i,j}$ la fréquence pondérée du mot i dans le document j :

$$f_{i,j} = tf_{i,j} * idf_i, i = \overline{1, K}$$

$$tf_{i,j} = \frac{freq_{i,j}}{max_l freq_{l,j}}; \quad idf_i = log\frac{N}{n_i}$$

avec $freq_{i,j}$ la fréquence d'apparition du mot i dans le document j, N le nombre total de documents dans la base et n_i le nombre de documents dans lesquels le mot i apparaît. Ainsi, $tf_{i,j}$ représente la fréquence normalisée du terme.

Dans le cas où un point est apparié avec plus d'un mot visuel, nous avons adapté ce modèle. Plus précisement, la fréquence $freq_{i,j}$ d'apparition du mot i dans le document j est redéfinie par (5.7) où $\mathcal{W}_j^D = \{w_1, w_2, .., w_M\}$ est l'ensemble des mots associés aux points d'intérêt dans le document j avec des degrés de confiance respectifs $\{dConf_1, dConf_1, .., dConf_M\}$:

$$freq_{i,j}^F = \frac{\sum_{w_m \in \mathcal{W}_j^D, w_m \equiv i} dConf_m}{\sum_{m=1}^{M} dConf_m} \tag{5.7}$$

Pour déterminer les documents similaires à la requête, le vecteur *tf-idf* de la requête $\vec{s_q}$ est calculé de la même façon : 1) déterminer les descripteurs CFPI, puis, 2) mettre en correspondance ces descripteurs avec des mots visuels et enfin, 3) calculer les fréquences pondérées des mots existants dans la requête. Le degré de similarité entre la requête et un document j de la base de données est quantifié par leur corrélation mesurant l'angle de projection entre les deux vecteurs $\vec{s_j}$ et $\vec{s_q}$.

$$sim(s_q, s_j) = \frac{\vec{s_j} \bullet \vec{s_q}}{|\vec{s_j}| \times |\vec{s_q}|} \tag{5.8}$$

Les degrés de similarité entre la requête et les documents de la base sont utilisés afin d'obtenir une liste de documents ordonnés par ordre de pertinence.

5.2.4.2 Fichier inverse

Dans le cas d'un document graphique pouvant contenir plusieurs symboles non-segmentés, le vecteur *tf-idf* n'est pas suffisamment représentatif pour permettre de localiser précisément les symboles (il contient uniquement une information globale). Ainsi, nous proposons de nous appuyer sur une décomposition des documents par une structure de type fichier inverse composée de deux éléments : le vocabulaire et les occurrences des mots. Pour chaque mot, la liste des positions de ses occurrences dans les documents est sauvegardée. Chaque élément dans la liste indique le document où ce mot est apparu, les positions des occurrences dans le document ainsi que les degrés de confiance correspondants. La figure 5.6 illustre une entrée du fichier pour un mot w_i.

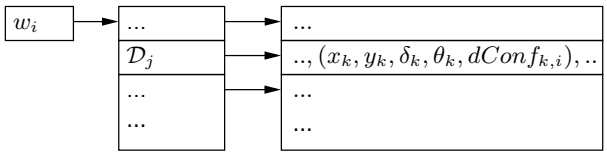


FIG. 5.6 – Structure de l'entrée du mot w_i dans le fichier inverse.

5.3 Localisation de symboles dans les documents graphiques

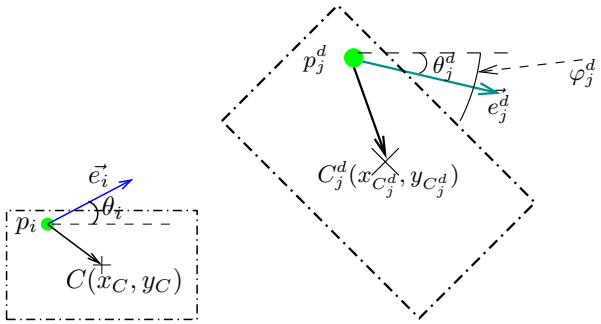
Lors de la localisation de symboles, le symbole requête est traité selon les mêmes étapes que celles effectuées sur les documents dans la base : le CFPI à chaque point d'intérêt du symbole est calculé et mis en correspondance avec les mots visuels construits sur la base de documents. Nous utilisons ensuite le fichier inverse afin de trouver des documents qui contiennent probablement ce symbole. Nous sélectionnons en effet à partir de ce fichier tous les documents qui contiennent les mots visuels apparus dans le symbole requête. Le symbole requête est en général beaucoup plus petit que le document entier. Ainsi, un document peut contenir tous les mots du symbole requête sans pour autant contenir celui-ci. Dans le cadre de cette thèse, nous ne chercherons pas à optimiser le choix de cette liste de documents pouvant hypothétiquement contenir le symbole requête. Nous nous concentrons sur la localisation des symboles dans les documents de cette liste.

Dans cette partie, nous montrons comment localiser les occurrences du symbole requête dans un document particulier. Des régions candidates sont tout d'abord déterminées et couplées avec un processus de vote pour mettre en évidence les régions susceptibles de contenir le symbole requête.

5.3.1 Régions candidates

Nous essayons de localiser dans le document des régions qui contiennent probablement des occurrences du symbole requête. Ces régions sont déterminées en se basant sur les relations entre les points d'intérêt et le rectangle englobant du symbole requête. Le rectangle englobant du symbole $rect = (x_C, y_C, w, h)$ (voir Fig. 5.7(a)) est défini à partir de son centre $C(x_C, y_C)$ et w, h dénotent respectivement sa largeur et sa hauteur.

Supposons que le point $p_i = (x_i, y_i, \delta_i, \theta_i)$ (de la requête) et le point $p_j^d = (x_j^d, y_j^d, \delta_j^d, \theta_j^d)$ (dans le document) sont associés au même mot visuel (c'est-à-dire que ces deux points sont similaires), la région candidate $rect_j^d = (x_{C_j^d}, y_{C_j^d}, w_j^d, h_j^d, \varphi_j^d)$ dans le document est déterminée en fonction de $\{rect, p_i, p_j^d\}$ (cf. Fig. 5.7.b) via des transformations de p_i à p_j^d. Il s'agit d'une translation du repère cartésien de la requête vers le point p_i (l'équation (5.15) donne les nouvelles coordonnées du centre C après la translation), suivie par une rotation vectorielle de la requête par un angle φ_j^d (cf. l'équation (5.13)) pour que l'orientation dominante de p_i ait la même direction que celle de p_j^d, puis un changement d'échelle par un facteur ξ (cf. l'équation (5.14)) et d'une translation du repère cartésien de p_j^d à l'origine $(0, 0)$ du document. Les coordonnées du centre de la région candidate obtenues après ces transformations sont définies par les équations (5.9) et (5.10). Les équations (5.11) et (5.12) définissent la largeur et la longueur de la région. p_i est considéré comme le point de contrôle de la région requête et p_j^d celui de la région à apparier dans le document. Un exemple de ces régions est montré en Fig. 5.8.



(a) requête (b) région dans le document

FIG. 5.7 – Localisation d'un rectangle englobant dans le document correspondant à la requête.

$$x_{C_j^d} = x_j^d + \xi * (x_{piC} * cos(\varphi_j^d) - y_{piC} * sin(\varphi_j^d)) \quad (5.9)$$

$$y_{C_j^d} = y_j^d + \xi * (x_{piC} * sin(\varphi_j^d) + y_{piC} * cos(\varphi_j^d)) \quad (5.10)$$

$$w_j^d = w * \xi \quad (5.11)$$

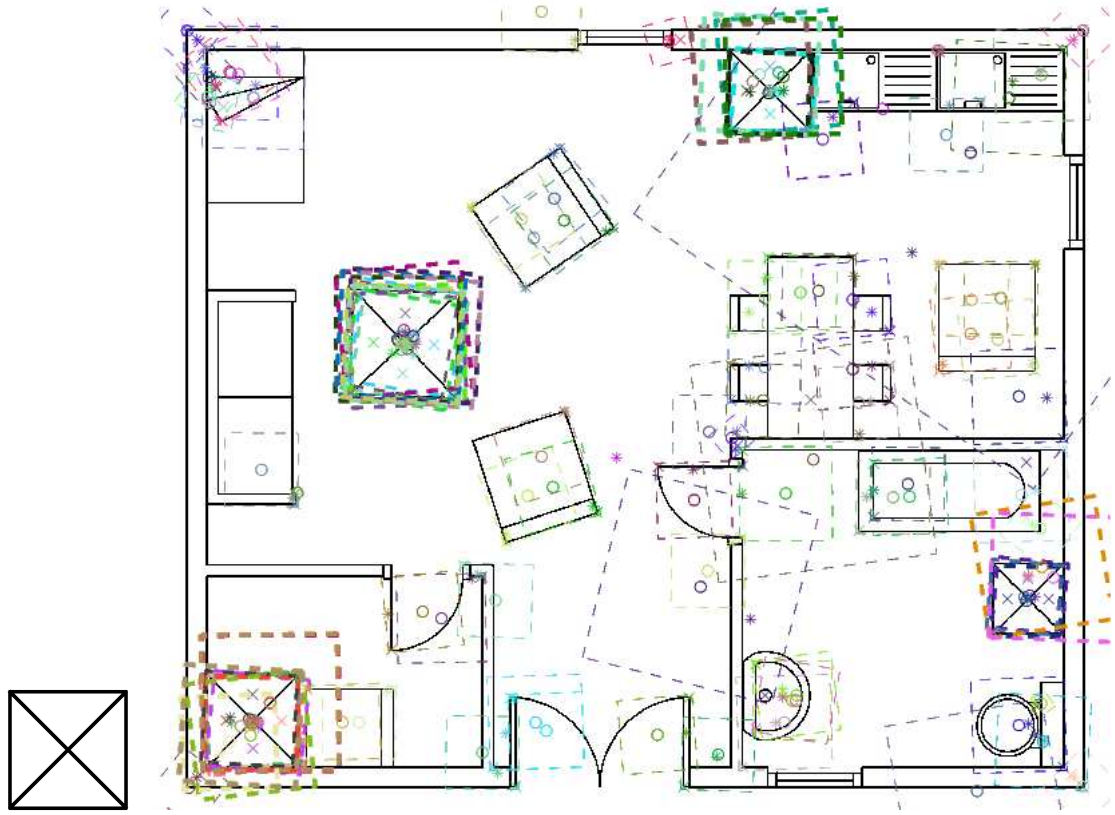
$$h_j^d = h * \xi \quad (5.12)$$

où :

$$\varphi_j^d = \theta_j^d - \theta_i \tag{5.13}$$

$$\xi = \delta_j^d / \delta_i \tag{5.14}$$

$$(x_{piC}, y_{piC}) = (x_C, y_C) - (x_i, y_i) \tag{5.15}$$



(a) (b) Régions candidates

FIG. 5.8 – (a) Symbole requête. (b) Régions candidates correspondantes dans le document. L'épaisseur du trait des rectangles représente la similarité entre la requête et ces régions.

5.3.2 Processus de vote

Une fois les régions candidates définies, un vote sur celles-ci basé sur le modèle vectoriel est effectué afin de ne retenir que celles représentant les occurrences du symbole requête.

Supposons que $W_r = \{w_{r_1}, w_{r_2}, .., w_{r_M}\}$ est l'ensemble des mots associés aux points d'intérêt pour une région candidate r avec les degrés de confiance correspondants $\{dConf_{r_1}, dConf_{r_2}, .., dConf_{r_M}\}$, la fréquence d'apparition du mot i dans cette région est définie par tf_i^r :

$$tf_i^r = \frac{\sum_{w_{r_k} \in W_r, w_{r_k} \equiv i} dConf_{r_k}}{\sum_{k=1}^{M} dConf_{r_k}} \tag{5.16}$$

et sa fréquence pondérée pour cette région :

$$f_i^r = tf_i^r * idf_i \tag{5.17}$$

La région est donc représentée par un vecteur de fréquences des mots pondérées s^r :

$$s^r = (f_1^r, f_2^r, .., f_K^r) \tag{5.18}$$

La distance cosinus entre s^r et s^q (le vecteur de fréquences des mots pondérées de la requête) représente la valeur du vote en faveur de la région r. Les régions obtenant des valeurs de vote élevées sont considérées comme des occurrences du symbole requête dans le document.

idf au niveau du symbole : Dans la formule (5.17), idf_i est un poids pondéré de la fréquence d'apparition du mot visuel i dans la région. L'idf_i indique l'importance du mot i pour la discrimination entre régions. Comme les documents ne sont pas pré-découpés en régions, il n'est pas possible de déterminer l'idf_i directement à partir des documents entiers. De plus, le rôle de l'idf_i dans la formule (5.17) est de fournir une meilleure séparation entre l'occurrence du symbole requête et d'autres régions qui contiennent d'autres symboles. Ainsi, nous proposons de déterminer l'idf_i à partir d'une base de symboles segmentés (isolés) prédéfinie qui sert de base de référence. L'idf_i est donc défini par Eq. (5.19).

$$idf_i = log(N_s/n_{s,i}) \tag{5.19}$$

où N_s est le nombre total de symboles dans cette base, $n_{s,i}$ le nombre de symboles dans lesquels le mot i apparaît.

Renforcement de la similarité entre la requête et une région candidate : Les points d'intérêt étant détectés à plusieurs résolutions, il peut arriver que des points du symbole définis à une résolution donnée ne correspondent à aucun point détecté dans une occurrence de ce symbole. Ce type de problème arrive généralement dans le cas de points détectés à basse résolution pour deux symboles de tailles très différentes. Nous proposons donc, pour rendre la similarité plus précise, de tenir uniquement compte des points d'intérêt qui pourraient être détectés à la fois sur la requête et sur la région considérée. Pour ce faire, nous déterminons un intervalle de résolutions valable pour la région ($[\delta_{min}^R, \delta_{max}^R]$) en fonction des résolutions minimale et maximale de la requête ($\delta_{min}^Q, \delta_{max}^Q$) et ainsi que des résolutions de la paire de points appariés (δ_i, δ_j^d). Il s'agit d'avoir le même nombre de résolutions sur la requête et la région à apparier (Fig. 5.9). Cet intervalle est déterminé par :

$$[\delta_{min}^R, \delta_{max}^R] = [\delta_j^d/(2^{\lfloor log_2(\delta_i/\delta_{min}^Q)\rfloor}), \delta_j^d * 2^{\lfloor log_2(\delta_{max}^Q/\delta_i)\rfloor}]$$

Seuls les mots correspondant aux points détectés dans cet intervalle sont utilisés pour le calcul de la similarité entre la requête et la région candidate.

Filtrage des faux positifs : Notre approche permet de construire un ensemble de mots visuels pour chaque région. Néanmoins, cet ensemble ne fournit pas d'information sur les relations spatiales existantes entre ces mots. Il peut ainsi arriver que des régions ne contenant pas d'occurrences du symbole requête aient une grande valeur de vote, c'est-à-dire qu'une région contienne bien les mêmes mots visuels que ceux du symbole requête mais que ces derniers soient

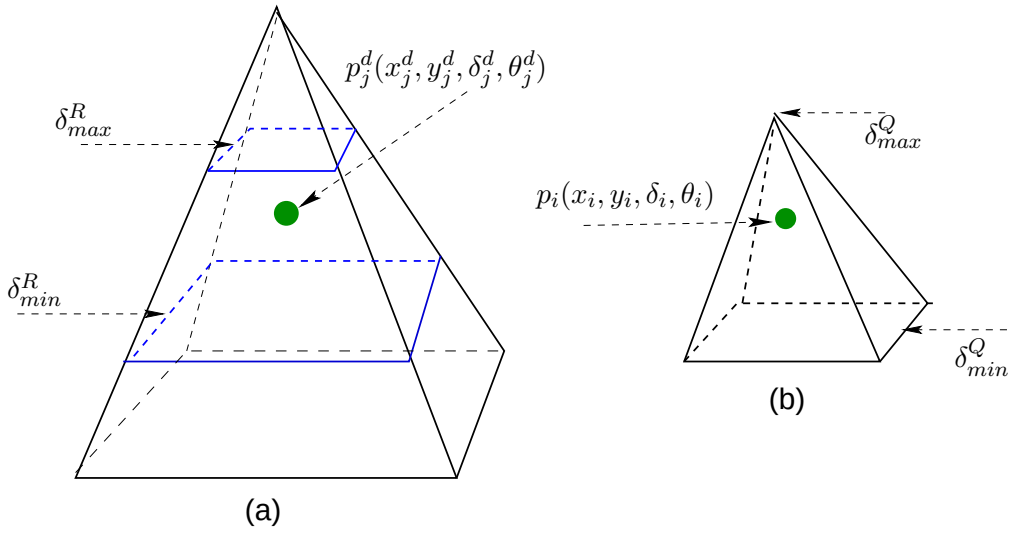
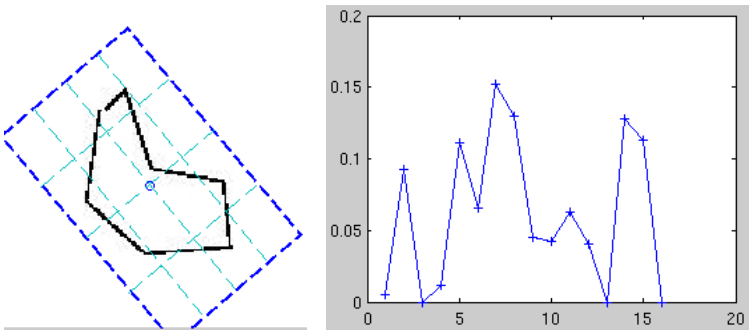


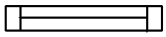
FIG. 5.9 – Intervalle de résolutions valable pour la région par rapport à une requête. (a) Résolutions d'une région dans un document. (b) Résolutions de la requête.

liés de façon très différentes dans la requête et dans la région. Ce problème peut entraîner des erreurs de faux positifs. Afin de les réduire, nous proposons d'effectuer une étape de filtrage sur les régions candidates ayant des grandes valeurs de vote en prenant en compte des informations spatiale sur le symbole. Nous proposons de représenter ces régions par un histogramme (H_r) de distribution des points de contour par rapport au centre de la région. Cet histogramme est calculé sur le même principe que le contexte de forme sauf qu'il est déterminé par rapport au centre de la région et sur un quadrillage. En effet, nous proposons de diviser la région en un quadrillage de taille $M \times M$. Les proportions des points de contour dans ces cases définissent l'histogramme H_r correspondant à la région (Fig. 5.10). La distance entre l'histogramme correspondant à la requête et celui correspondant à la région permet de réduire les faux positifs dans les résultats obtenus. Dans le cadre de notre expérimentation, nous avons utilisé $M = 4$. Dans la Fig. 5.11, nous montrons un exemple du résultat de localisation d'un symbole avant et après le filtrage.

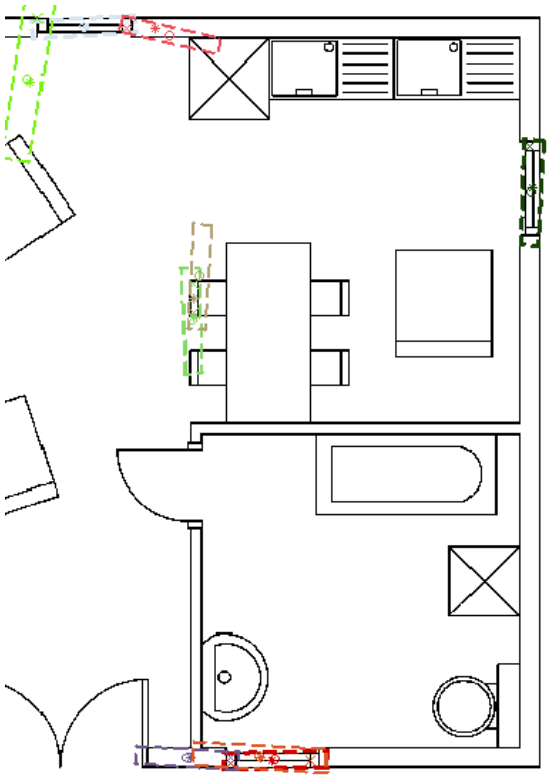


(a) quadrillage (b) H_r

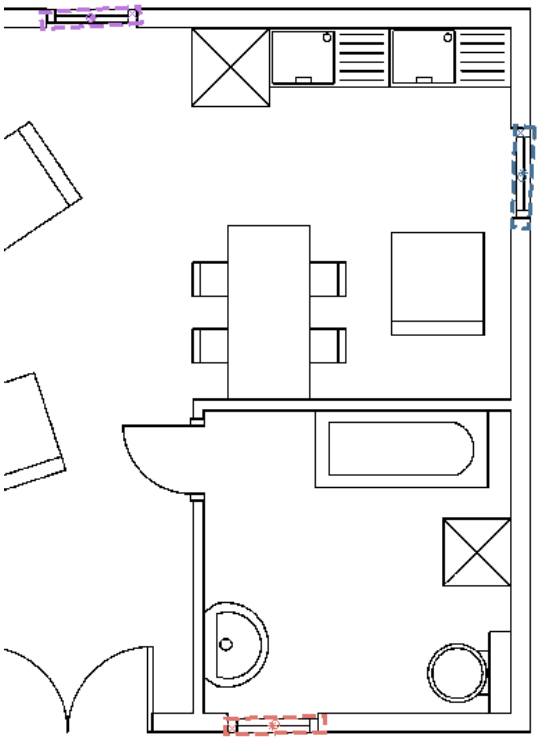
FIG. 5.10 – Relations spatiales entre les points de contour dans une région candidate (histogramme H_r).



(a) requête



(b) avant le filtrage



(c) après le filtrage

FIG. 5.11 – Résultat de la localisation d'un symbole requête avant et après le filtrage.

5.4 Résultats expérimentaux

Dans nos expérimentations, nous avons utilisé le *k-means* pour construire des mots visuels à partir d'un base de documents. Il n'est pas possible de savoir exactement à combien de classes (K) les CFPIs calculés sur les documents sont bien distribués. Pour cela, nous avons effectué le *k-means* avec différentes valeurs de K ($K = \overline{140, 180}$) et repris la valeur de K maximisant un index de validité[17] défini dans les méthodes de clustering [Maulik 02]. Cette procédure est répétée quelques fois et nous gardons la classification qui nous donne de bons résultats a posteriori. Nous avons utilisé $K = 170$ pour tous les tests dans cette thèse. Remarquons que le choix de l'intervalle de K est arbitraire et que les techniques de clustering sont généralement loin d'être parfaites surtout pour les données réelles. Donc, il y a toujours de l'ambiguïté dans la classification obtenue. C'est l'une des raisons pour lesquelles nous proposons d'utiliser l'appariement multiple lors de mise en correspondance des CFPIs aux mots visuels.

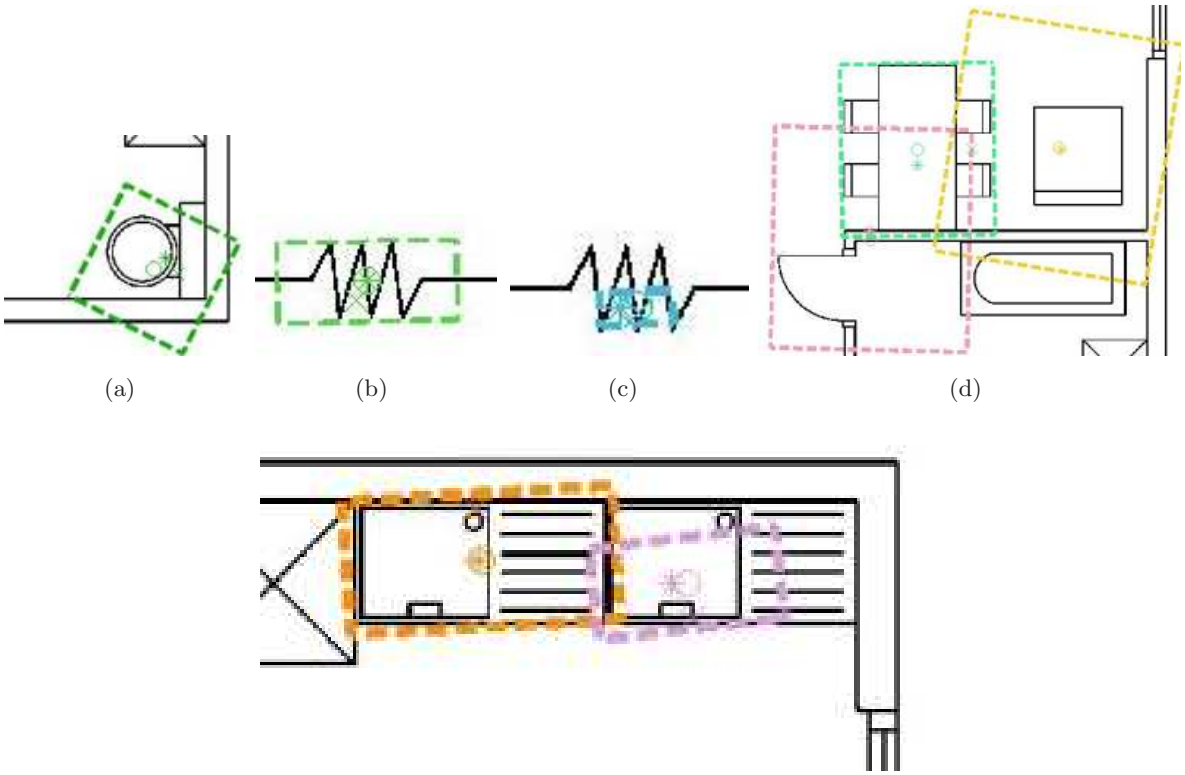
5.4.1 Mesures de performance

Afin d'évaluer un système de localisation de symboles, nous avons besoin d'une vérité terrain et de mesures de performance. À notre connaissance, les seules bases de vérité terrain disponibles pour le problème de localisation de symboles sont les bases du projet SYSED [Delalandre 09]. Dans ces bases, chaque symbole est intégré dans une région rectangulaire. Cette région peut contenir des zones de fond n'appartenant pas au symbole.

Concernant les mesures de performance, la localisation de symboles étant un cas particulier de la recherche d'information, il est possible d'utiliser les mesures de ce domaine (précision, rappel, *F-score*, etc.) pour évaluer un système de localisation de symboles. Dans [Rusinol 09a], les auteurs ont redéfini ces mesures pour les adapter au contexte de la localisation de symboles. Ils proposent ainsi de calculer ces mesures en se basant sur la surface de recouvrement et de non-recouvrement entre la région définie par la vérité terrain et celle détectée par le système. Malheureusement, ces mesures sont peu adaptées aux bases du projet SYSED. En effet, ces mesures sont calculées précisément au niveau du pixel, or la vérité terrain définie dans le projet SYSED peut contenir une partie du fond. Par conséquent, le rappel et la précision mesurés ne seront pratiquement jamais égaux à 1, même dans le cas d'une très bonne localisation.

Dans notre contexte, nous avons choisi d'utiliser la précision et le rappel au niveau des régions pour mesurer la capacité du système. C'est-à-dire qu'une région est dite soit pertinente soit non-pertinente. La précision P est définie comme le rapport entre le nombre de régions pertinentes détectées et le nombre de régions détectées. Le rappel R est défini comme le rapport entre le nombre de régions pertinentes récupérées et le nombre de symboles pertinents existants dans la base. Ainsi, dans les évaluations actuelles, nous calculons manuellement les mesures de performance. Pour ce faire, toutes les régions détectées qui couvrent parfaitement ou presque le symbole sont considérées comme des détections correctes (régions pertinentes), les autres comme de fausses détections. La Fig. 5.12 donne quelques exemples de détections correctes et incorrectes pour des requêtes tirées de la Fig. 5.13.

[17] Traduit de "validity index"



(a) (b) (c) (d)

(e)

FIG. 5.12 – Exemples de détections considérées comme correctes et incorrectes : (a) détection correcte pour la requête 5.13(i), (b) détection correcte pour la requête 5.13(k), (c) détection incorrecte pour la requête 5.13(k), (d) une détection correcte (rectangle vert) et deux détections incorrectes (rectangles jaune et rose clair) pour la requête 5.13(e), (e) une détection correcte (jaune) et une détection incorrecte (violet).

5.4.2 Résultats de la localisation

Pour évaluer notre approche de localisation de symboles dans les documents graphiques, nous avons effectué des tests sur des documents synthétiques du projet SESYD.

86 documents graphiques (50 documents de schémas architecturaux et 36 documents de schémas électriques) ont été utilisé comme base de documents. Nous avons utilisé la technique de classification des *k-means* à partir des descripteurs calculés sur cette base pour construire les mots visuels. Nous avons ensuite testé différentes requêtes pour évaluer notre approche. Nous rappelons que lors d'une requête, les CFPIs de la requête sont extraits et mis en correspondance avec l'ensemble des mots visuels construits. Les similarités entre les régions candidates et la requête sont ensuite calculées grâce au modèle vectoriel à partir des mots visuels et un filtrage pour vérifier l'adaptation entre les relations spatiales de la région et celle de la requête. Les régions dont la similarité est élevée sont considérées comme des occurrences potentielles du symbole requête.

Notre première expérimentation a consisté à comparer notre approche avec d'autres approches. Bien qu'il existe de nombreux travaux sur le problème de la localisation de symboles, il n'existe pas d'évaluation complète pour laquelle une même base seraient employée par toutes les approches existantes. Nous proposons dans le cadre de cette étude comparative de comparer les résultats de notre approche avec ceux obtenus par l'approche de Locteau qui travaille avec un graphe d'adjacence inexacte [Locteau 08]. Nous avons choisi de comparer notre approche avec celle-là car nous utilisons la même base de test et ses résultats sont disponibles. Dans le Tab. 5.1, nous présentons les résultats obtenus par notre approche (colonnes *(a)*) et les résultats obtenus par Locteau (colonnes *(b)*) pour 5 documents du projet SESYD[18]. Nous pouvons remarquer que, à part pour les deux premiers symboles, les résultats sont assez similaires avec de bons scores de précision (98.21%) et de rappel (82.1%). Pour le premier symbole, notre approche donne un meilleur résultat que celle de Locteau mais notre système ne répond pas bien au deuxième symbole. Nous reviendrons sur ce problème dans les paragraphes suivants.

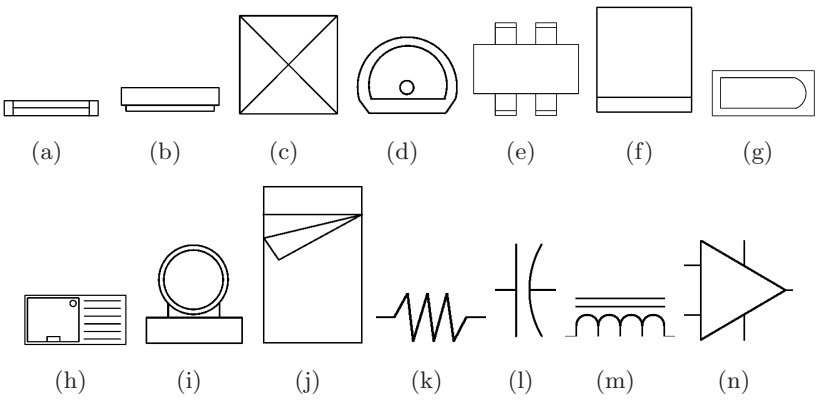


FIG. 5.13 – Requêtes.

[18]les images peuvent être récupérées sur le site : http ://mathieu.delalandre.free.fr/projects/sesyd/index.html

	Fig. 5.13(a)		Fig. 5.13(b)		Fig. 5.13(c)		Fig. 5.13(d)		Fig. 5.13(e)		Fig. 5.13(f)		Fig. 5.13(g)	
	(a)	*(b)*	*(a)*	*(b)*	*(a)*	*(b)*	*(a)*	*(b)*	*(a)*	*(b)*	*(a)*	*(b)*	*(a)*	*(b)*
D-0	0/0/0	0	0/0/3	2	4/4/4	4	1/1/1	1	1/1/1	1	3/3/4	2	1/1/1	1
D-1	3/3/3	0	0/0/0	0	4/4/4	4	1/1/1	1	1/1/1	1	3/3/4	2	1/1/1	1
D-2	1/1/1	0	0/0/2	1	4/4/4	4	0/0/0	0	1/1/1	1	3/3/4	2	1/2/1	0
D-3	1/1/1	0	0/0/2	1	4/4/4	4	1/1/1	1	1/1/1	1	2/3/3	1	1/1/1	0
D-4	2/2/2	0	0/0/1	1	4/4/4	4	0/0/0	0	1/1/1	1	3/3/4	2	1/1/1	1
Précison	7/7	-	-	5/5	20/20	20/20	3/3	3/3	5/5	5/5	14/15	9/9	5/6	3/3
Rappel	7/7	0/7	0/8	5/8	20/20	20/20	3/3	3/3	5/5	5/5	14/19	9/19	5/5	3/5
Moyenne	*(a)* : Précision = 55/56 = 98.21%, Rappel = 55/67 = 82.1%													
	(b) : Précision = 48/48= 100%, Rappel = 48/67= 71.64%													

TAB. 5.1 – Résultats de la localisation de symboles dans des documents graphiques obtenus *(a)* avec notre approche et *(b)* avec l'approche proposée par [Locteau 08]. La première colonne contient les documents, la première ligne contient les symboles requête. Les valeurs $x/y/z$ dans *(a)* désignent : le nombre de symboles localisés correctement (x), le nombre total de symboles récupérés (y), le nombre d'occurrences du symbole requête existant dans le document. Les valeurs dans *(b)* représentent le nombre des symboles localisés correctement dans chaque document.

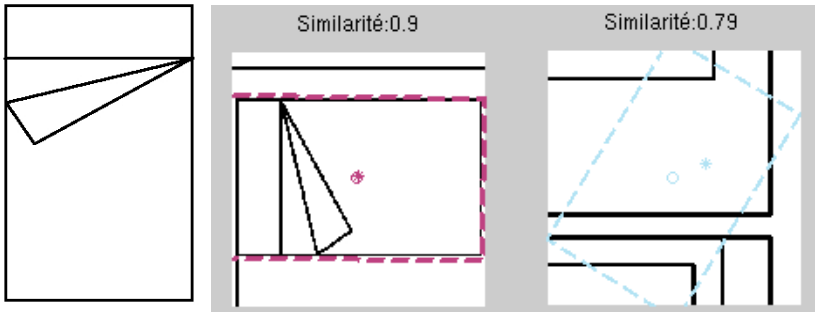
Req.	$N^{\circ}VT$	$N^{\circ}DC$	$N^{\circ}errP$	$N^{\circ}errN$	P	R
Fig. 5.13(a)	82	82	0	0	82/82	82/82
Fig. 5.13(b)	68	0	2	68	0/2	0/68
Fig. 5.13(c)	195	192	1	3	192/193	192/195
Fig. 5.13(d)	25	25	0	0	25/25	25/25
Fig. 5.13(e)	50	50	0	0	50/50	50/50
Fig. 5.13(f)	189	158	9	31	158/167	178/189
Fig. 5.13(g)	50	50	23	0	50/73	50/50
Fig. 5.13(h)	112	112	0	0	112/112	112/112
Fig. 5.13(i)	50	50	2	0	50/52	50/50
Fig. 5.13(j)	50	50	12	0	50/62	50/50
Fig. 5.13(k)	83	83	0	0	83/83	83/83
Fig. 5.13(l)	38	38	0	0	38/38	38/38
Fig. 5.13(m)	13	13	0	0	13/13	13/13
Fig. 5.13(n)	36	36	0	0	36/36	36/36
Moyen	Précision = 95%, Rappel = 90%					

TAB. 5.2 – Résultats de la localisation des symboles dans Fig. 5.13. Req. : requêtes. $N^{\circ}VT$: nombre d'occurrences de la requête dans le document. $N^{\circ}DC$: nombre de détections correctes. $N^{\circ}errP$: nombre de faux positifs. $N^{\circ}errN$: nombres de faux négatifs. P : précision, R : rappel.

Afin de fournir une évaluation plus complète, nous avons testé notre approche de localisation pour d'autres symboles requêtes sur les documents de la base. Le Tab. 5.2 présente les résultats obtenus pour les requêtes de la Fig. 5.13. La première colonne du tableau présente les requêtes. La deuxième ($N^{\circ}VT$) correspondent au nombre d'occurrences du symbole existant dans le document (vérité terrain). La troisième ($N^{\circ}DC$) et la quatrième ($N^{\circ}errP$) contiennent respectivement le nombre de détections correctes et le nombre de fausses détections (faux positifs). La dernière colonne ($N^{\circ}errN = N^{\circ}VT - N^{\circ}DC$) indique le nombre d'occurrences de la requête dans le document qui n'ont pas été trouvées (faux négatifs). Nous avons complété ce tableau par les taux moyens de précision ($P = N^{\circ}DC/(N^{\circ}DC + N^{\circ}errP)$) et de rappel ($R = N^{\circ}DC/N^{\circ}VT$) pour chaque requête. Les faux positifs ont une influence sur la précision du résultat et les faux négatifs sur la capacité à fournir toutes les réponses possibles. Ainsi, moins il y a de faux positifs, plus la précision est élevée et moins il y a de faux négatifs, plus le rappel est élevé.

Nous trouvons que les régions candidates sont bien déterminées. Cela permet de repérer les régions qui pourraient avoir une occurrence du symbole requête (Fig. 5.8). Les occurrences sont bien localisées en termes d'orientation ainsi que d'échelle même lorsqu'elles sont connectées avec d'autres éléments dans le document. Nous présentons en Fig. 5.16 quelques exemples de résultats de localisation de symboles dans des documents graphiques. Nous pouvons remarquer dans le Tab. 5.2 que la localisation de plusieurs symboles peut fournir des résultats parfaits avec des valeurs de précision et rappel très élevées (jusqu'à 100% pour la localisation de symboles dans 5.13(a), 5.13(h), 5.13(e), 5.13(k), 5.13(l), 5.13(n)). Il est possible d'observer pour certains cas des erreurs de faux positifs, mais si nous regardons l'ordre de similarité, les régions pertinentes sont classées avant les faux positifs (Fig. 5.14). L'existence des faux positifs est causée par la valeur

de seuil de similarité. Il est en effet difficile de définir manuellement un tel seuil qui fonctionne bien sur tous les symboles. À notre avis, une possibilité est d'apprendre un seuil pour chaque symbole en s'appuyant sur un ensemble des occurrences en contexte.



(a) (b)

FIG. 5.14 – Deux occurrences du symbole (a) sont détectées dans un document dont un faux positif avec un degré de similarité plus petit que la région correcte.

Les expérimentations effectuées (Tab. 5.1, 5.2) montrent que notre approche permet de bien localiser les symboles dans les documents graphiques. Néanmoins, on peut remarquer que certains types de symboles sont mal localisés (erreur de faux négatifs) tels que les symboles présentés en Fig. 5.13(b), 5.13(f) et 5.13(g). Ces problèmes de localisation s'expliquent par ce fait que ces symboles possèdent peu de points d'intérêt discriminants et représentatifs. Ces symboles sont ainsi plus sensibles aux connexions avec d'autres éléments externes et donc plus difficile à localiser. Dans le cas des symboles Fig. 5.15(a) et 5.15(b), la plupart des points d'intérêt se trouvent près des connexions externes (Fig. 5.15(c), 5.15(d)). Les informations représentées par ces points sont moins robustes car elles contiennent beaucoup d'informations externes. Donc, la similarité entre le symbole requête et les régions contenant ces symboles n'est pas assez forte. Afin d'améliorer la localisation de ce type de symboles, une approche consisterait à réduire la valeur des seuils utilisés pour récupérer la région et appliquer un traitement plus robuste. Cependant, appliquer une telle approche dans le cas général peut entraîner une détérioration des résultats. Il est donc important de réduire ces seuils uniquement pour ce type de symboles (une analyse automatique du symbole peut être effectuée pour déterminer son type).

Une autre remarque sur les résultats obtenus est que l'orientation des occurrences détectées n'est pas toujours parfaitement estimée. Une méthode pour améliorer ce point pourrait être de corriger la position des régions par une technique de *"template matching"*. Nous reviendrons sur ces points dans la description des perspectives liées à ce travail de thèse.

Comme il n'est pas évident de savoir ou estimer exactement le nombre de mots visuels correspondant à l'ensemble des données calculées, nous avons fixé la taille du vocabulaire et utilisé l'appariement multiple. L'appariement multiple permet non seulement de réduire les impacts des points se trouvant aux frontières des classes mais aussi atténuer la dépendance de la taille du vocabulaire. Le tableau 5.3 montre l'efficacité de cet appariement par rapport à l'appariement simple lors de la localisation de symboles. Cette expérimentation est effectuée avec les mêmes requêtes et les mêmes documents que celle dont le résultat est montré dans le tableau 5.2.

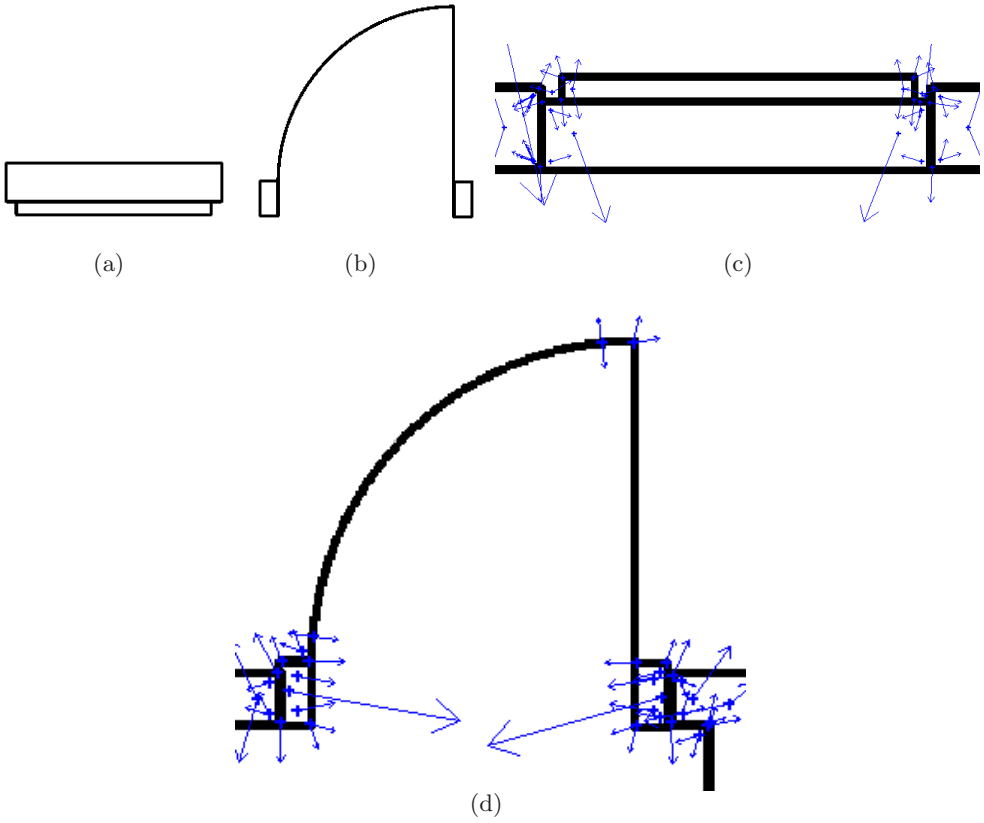


FIG. 5.15 – Symboles manquants de points d'intérêt discriminants, la plupart des points d'intérêt se trouvent près de connexions externes. (a) (b) les modèles isolés, (c) (d) les occurrences de ces modèles dans les documents avec des points d'intérêt détectés.

(a) (b)

(c) (d)

FIG. 5.16 – Réponses de notre approche pour la localisation des requêtes des figures 5.13(c), 5.13(a), 5.13(h) et 5.13(m)

Différentes valeurs de la taille du vocabulaire sont mises en place. Nous constatons que quelle que soit la taille, l'utilisation de l'appariement multiple nous permet d'augmenter significativement le taux de rappel sans trop diminuer le taux de précision. Cela veut dire que nous pouvons obtenir plus de régions pertinentes dans les documents mais avec peu de faux positifs.

Taille du vocabulaire	Appariement	Précision	Rappel
136	*Multiple*	0.96	0.78
	Simple	1	0.62
175	*Multiple*	0.95	0.90
	Simple	0.99	0.66
190	*Multiple*	0.98	0.81
	Simple	1	0.69

TAB. 5.3 – Effet de l'appariement multiple vs l'appariement simple.

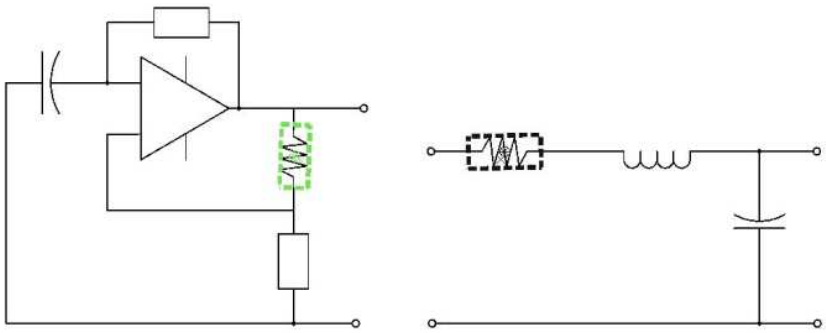
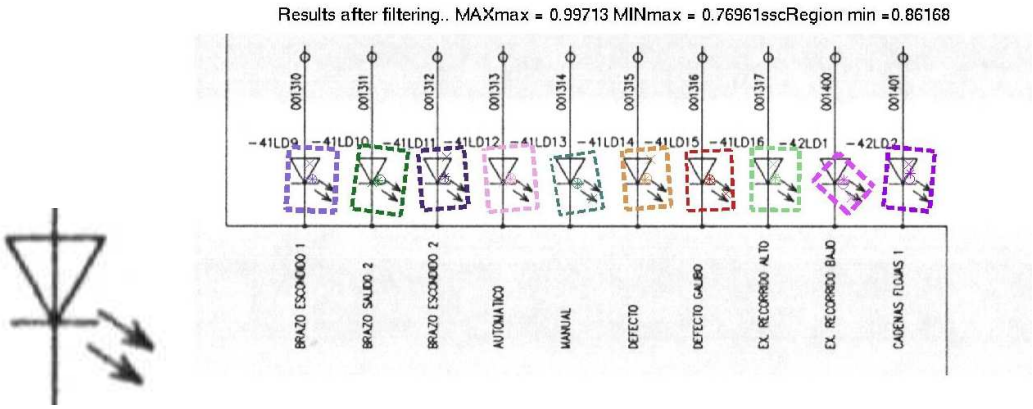


FIG. 5.17 – Résultat obtenu par notre approche pour localiser le symbole Fig. 5.13(k) dans un document.

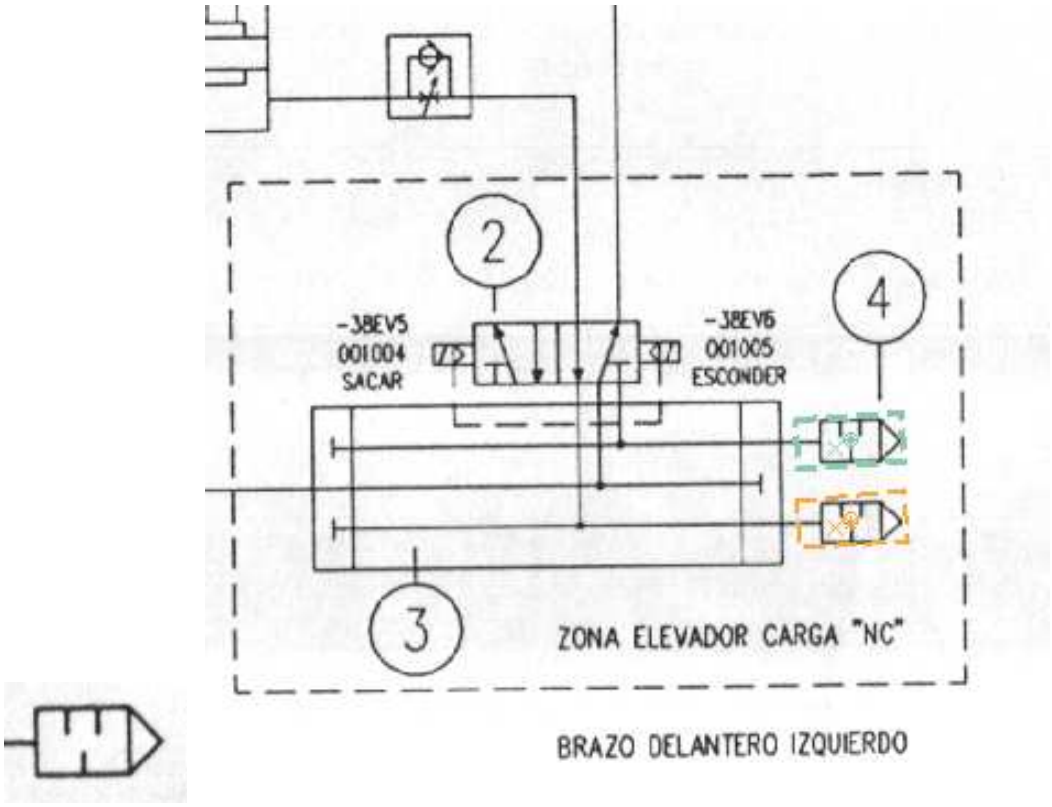
5.5 Conclusion

Nous avons proposé dans ce chapitre une approche de localisation de symboles dans les documents graphiques. Notre approche est basée sur la notion de *"mots visuels"* qui permet d'appliquer sur les documents visuels (images) des techniques issues du domaine du traitement des documents textuels.

Pour représenter le contenu d'un document graphique, nous utilisons les descripteurs CFPIs calculé sur les régions locales des points d'intérêt détectés par le détecteur SIFT. Cette représentation permet de prendre en considération le document à plusieurs résolutions. Les CFPIs calculés à partir d'une base de documents sont ensuite regroupés en classes par une technique de classification non-supervisée. Un ensemble de mots visuels est alors construit, chaque centroïde de classes représentant un mot. Nous proposons ensuite d'apparier les mots visuels avec les CFPIs à l'aide d'une technique d'appariement multiple. Un CFPI est ainsi représenté par plusieurs mots avec différents degrés de confiance. Cette construction d'un vocabulaire visuel et d'un appariement CFPI/mots visuels nous permet de traiter les documents graphiques comme



(a) requête 1 (b) résultat de la localisation de la requête 1

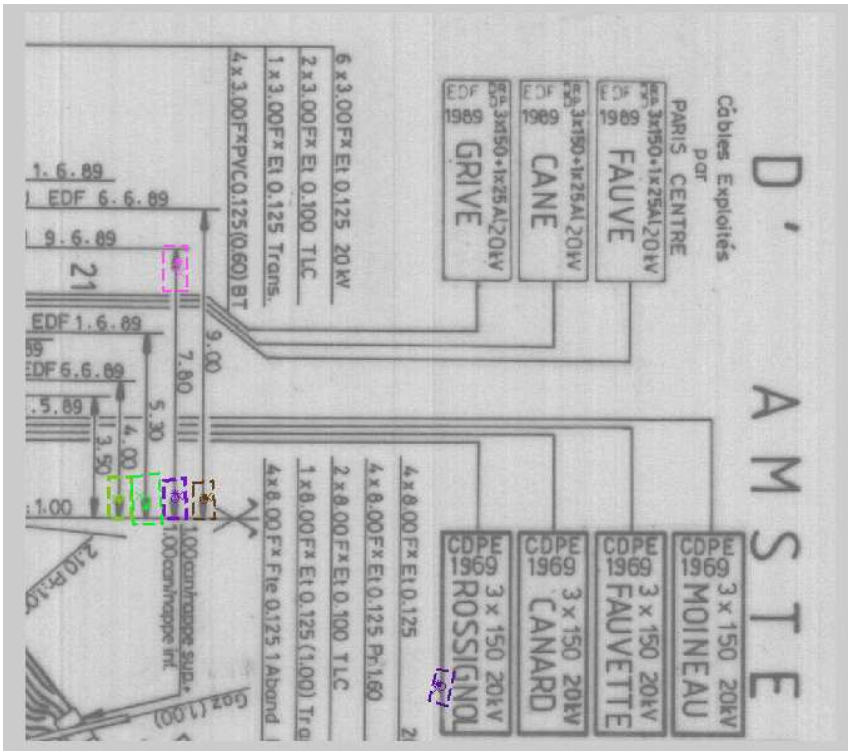


(c) requête 2 (d) résultat de la localisation de la requête 2

FIG. 5.18 – Exemple de localisation de symboles dans une image au niveau de gris. (a) (c) requête. (b)(d) résultats de la localisation.

(a)

(b)



(c)

FIG. 5.19 – Exemple de localisation de flèches dans une image de documents graphiques (c). (a) requête. (b) régions obtenues les plus proches à la requête.

des documents textuels. La localisation de symboles consiste ainsi, dans une première étape, à chercher les documents contenant les mêmes mots visuels que ceux du symbole. Une fois ces documents sélectionnés, des régions candidates correspondant à la requête sont déterminées dans chaque document. Un modèle vectoriel est ensuite adapté et utilisé pour trouver les régions similaires qui contiennent potentiellement des occurrences du symbole.

Bien que notre approche de localisation de symboles ne retourne pas de résultats parfaits, cette dernière offre bien des avantages par rapport aux autres approches existantes dans la littérature. Premièrement, notre approche s'abstrait de toute étape de squelettisation ou de vectorisation, étapes qui entraînent une forte sensibilité aux bruits et aux petites déformations. Deuxièmement, notre approche n'implique aucune hypothèse ou contrainte sur les symboles traités telle que la convexité, la connexité ou la fermeture des composantes du symbole. Notre approche peut fonctionner avec des symboles non-connectés (Fig. 5.13(l), 5.13(m)) ou des symboles ouverts (Fig. 5.13(k), 5.13(m), voir Fig. 5.17). En fin, les erreurs de binarisation n'ont pas un impact fort sur l'approche. Un exemple de localisation sur des images réelles en niveaux de gris est montré en Fig. 5.18. La Fig. 5.19 montre un autre exemple de localisation de flèches dans un document graphique. Nous pouvons constater de bonnes détections sur les réponse les plus proches. Malgré des erreurs de faux positifs et de faux négatifs, elles pourrait être atténuées par une analyse poussée du symbole et de documents réels ou par agrégation de plusieurs descripteurs [Grabisch 95, Terrades 06].

Chapitre 6

Conclusion Générale

Nous avons présenté dans cette thèse nos contributions pour le problème de la localisation de symboles dans les documents graphiques. Le problème de localisation est abordé d'un point de vue différent de celui de la plupart des méthodes existantes dans la littérature. Dans le contexte de l'analyse de documents graphiques, et plus spécifiquement pour le problème de la localisation de symboles, presque toutes les études se focalisent sur l'aspect structurel du document, ce qui nécessite de résoudre plusieurs autres problèmes difficiles tels qu'en amont la vectorisation et en aval la détection d'isomorphisme de (sous-) graphes. Cette thèse tente de voir ce problème de localisation sous un autre point de vue qui est très rarement abordé dans les travaux précédents, l'aspect pixelaires. Ainsi, dans nos travaux, nous avons abordé deux points essentiels pour résoudre ce problème. Le premier concerne le choix d'une représentation des informations des images de documents. Le second est lié au processus de localisation de ces symboles. Notre manuscrit est ainsi organisé pour justifier nos choix relatifs à ces deux notions.

Afin de faciliter le traitement des documents graphiques, nous avons cherché en premier lieu un descripteur qui s'adapte bien aux symboles graphiques. Ainsi, dans le chapitre 2, nous avons proposé un état de l'art sur les descripteurs de formes en considérant deux catégories : les descripteurs structurels et les descripteurs statistiques. En se basant sur les primitives de formes, les descripteurs structurels offrent la possibilité d'appariement partiel. Cet atout doit être nuancé au regard de leur instabilité en présence de dégradations et le coût algorithmique associé à la comparaison de deux objets. En effet, l'usage de descripteurs structurels doit être envisagé en aval d'une segmentation des formes en primitives tandis que le calcul de similarité est classiquement basé sur un appariement entre graphes ou une analyse syntaxique de chaînes. Cela peut expliquer les raisons pour lesquelles les descripteurs statistiques sont souvent utilisés et étudiés en reconnaissance de formes. Parmi les descripteurs existants, le descripteur GFD et les moments de Zernike sont considérés comme les descripteurs les plus performants. Cependant, étant des descripteurs globaux, les taux de reconnaissance obtenus par ces descripteurs chutent en présence d'occlusions partielles. Nous nous sommes donc intéressés au *contexte de forme* dont les performances restent stables pour ce type de dégradation. Dans sa formulation initiale, son usage est néanmoins contraint par sa complexité d'appariement du fait de nombreux vecteurs utilisés pour la représentation d'une forme. Ainsi, dans l'optique de représenter le contenu des documents graphiques, nous avons proposé une adaptation du *contexte de forme*. Deux critères

antagonistes apparaissent : la maîtrise de la complexité lors du calcul de similarité *vs* la tolérance aux occlusions. Aussi, nous avons proposé une définition du contexte de forme aux seuls points d'intérêt. La description d'un objet est ainsi compacte en préservant les informations discriminantes. Cette contribution permet de faciliter le traitement de symboles de manière générale, tant en reconnaissance de formes qu'en localisation.

Concernant la localisation de symboles, un panorama des méthodes de localisation est décrit dans le chapitre 3. Nous les classons également en deux groupes. Le premier se base sur la représentation structurelle des images de documents (*approches structurelles*). Les documents sont tout d'abord segmentés en primitives (telles que les lignes, les régions) et les occurrences du symbole requête sont retrouvées grâce à la recherche exacte ou non d'isomorphisme de (sous-)graphes ou à un processus de regroupement des primitives selon des heuristiques. Le second groupe vise à résoudre le problème de la localisation directement à partir des informations photométriques de l'image en évitant toutes étapes intermédiaires (*approches pixelaires*). Comme les descripteurs structurels, les approches structurelles dépendent fortement de l'étape de segmentation et/ou de regroupement des primitives ainsi que de la qualité de l'appariement des (sous-)graphes. De plus, elles nécessitent souvent des hypothèses sur les symboles traités. Ainsi, avec l'objectif d'être moins dépendant des étapes intermédiaires et des hypothèses, nous proposons dans le chapitre 5 une méthode de localisation qui appartient au groupe des approches pixelaires. Le contenu du document est représenté par extension du descripteur de formes proposé dans le chapitre 4. Les régions candidates dans les documents sont déterminées en fonction de l'appariement local entre la requête et les documents. Pour réduire la complexité du calcul de similarité entre les régions candidates et la requête, nous avons adapté et intégré une technique bien établie dans le domaine de la recherche d'informations textuelles en utilisant la notion de *mots visuels*. Les résultats obtenus sur les documents synthétiques soulignent la pertinence de notre approche. Bien que notre approche de localisation ne donne pas de résultats parfaits mais elle offre des avantages par rapport aux méthodes existantes. Ni étapes préalables telles que la squelettisation, la vectorisation sont nécessaires. Elle n'impose aucune contrainte ou hypothèse sur les symboles considérés telle que la connexité, le convexité ou la fermeture. Cela nous permet de récupérer des symboles que d'autres approches ignorent. Elle offre également la possibilité d'adaptation à d'autres types de localisation car notre approche est peu tributaire à la segmentation et pas liée à des contraintes particulières.

Dans les chapitres 4 et 5, nous avons analysé les avantages et les inconvénients de nos propositions. Pour conclure ce manuscrit, nous dressons une liste de perspectives envisageables afin de mettre en valeur les travaux réalisés dans le cadre de cette thèse.

Les premiers tests effectués ont conduit à de bons résultats en localisation de symboles sur des documents réels. Il conviendrait d'analyser plus finement le comportement de notre système lorsque la qualité des documents varie. Cela nous permettrait de cerner les éléments susceptibles d'être améliorés pour un cas d'usage donné. Par ailleurs, nous n'avons pas eu l'opportunité de tester à grande échelle notre problème sur des documents réels, mais de premiers résultats nous laissent présager une bonne capacité d'adaptation.

Dans nos expérimentations, les seuils de similarité, utilisés pour décider si la région contient ou non une occurrence de la requête, sont fixés à l'avance. Puisque nous avons utilisé la dis-

tance cosinus dont la valeur est comprise entre 0 et 1, l'estimation des seuils a été simplifiée. Cependant, il serait bénéfique que ces valeurs soient déterminées automatiquement. Une piste envisageable est de les estimer par apprentissage (par exemple la méthode de régression logistique [Bishop 06]) en s'appuyant sur une base de symboles disponible ou grâce à un processus de retour de pertinence. Apprendre automatiquement des seuils pour chaque type de symbole permettrait peut-être d'améliorer la localisation de certains symboles (voir chapitre 5). Une autre possibilité pour augmenter la performance pour ces symboles serait de modifier le détecteur de points d'intérêt, notamment afin d'obtenir une description pertinente des courbes pour lesquelles trop peu de points stables sont identifiés.

Pour quelques cas, essentiellement en localisation de symboles sur des documents réels, l'estimation de l'orientation de la région peut ne pas être satisfaisante. Pour y remédier, une perspective consisterait à employer une technique de *template matching* au moment de l'appariement d'une paire de régions requête-document.

Bien que la localisation dans chaque document soit très bonne, la recherche des documents pertinents n'est pas optimale. Ceci est dû à la différence de taille, tant au niveau de l'image que de sa description, entre l'élément présenté en requête et les documents indexés. En effet, la liste des documents contenant tous les mots visuels existant dans la requête peut être très longue. Elle contient probablement plusieurs documents non-pertinents. Nous souhaitons, à plus long-terme, améliorer l'indexation d'une base de documents. Pour ce faire, il est possible de compléter le fichier inverse par des informations sur les voisinages de chaque mot visuel. Autrement dit, nous souhaitons construire un fichier inverse dont chaque entrée est un groupe de mots et non plus un mot visuel isolé.

Une autre perspective intéressante à long terme de ces travaux en particulier, et des méthodes de localisation en générale est d'utiliser les résultats de localisation de symboles pour re-indexer les documents dans la base de façon sémantique, au niveau des symboles. Cependant, une telle tâche ne peut être conduite qu'avec de nombreuses précautions dans la mesure où la qualité des résultats de cette indexation sémantique découlerait de ceux obtenus en localisation. La mise en place d'un retour de pertinence y jouerait ici un rôle primordial.

Bibliographie

[Adam 01] S. Adam, J.-M. Ogier, C. Cariou, R. Mullot, J. Gardes & Y. Lecourtier. *Utilisation de la transformée de Fourier-Mellin pour la reconnaissance de formes multi-orientées et multi-échelles : application à l'analyse automatique de documents techniques.* Traitement du signal, vol. 18, pages 17–33, 2001.

[Agarwal 04] S. Agarwal, A. Awan & D. Roth. *Learning to detect objects in images via a sparse, part-based representation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 11, pages 1475–1490, November 2004.

[Baeza-Yates 99] R. Baeza-Yates & B. Ribeiro-Neto. Modern information retrieval. ACM Press / Addison-Wesley, New York, 1999.

[Barrat 08] S. Barrat & S. Tabbone. *Visual features with semantic combination using Bayesian network for a more effective image retrieval.* In International Conference on Pattern Recognition, Tampa, Florida, USA, December 2008.

[Bartolini 05] I. Bartolini, P. Ciaccia & M. Patella. *WARP : Accurate retrieval of shapes using phase of fourier descriptors and time warping distance.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, pages 142–147, January 2005.

[Baumann 95] S. Baumann. *A simplified attributed graph grammar for high-level music recognition.* In The Third International Conference on Document Analysis and Recognition, volume 2, page 1080, Washington, DC, USA, 1995. IEEE Computer Society.

[Bay 08] H. Bay, , T. Tuytelaars & Luc Van Gool. *SURF : Speeded Up Robust Features.* Computer Vision and Image Understanding (CVIU), vol. 110, no. 3, pages 346–359, 2008.

[Belkasim 91] S.O. Belkasim, M. Shridar & M. Ahmadi. *Pattern recognition with moment invariants : a comparative study and new results.* Pattern Recognition, vol. 24, pages 1117–1138, 1991.

[Belkasim 07] S. Belkasim, E. Hassan & T. Obeidi. *Explicit invariance of Cartesian Zernike moments.* Pattern Recognition Letters, vol. 28, no. 15, pages 1969–1980, November 2007.

[Belongie 02] S. Belongie, J. Malik & J. Puzicha. *Shape Matching and Object Recognition Using Shape Contexts.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pages 509–522, Avril 2002.

[Bernier 03] T. Bernier & J.-A. Landry. *A new method for representing and matching shapes of natural objects.* Pattern Recognition, vol. 36, no. 8, pages 1711–1723, August 2003.

[Bishop 06] Christopher M. Bishop. Pattern recognition and machine learning. Springer-Verlag New York Inc., 2006.

[Blum 67] H. Blum. *A Transformation for Extracting New Descriptors of Shape.* In Weiant Wathen-Dunn, editeur, Models for the Perception of Speech and Visual Form, pages 362–380. MIT Press, 1967.

[Bober 01] M. Bober. *MPEG-7 visual shape descriptors.* IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 6, pages 716–719, 2001.

[Bosch 06] A. Bosch, A. Zisserman & X. Munoz. *Scene Classification via pLSA.* In Computer Vision - ECCV 2006, volume 3954/2006, pages 517–530. Springer Berlin / Heidelberg, May 2006.

[Brady 84] M. Brady & H. Asada. *Smoothed Local Symmetries and Their Implementation.* Rapport technique, Massachusetts Institute of Technology, Cambridge, MA, USA, 1984.

[Bribiesca 99] E. Bribiesca. *A new chain code.* Pattern Recognition, vol. 32, no. 2, pages 235–251, 1999.

[Bunke 82] H. Bunke. *Attributed Programmed Graph Grammars and Their Application to Schematic Diagram Interpretation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, pages 574–573, 1982.

[Chong 03] C.-W. Chong, P. Raveendran & R. Mukundan. *Translation invariants of Zernike moments.* Pattern Recognition, vol. 36, no. 8, pages 1765–1773, August 2003.

[Conte 04] D. Conte, P. Foggia, C. Sansone & M. Vento. *Thirty years of graph matching in pattern recognition.* International Journal of Pattern Recognition and Artificial Intelligence, vol. 18, no. 3, pages 265–298, 2004.

[Cordella 00] L.P. Cordella, P. Foggia, C. Sansone & M. Vento. *Fast graph matching for detecting CAD image components.* In International Conference on Pattern Recognition, 2000.

[Cordella 01] L. P. Cordella, P. Foggia, C. Sansone & M. Vento. *An improved algorithm for matching large graphs.* In 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, pages 149–159, 2001.

[Couäsnon 96] B. Couäsnon & J. Camillerapp. *Segmentation et reconnaissance de documents guidées par la connaissance a priori : application aux partitions musicales.* PhD thesis, Université de Rennes 1, 1996.

[Courant 53] R. Courant & D. Hilbert. Methods of mathematical physics, volume 1. New York : Interscience, 1953.

[Crowley 81] J.-L. Crowley. *A representation for visual information.* PhD thesis, Carnegie Mellon University, 1981.

[Crowley 84] J.-L. Crowley & A.-C. Parker. *A representation for shape based on peaks and ridges in the difference of low pass transform.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, no. 2, pages 156–169, March 1984.

[Csurka 04] G. Csurka, C. R. Dance, L. Fan, J. Willamowski & C. Bray. *Visual categorization with bags of keypoints.* In ECCV International Workshop on Statistical Learning in Computer Vision, 2004.

[Dalal 05] N. Dalal & B. Triggs. *Histograms of Oriented Gradients for Human Detection.* In Computer Vision and Pattern Recognition, volume 1, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[Delalandre 09] M. Delalandre, E. Valveny, T. Pridmore & D. Karatzas. *Generation of Synthetic Documents for Performance Evaluation of Symbol Recognition & Spotting Systems.* International Journal on Document Analysis and Recognition (IJDAR), 2009.

[Dosch 00] P. Dosch. *Un environnement pour la reconstruction 3D d'édifices à partir de plans d'architecte.* PhD thesis, Université Henry Poincaré - Nancy 1, 2000.

[Dosch 04] P. Dosch & J. Lladós. *Vectorial Signatures for Symbol Discrimination.* In Graphics Recognition. Springer Berlin / Heidelberg, 2004.

[Escalera 09] S. Escalera, A. Fornés, O. Pujol, A. Escudero & P. Radeva. *Circular Blurred Shape Model for Symbol Spotting in Documents.* In IEEE International Conference on Image Processing, Novembre 2009.

[Fonseca 05] J.M Fonseca, A. Ferreira & J.A Joaquim. *Content-based retrieval of technical drawings.* International Journal of Computer Applications in Technology, vol. 23, no. 2-3, pages 86–100, March 2005.

[Fränti 00] P. Fränti, A. Mednonogov, V. Kyrki & H. Kälviäinen. *Content-based matching of line-drawing images using the Hough transform.* International Journal on Document Analysis and Recognition, vol. 3, pages 117–124, 2000.

[Freeman 74] H. Freeman. *Computer processing of line-drawing images.* Computing Surveys, vol. 6, no. 1, pages 57–97, March 1974.

[Fu 74] K.-S. Fu. Syntactic methods in pattern recognition. Academic Press, New York, 1974.

[Fu 86] K.-S. Fu. *A step towards unification of syntactic and statistical pattern recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 3, pages 398–404, 1986.

[Gevers 04] T. Gevers & A. W. M. Smeulders. Content-based image retrieval : An overview, chapitre 8. Prentice Hall, 1st edition, July 2004.

[Ghosh 05] A. Ghosh & N. Petkov. *Robustness of Shape Descriptors to Incomplete Contour Representations.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 11, pages 1793–1804, November 2005.

[Gonzalez 02] R. C. Gonzalez & R. E. Woods. Digital image processing (2nd edition). Prentice Hall, 2002.

[Grabisch 95] M. Grabisch. *Fuzzy integral in multicriteria decision making.* Fuzzy Sets and Systems, vol. 69, no. 3, pages 279–298, 1995.

[Granlund 72] G.H. Granlund. *Fourier Preprocessing for hand print character recognition.* IEEE Trans. Computers, vol. C-21, pages 195–201, 1972.

[Grigorescu 03] C. Grigorescu & N. Petkov. *Distance Sets for Shape Filters and Shape Recognition.* IEEE Transactions on Image Processing, vol. 12, no. 10, pages 1274–1286, 2003.

[Hilaire 06] X. Hilaire & K. Tombre. *Robust and accurate vectorization of line drawings.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, pages 890–904, 2006.

[Hörster 08] E. Hörster, R. Lienhart & M. Slaney. *Continuous visual vocabulary models for pLSA-based scene recognition.* In International conference on Content-based image and video retrieval, pages 319–328, New York, NY, USA, 2008. ACM.

[Hu 62] M.-K. Hu. *Visual Pattern Recognition by Moment Invariants.* IRE Transactions on Information Theory, vol. 8, pages 179–187, 1962.

[Hwang 06] S.-K. Hwang & W.-Y. Kim. *Fast and Efficient Method for Computing ART.* IEEE Transactions on Image Processing, vol. 15, no. 1, pages 112–117, January 2006.

[Idris 96] F. Idris & S. Panchanathan. *Algorithms for indexing of compressed images.* In International Conference on Visual Information Systems, pages 303–308, Melbourne, 1996.

[Iivarinen 96] J. Iivarinen & A. Visa. *Shape recognition of irregular objects.* In Intelligent Robots and Computer Vision XV : Algorithms, Techniques, Active Vision, and Materials Handling, Proc. SPIE 2904, pages 25–32, 1996.

[Jouili 09] S. Jouili & S. Tabbone. *Graph Matching based on Node Signatures.* In Workshop on Graph-based Representations in Pattern Recognition, pages 154–163. Springer, Venice, Italy, 2009.

[Jurie 05] F. Jurie & B. Triggs. *Creating efficient codebooks for visual recognition.* In International Conference on Computer Vision, volume 1, pages 604–610, Beijing, China, October 2005.

[Kadir 01] T. Kadir & M. Brady. *Scale, saliency and image description.* International Journal of Computer Vision, vol. 45, no. 2, pages 83–105, 2001.

[Ke 04] Y. Ke & R. Sukthankar. *PCA-SIFT : A More Distinctive Representation for Local Image Descriptors.* Computer Vision and Pattern Recognition, vol. 2, pages 506–513, 2004.

[Kherfi 04] M. L. Kherfi, D. Ziou & A. Bernardi. *Image Retrieval from the World Wide Web : Issues, Techniques, and Systems.* ACM Comput. Surv., vol. 36, no. 1, pages 35–67, 2004.

[Kim 99] W.-Y. Kim & Y.-S. Kim. *A New Region-Based Shape Descriptor.* Rapport technique, SO/IEC MPEG99/M5472, TR 15-01, Maui, Hawaii, 1999.

[Kolesnikov 03] A. Kolesnikov. *Efficient algorithms for vectorization and polygonal approximation*. PhD thesis, University of Joensu, Finlande, 2003.

[Kotoulas 05] L. Kotoulas & I. Andreadis. *Image Analysis Using Moments*. In The 5th International Conference on Technology and Automation, pages 360–364, Greece, October 2005.

[Kotoulas 08] L. Kotoulas & I. Andreadis. *An Efficient Technique for the Computation of ART*. IEEE transactions on circuits and systems for video technology, vol. 18, no. 5, pages 682–686, 2008.

[Larlus 06] D. Larlus, G. Dorkó & F. Jurie. *Création de vocabulaires visuels efficaces pour la catégorisation d'images*. In Reconnaissance des Formes et Intelligence Artificielle, Tours, France, 2006.

[Lazebnik 06] S. Lazebnik, C. Schmid & J. Ponce. *Beyond Bags of Features : Spatial Pyramid Matching for Recognizing Natural Scene Categories*. In Computer Vision and Pattern Recognition, volume 2, pages 2169–2178, 2006.

[Li 92] Y. Li. *Reforming the theory of invariant moments for pattern recognition*. Pattern Recognition, vol. 25, pages 723–730, 1992.

[Li 03] J. Li, Q. Pan, H. Zhang & P. Cui. *Image recognition using Radon transform*. In Intelligent Transportation Systems, pages 741–744, October 2003.

[Li 05] F.-F. Li & P. Perona. *A bayesian hierarchical model for learning natural scene categories*. In Computer Vision and Pattern Recognition, pages 524–531, 2005.

[Liao 96] S. X. Liao & M. Pawlak. *On Image Analysis by Moments*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 3, pages 254–266, 1996.

[Liu 07] Y. Liu, , D. Zhang, G. Lu & W.-Y. Ma. *A survey of content-based image retrieval with high-level semantics*. Pattern Recognition, vol. 40, pages 262–282, 2007.

[Liu 09] R. Liu, Y. Wang, T. Baba & D. Masumoto. *Shape Detection from Line Drawings by Hierarchical Matching*. In Computer Analysis of Images and Patterns, volume 5702/2009, pages 922–929. Springer Berlin / Heidelberg, 2009.

[Lladós 02] J. Lladós, E. Valveny, G. Sánchez & E. Martí. *Symbol Recognition : Current Advances and Perspectives*. In Graphics Recognition Algorithms and Applications, volume Volume 2390/2002, pages 104–128. Springer Berlin / Heidelberg, 2002.

[Locteau 07] H. Locteau, S. Adam, E. Trupin, J. Labiche & P. Heroux. *Symbol Spotting Using Full Visibility Graph Representation*. In Seventh IAPR International Workshop on Graphics Recognition, Curitiba, Brazil, September 2007.

[Locteau 08] H. Locteau. *Contribution à la localisation de symboles dans les documents graphiques*. PhD thesis, Université de Rouen, 2008.

[Long 02] F. Long, H. Zhang & D. D. Feng. Multimedia information retrieval and management - technological fundamentals and applications, chapitre Fundamentals of Content-based Image retrieval. Spinger, 2002.

[Lowe 04] D. G. Lowe. *Distinctive image features from scale-invariant keypoints.* International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, November 2004.

[Lu 97] G. Lu. *Chain code-based shape representation and similarity measure.* In Visual Information Systems, volume 1306/1997, pages 135–150. Springer Berlin / Heidelberg, 1997.

[Lu 99] G. Lu & S. Teng. *A novel image retrieval technique based on vector quantization.* In International Conference on Computational Intelligence for Modelling, Control and Automation, pages 36–41, 1999.

[MacLean 08] W. J. MacLean & J. K. Tsotsos. *Fast pattern recognition using normalized grey-scale correlation in a pyramide image representation.* Machine Vision and Application, vol. 19, no. 3, pages 163–179, May 2008.

[Marcus 92] J. N. Marcus. *A novel algorithm for HMM word spotting performance evaluation and error analysis.* In IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 2, pages 89–92, Los Alamitos, CA, USA, 1992. IEEE Computer Society.

[Maulik 02] U. Maulik & S. Bandyopadhyay. *Performance Evaluation of Some Clustering Algorithms and Validity Indices.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 12, pages 1650–1654, 2002.

[Messmer 95] B. T. Messmer & H. Bunke. *Automatic Learning and Recognition of Graphical Symbols in Engineering Drawings.* In Selected Papers from the First International Workshop on Graphics Recognition, Methods and Applications, pages 123 – 134. Springer-Verlag, London, UK, 1995.

[Mikolajczyk 02] K. Mikolajczyk. *Detection of local features invariant to affine transformations.* PhD thesis, Institut National Polytechnique de Grenoble, France, 2002.

[Mikolajczyk 04] K. Mikolajczyk & C. Schmid. *Scale and Affine Invariant Interest Point Detectors.* International Journal of Computer Vision, vol. 60, no. 1, pages 63–86, 2004.

[Mikolajczyk 05] K. Mikolajczyk & C. Schmid. *A performance evaluation of local descriptors.* IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 27, no. 10, pages 1615–1630, October 2005.

[Mukundan 01] R. Mukundan, S.H. Ong & P.A. Lee. *Image analysis by Tchebichef moments.* IEEE Transactions on Image Processing, vol. 10, no. 9, pages 1357–1364, September 2001.

[Nilsback 06] M-E. Nilsback & A. Zisserman. *A Visual Vocabulary for Flower Classification.* In The IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 1447–1454, 2006.

[Nishida 95] H. Nishida. *Curve description based on directional features and quasi-convexity/concavity.* Pattern Recognition, vol. 28, no. 7, pages 1045–1051, 1995.

[Nishida 02] H. Nishida. *Structural feature indexing for retrieval of partially visible shapes.* Pattern Recognition, vol. 35, no. 1, pages 55–67, January 2002.

[Nowak 06] E. Nowak, F. Jurie & B. Triggs. *Sampling strategies for bag-of-features image classification.* In ECCV, pages 490–503. Springer, 2006.

[Otsu 79] N. Otsu. *A threshold selection method from grey-level histograms.* IEEE Transactions on Systems, Man, and Cybernetics, vol. vol. SMC-9, no. 1, pages 62–66, 1979.

[Park 00] S. H. Park, K. M. Lee & S. U. Lee. *A Line Feature Matching Technique Based on an Eigenvector Approach.* Computer Vision and Image Understanding, vol. 77, no. 3, pages 263–283, March 2000.

[Park 03] B. G. Park, K. M. Lee, S. U. Lee & J. H. Lee. *Recognition of partially occluded objects using probabilistic ARG (attributed relational graph)-based matching.* Computer Vision and Image Understanding, vol. 90, no. 3, pages 217–241, 6 2003.

[Pavlidis 78] T. Pavlidis. *A review of algorithms for shape-analysis.* Computer Graphics and Image Processing, vol. 7, pages 243–258, 1978.

[Persoon 77] E. Persoon & K. S. Fu. *Shape discrimination using Fourier Descriptors.* IEEE Trans. Systems, Man, and Cybernetics, vol. SMC-7, no. 3, pages 170–179, March 1977.

[Pham 09] T.-T. Pham, L. Maisonnasse, P. Mulhem & E. Gaussier. *Modèle de langue visuel pour la reconnaissance de scènes.* In CORIA, 2009.

[Ping 02] Z. Ping, R. Wu & Y. Sheng. *Image description with Chebyshev-Fourier moments.* J. Opt. Soc. Am., vol. A19, pages 1748–1754, 2002.

[Prokop 92] R.J. Prokop & A.P. Reeves. *A survey of moment-based techniques for unoccluded object representation and recognition.* CVGIP : Graphical Models and Image Processing, vol. 54, no. 5, pages 438–460, 1992.

[Qureshi 08] R. J. Qureshi, J.-Y. Ramel, D. Barret & H. Cardot. *Spotting Symbols in Line Drawing Images Using Graph Representations.* In Graphics Recognition. Recent Advances and New Opportunities, pages 91–103. Springer-Verlag, Berlin, Heidelberg, 2008.

[Rafiei 02] D. Rafiei & A. O. Mendelzon. *Efficient retrieval of similar shapes.* The Very Large Data Bases Journal, vol. 11, no. 1, pages 17–27, August 2002.

[Ramos 04] O. Ramos, S. Tabbone, L. Wendling & E. Valveny. *Symbol Recognition based on a Multiresolution Analysis of the Radon Transform.* In International Workshop on Image, Video, and Audio Retrieval and Mining, 2004.

[Rath 03] T. M. Rath & R. Manmatha. *Features for Word Spotting in Historical Manuscripts.* In The Seventh International Conference on Document Analy-

sis and Recognition, page 218, Washington, DC, USA, 2003. IEEE Computer Society.

[Ren 03] H. Ren, Z. Ping, W. Bo, W. Wu & Y. Sheng. *Multidistortion-invariant image recognition with radial harmonic fourier moments*. Journal of the Optical Society of America. A, Optics, image science, and vision, vol. A 20, pages 631–637, 2003.

[Revaud 09] J. Revaud, G. Lavoué & A. Baskurt. *Improving Zernike Moments Comparison for Optimal Similarity and Rotation Angle Retrieval*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 4, pages 627–636, Avril 2009.

[Rosin 97] P. L. Rosin. *Techniques for Assessing Polygonal Approximation of Curves*. IEEE Transactions on Pattern Analysis and Machine Learning, vol. 19, no. 6, pages 659–666, 1997.

[Rosin 03] P. L. Rosin. *Assessing the behaviour of polygonal approximation algorithms*. Pattern Recognition, vol. 36, pages 505–518, 2003.

[Rui 98] Y. Rui, A. She & T.S. Huang. *A modified Fourier descriptor for shape matching in MARS*. In Workshop on Image Databases and Multi Media Search, volume 8, pages 165–180, Amsterdam, Netherlands, 1998.

[Rusinol 06] M. Rusinol & J. Llados. *Symbol Spotting in Technical Drawings Using Vectorial Signatures*. In Graphics Recognition. Ten Years Review and Future Perspectives, volume 3926/2006, pages 35–46. Springer Berlin / Heidelberg, October 2006.

[Rusinol 07] M. Rusinol & J. Llados. *A Region-Based Hashing Approach for Symbol Spotting in Technical Documents*. In Seventh IAPR International Workshop on Graphics Recognition, Curitiba, Brazil, September 2007.

[Rusinol 08] M. Rusinol & J. Lladós. *Word and Symbol Spotting Using Spatial Organization of Local Descriptors*. In The Eighth IAPR International Workshop on Document Analysis Systems (DAS'08), Nara, Japan, September 2008.

[Rusinol 09a] M. Rusinol & J. Lladós. *A Performance Evaluation Protocol for Symbol Spotting Systems in Terms of Recognition and Location Indices*. International Journal on Document Analysis and Recognition, vol. 12, no. 2, pages 83–96, july 2009.

[Rusinol 09b] M. Rusinol, J. Lladós & G. Sánchez. *Symbol Spotting in Vectorized Technical Drawings Through a Lookup Table of Region Strings*. Pattern Analysis and Applications, 2009.

[Schmid 00] C. Schmid, R. Mohr & C. Bauckhage. *Evaluation of interest point detectors*. International Journal of Computer Vision, vol. 37, no. 2, pages 151–172, 2000.

[Sebastian 01] T. B. Sebastian, P. N. Klein & B. B. Kimia. *Recognition of Shapes by Editing Shock Graphs*. In IEEE International Conference on Computer Vision, pages 755–762, 2001.

[Shapiro 01] L. Shapiro & G. Stockman. Computer vision, chapitre 8. Content-Based Image Retrieval, pages 249–273. Prentice Hall, 2001.

[Shaw 97] W. M. Jr Shaw, R. Burgin & P. Howell. *Performance standards and evaluations in IR test collections : Cluster-based retrieval models*. Information Processing & Management, vol. 33, no. 1, pages 1–14, January 1997.

[Shen 94] L. Shen, R.M. Rangayyan & J.E.L. Desautels. *Application of shape analysis to mammographic calcifications*. IEEE Transactions on Medical Imaging, vol. 13, no. 2, pages 263–274, June 1994.

[Shen 99] D. Shen & H. H.S. Ip. *Discriminative wavelet shape descriptors for recognition of 2-D patterns*. Pattern Recognition, vol. 32, no. 2, pages 151–165, 1999.

[Sheng 94] Y. Sheng & L. Shen. *Orthogonal Fourier-Mellin moments for invariant pattern recognition*. Journal of the Optical Society of America. A, Optics and image science, vol. 11, pages 1748–1757, 1994.

[Sidère 08] N. Sidère, P. Héroux & J.-Y. Ramel. *Représentation vectorielle pour l'indexation d'informations structurelles*. In Actes du Colloque International Francophone sur l'Ecrit et le Document, pages 19–24, 2008.

[Sivic 03] J. Sivic & A. Zisserman. *Video Google : A Text Retrieval Approach to Object Matching in Videos*. In ICCV '03 : Proceedings of the Ninth IEEE International Conference on Computer Vision, Washington, DC, USA, 2003. IEEE Computer Society.

[Sivic 06] J. Sivic & A. Zisserman. *Video Google : Efficient Visual Search of Videos*. In Toward Category-Level Object Recognition, volume 4170/2006, pages 127–144. Springer Berlin / Heidelberg, 2006.

[Smith 98] J. R. Smith. *Image Retrieval Evaluation*. In IEEE Workshop on Content-based Access to Image and Video Databases, page 112, Bombay, India, June 1998. IEEE Computer Society Washington, DC, USA.

[Sonka 99] M. Sonka, V. Hlavac & R. Boyle. Image processing, analysis, and machine vision (2rd edition). Pacific Grove (CA) : PWS Pub., 1999.

[Squire 00] D. McG. Squire, W. Müllera, H. Müllera & T. Puna. *Content-based query of image databases : inspirations from text retrieval*. Pattern Recognition Letters, Selected Papers from The 11th Scandinavian Conference on Image, vol. 21, no. 13–14, pages 1193–1198, December 2000.

[Tabbone 05a] S. Tabbone, L. Alonso & D. Ziou. *Behavior of the Laplacian of Gaussian Extrema*. Journal of Mathematical Imaging and Vision, vol. 23, no. 1, pages 107–128, July 2005.

[Tabbone 05b] S. A. Tabbone. *Quelques contributions à la reconnaissance de formes dans des documents graphiques*. Habilitation à diriger des recherches, Université Nancy 2, 2005.

[Tabbone 06a] S. Tabbone, T.-O. Nguyen & G. Masini. *Une méthode de binarisation hiérarchique floue*. In Colloque International Francophone sur l'Ecrit et le Document - CIFED'06, Fribourg, Suisse, September 2006.

[Tabbone 06b] S. Tabbone, L. Wendling & J.-P. Salmon. *A new shape descriptor defined on the Radon transform*. Computer Vision and Image Understanding, vol. 102, no. 1, pages 42–51, Avril 2006.

[Tabbone 07] S. Tabbone & D. Zuwala. *An indexing method for graphical documents*. In International Conference on Document Analysis and Recognition, volume 2, pages 789–793, Curitiba, Brazil, 2007.

[Tanaka 95] E. Tanaka. *Theoretical aspects of syntactic pattern recognition*. Pattern Recognition, vol. 28, no. 7, pages 1053–1061, 1995.

[Teague 80] M. R. Teague. *Image analysis via the general theory of moments*. Journal of the Optical Society of America, vol. 70, pages 920–930, 1980.

[Teh 88] C.-H. Teh & R.T. Chin. *On image analysis by the methods of moments*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 4, no. 4, pages 496 – 513, July 1988.

[Terrades 06] O. Ramos Terrades. *Linear Combination of multiresolution descriptors : Application to Graphics Recognition*. PhD thesis, Universitat Autònoma de Barcelona - Université de Nancy 2, 2006.

[Terrades 07] O. R. Terrades, S. Tabbone & E. Valveny. *A Review of Shape Descriptors for Document Analysis*. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), 2007.

[Tirilly 09] P. Tirilly, V. Claveau & P. Gros. *A review of weighting schemes for bag of visual words image retrieval*. Publication interne, Avril 2009.

[Tuytelaars 08] T. Tuytelaars & K. Mikolajczyk. *Local invariant feature detectors : a survey*. Foundations and Trends® in Computer Graphics and Vision, vol. 3, no. 3, pages 177–280, 2008.

[Ullmann 76] J. R. Ullmann. *An Algorithm for Subgraph Isomorphism*. Journal of ACM, vol. 23, no. 1, pages 31–42, 1976.

[Valveny 07] E. Valveny, P. Dosch, A. Winstanley, Y. Zhou, S. Yang, L. Yan, L. Wenyin, D. Elliman, M. Delalandre, E. Trupin, S. Adam & J.-M. Ogier. *A general framework for the evaluation of symbol recognition methods*. International journal on document analysis and recognition, vol. 9, no. 1, pages 59–74, 2007.

[van Gemert 09] J. C. van Gemert, C. J. Veenman, A. W.M. Smeulders & J.-M. Geusebroek. *Visual Word Ambiguity*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 99, no. 1, 2009.

[van Rijbergen 79] C. J. van Rijbergen. Information retrieval. Butterworths, 1979.

[Vernon 91] D. Vernon. Machine vision : Automated visual inspection and robot vision. Practice Hall International (UK), 1991.

[Wenyin 07] L. Wenyin, W. Zhang & L. Yan. *An interactive example-driven approach to graphics recognition in engineering drawings*. International Journal of Document Analysis and Recognition, vol. 9, no. 1, pages 13–29, March 2007.

[Xia 07] T. Xia, H. Zhu, H. Shu, P. Haigron & L. Luo. *Image description with generalized pseudo-Zernike moments*. Journal of the Optical Society of America. A, Optics, image science, and vision, vol. A 24, pages 50–59, 2007.

[Xiaogang 04] X. Xiaogang, S. Zhengxing, P. Binbin, J. Xiangyu & L. Wenyin. *An online composite graphics recognition approach based on matching of spatial relation graphs*. International Journal on Document Analysis and Recognition, vol. 7, no. 1, pages 44–55, March 2004.

[Yang 05] S. Yang. *Symbol recognition via statistical integration of pixel-level constraint histograms : a new descriptor*. IEEE Transactions on PAMI, vol. 27, no. 2, pages 278–281, 2005.

[Yap 03] P.-T. Yap, R. Paramesran & S.-H. Ong. *Image Analysis by Krawtchouk Moments*. IEEE Transactions on Image Processing, vol. 12, no. 11, pages 1367–1377, November 2003.

[Zahn 72] C.T. Zahn & R.Z. Roskies. *Fourier descriptors for plane closed curves*. IEEE Transaction on Computers, vol. C-21, pages 269–281, March 1972.

[Zhang 02a] D. Zhang & G. Lu. *A Comparative Study of Fourier Descriptors for Shape Representation and Retrieval*. In Asian Conference on Computer Vision, pages 646–651, Melbourne, Australia, January 2002.

[Zhang 02b] D. Zhang & G. Lu. *Generic Fourier descriptor for shape-based image retrieval*. In IEEE International Conference on Multimedia and Expo, pages 425–428, 2002.

[Zhang 02c] D. Zhang & G. Lu. *Shape-based image retrieval using generic Fourier descriptor*. Signal Processing : Image Communication, vol. 17, pages 825–848, 2002.

[Zhang 03] J. Zhang, X. Zhang, H. Krim & G.G. Walter. *Object representation and recognition in shape spaces*. Pattern Recognition, vol. 36, pages 1143–1154, 2003.

[Zhang 04] D. Zhang & G. Lu. *Review of shape representation and description techniques*. Pattern Recognition, vol. 37, no. 1, pages 1–19, January 2004.

[Zhang 07] W. Zhang & W. Liu. *A New Vectorial Signature for Quick Symbol Indexing, Filtering and Recognition*. In The Ninth International Conference on Document Analysis and Recognition, pages 536–540, Washington, DC, USA, 2007. IEEE Computer Society.

[Zhou 05] J. Zhou, H. Shu, H. Zhu, C. Toumoulin & L. Luo. *Image Analysis by Discrete Orthogonal Hahn Moments*. In Image Analysis and Recognition, volume Volume 3656/2005, pages 524–531. Springer Berlin / Heidelberg, October 2005.

[Zhu 00] L. Zhu, A. Zhang, A. Rao & R. Srihari. *Keyblock : an approach for content-based image retrieval*. In The eighth ACM international conference on Multimedia, pages 157–166, New York, NY, USA, 2000. ACM.

[Zhu 07a] H. Zhu, H. Shu, T. Xia, L. Luo & J. L. Coatrieux. *Translation and scale invariants of Tchebichef moments*. Pattern Recognition, vol. 40, no. 9, pages 2530–2542, Septembre 2007.

[Zhu 07b] H. Zhu, H. Shu, J. Zhou, L. Luo & J.-L. Coatrieux. *Image analysis by discrete orthogonal dual Hahn moments*. Pattern Recognition Letters, vol. 28, no. 13, pages 1688–1704, Octobre 2007.

[Zuwala 06a] D. Zuwala. *Reconnaissance de symboles sans connaissance a priori*. PhD thesis, Institut National Polytechnique de Lorraine, 2006.

[Zuwala 06b] D. Zuwala & S. Tabbone. *Une méthode de localisation et de reconnaissance de symboles sans connaissance a priori*. In Colloque International Francophone sur l'Ecrit et le Document - CIFED'06, pages 127–131, Fribourg, Suisse, 2006.