

Système de reconnaissance structurelle de symboles, basé sur une multi représentation en graphes de régions, et exploitant une représentation XML des données

Mathieu Delalandre¹ – Eric Trupin¹ – Jean-Marc Ogier²

¹ Laboratoire PSI, Université de Rouen
76 821 Mont Saint Aignan, France

² Laboratoire L3I, Université de La Rochelle
17042 La Rochelle, France

¹{prénom, nom}@univ-rouen.fr – ²jmogier@univ-lr.fr

Résumé : Dans cet article nous présentons un système de reconnaissance de symboles pour les documents graphiques. Ce système exploite une combinaison d'approches structurelles pour la multi représentation en graphes de régions des symboles. Il permet l'extraction des composantes et des occlusions des symboles, ainsi que leurs relations d'inclusion et de voisinage. Il est ainsi possible d'extraire différents types de graphe selon le problème de reconnaissance envisagé. Ce système exploite une représentation XML des données, ce qui en permet une manipulation aisée. Nous présentons quelques résultats sur un jeu de symboles du concours de reconnaissance GREC2003.

Mots-clés : reconnaissance de symboles, graphes de voisinage, graphes d'inclusion, prétraitement d'images, XML.

1 Introduction

Cet article aborde le problème de la reconnaissance de symboles appliquée aux documents graphiques. Nous présentons ici un système exploitant une combinaison d'approches structurelles pour la multi-représentation en graphes de régions des symboles. Nous présentons dans la section (2) le problème général de la reconnaissance structurelle de symboles appliquée aux documents graphiques. Les deux sections suivantes sont dédiées aux deux parties principales de notre système : l'analyse structurelle et le prétraitement d'images dans la section (3), et la classification structurelle dans la section (4). Dans la section (5) nous présentons l'utilisation du langage XML dans le système. Dans la section (6) nous présentons quelques résultats et leur analyse sur un jeu de symboles du concours de reconnaissance GREC2003. Finalement, dans la section (7), nous concluons et donnons quelques perspectives.

2 La reconnaissance structurelle de symboles

Classiquement, un processus de reconnaissance de symboles est décomposé en deux étapes principales [KAS 02] : une étape d'extraction d'informations à partir

des images, et une étape de reconnaissance. Deux catégories d'approches se distinguent dans la littérature : statistique/connexionniste et structurelle/syntaxique¹. Cet article traite plus spécialement de l'approche structurelle. Celle-ci utilise des graphes pour représenter des objets du document. Dans les documents graphiques, de nombreux objets peuvent être représentés de cette manière, et plus particulièrement les symboles [LLA 01]. Ainsi, dans un système de reconnaissance de symboles, l'étape d'extraction d'informations extrait de l'image des graphes représentant les symboles, et l'étape de reconnaissance structurelle les exploite.

L'étape de reconnaissance structurelle est classiquement utilisée pour la reconnaissance des symboles. Cependant elle permet d'autres applications comme : l'apprentissage, l'indexation, la structuration des données, etc. Elle utilise principalement deux grandes familles de méthodes : les méthodes à base d'appariement de graphes [HAN 02], et les méthodes à base de grammaire de graphes [BLO 95]. Les premières appariement les graphes extraits avec des graphes modèles dans le but de trouver les graphes modèles les plus proches. Les secondes appliquent différentes règles afin de transformer les graphes extraits en graphes modèles. Si les graphes extraits correspondent exactement aux graphes modèles, le problème est considéré comme exact. Malheureusement, dans les applications de traitements d'images de documents, les graphes extraits sont souvent bruités. Le problème de reconnaissance est donc un problème de graphes/sous-graphes inexacts.

L'étape d'extraction d'informations à partir des images extrait (ou construit) les graphes correspondant aux symboles. Dans une publication antérieure [DEL 03], nous nommons cette étape "analyse structurelle". L'analyse structurelle fait appel à de nombreuses méthodes. Parmi elles, beaucoup sont basées sur les graphes de régions. Celles-ci exploitent les relations spatiales entre les composantes connexes afin de construire des graphes de régions représentant les

¹ Dans la suite nous emploierons « structurelle » pour « structurelle/syntaxique ».

symboles, afin les structurer pour la future étape de reconnaissance. En conséquence de quoi, ces méthodes analysent les symboles segmentés (non connectés). De plus, elles sont utilisées pour les problèmes de reconnaissance à faible nombre de classe. En effet, les composantes connexes sont des primitives graphiques de « faible sémantique », ce qui permet peu de distinction entre les symboles. Il existe trois catégories d'approche. La première extrait simplement les caractéristiques topologiques² des composantes connexes dans le but de structurer ces composantes connexes lors de l'étape de reconnaissance. L'étape de reconnaissance exploite alors des contraintes de distances entre les composantes connexes, généralement avec un algorithme de parcours de graphes [LOO 01]. Les deux autres approches extraient des relations plus « complètes » entre les composantes connexes : les relations d'inclusion [BAD 02] et de voisinage [BUR 95]. La première est généralement basée sur l'exploitation d'algorithmes de marquage de composantes connexes [BAD 02], et la deuxième sur les diagrammes de Voronoï généralisé [BUR 95]. La FIG. 1 donne un exemple de symbole avec son graphe d'inclusion (a), et le graphe de voisinage de sa partie centrale (b). Enfin, il existe quelques approches hybrides comme par exemple les approches à base de graphes de concavité [BAD 02] et statistico-structurale [DEL 04].

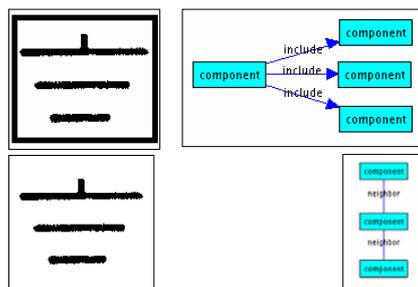


FIG. 1 - (a) relations d'inclusion
(b) relations de voisinage³

Dans cet article nous présentons un système de reconnaissance de symboles exploitant une combinaison d'approches structurales pour la multi-représentation en graphes de régions des symboles. Ce système permet d'extraire de l'image les composantes et leurs occlusions avec leurs relations d'inclusion et de voisinage. Il est ainsi possible de construire différents types de graphe selon le problème de reconnaissance envisagé. Ces graphes sont ensuite exploités par un algorithme d'appariement de graphes. Nous présentons ce système dans les trois sections suivantes.

3 Analyse structurale et prétraitement d'images

Cette section présente la première partie du système pour le traitement d'images. Nous présentons dans la sous-section (3.1) une méthode pour le marquage des

composantes connexes et l'extraction des occlusions. Dans la sous-section (3.2), nous présentons les méthodes utilisées pour construire les graphes de composantes connexes. Finalement, dans la sous-section (3.3), nous présentons une méthode de prétraitement d'images basée sur l'analyse des composantes connexes.

3.1 Marquage de composantes connexes et extraction d'occlusions

La partie centrale des traitements d'images de notre système est une méthode de marquage de composantes connexes. Celle-ci est basée sur un principe d'agrégation des composantes connexes par l'analyse du voisinage 8-connexes des pixels de couleur de la forme. La FIG. 2 illustre ce processus d'agrégation sur deux composantes connexes (de tailles respectives 6 et 4 pixels), avec une image initiale (a), et quatre étapes successives d'agrégation (b). Sur cet exemple, les pixels agrégés sont itérativement effacés. Le marquage de chacune des composantes connexes se fait alors séquentiellement durant ce processus, contrairement aux méthodes classiques [AGG 77] où le marquage est parallèle.



FIG. 2 - (a) image (b) quatre étapes d'agrégation successives

Cette méthode est un peu plus complexe que les méthodes classiques de marquage [AGG 77]. Cependant, son intérêt réside dans le calcul simultané des caractéristiques topologiques² des composantes connexes durant le marquage, ne nécessitant pas de post-traitement supplémentaire. La FIG. 3 donne un exemple d'image (a) avec le résultat de son marquage au format SVG (b) (section 5).

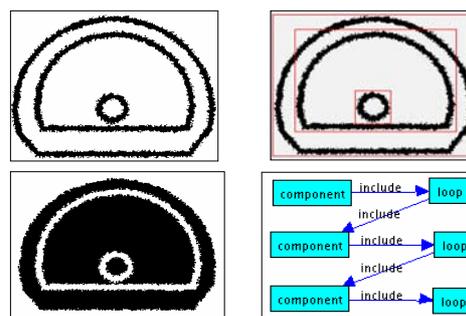


FIG. 3 - (a) symbole (b) marquage
(c) occlusions (d) graphe d'inclusion

Nous utilisons également notre méthode de marquage pour extraire les occlusions (FIG. 3 (c)) des composantes de l'image (FIG. 3 (a)). Dans une première étape nous initialisons les bords de l'image dans la couleur du fond de l'image. L'image est ensuite inversée, et ses composantes connexes marquées. L'initialisation des bords de l'image implique systématiquement que la première composante connexe marquée soit le fond de l'image. Celle-ci est effacée, les autres composantes connexes correspondent alors aux occlusions de l'image (FIG. 3 (c)).

² centres de gravité, rectangles englobants, etc.

³ Nous conservons dans cet article la convention graphique en anglais fidèle à notre représentation XGMML (voir section 5).

3.2 Construction des graphes

Deux relations principales existent entre les composantes connexes (section 2), les relations d'inclusion et de voisinage. Nous présentons dans un premier temps deux méthodes pour extraire ces relations. Nous présentons ensuite une méthode pour construire des graphes hybrides combinant ces deux relations.

Premièrement, nous extrayons les relations d'inclusion entre les composantes et leurs occlusions. La FIG. 3 (d) donne un exemple de graphe d'inclusion de la FIG. 3 (a). Ces relations d'inclusion sont orientées, et permettent de construire une représentation arbre des composantes et occlusions des symboles. Premièrement, notre méthode extrait l'image des occlusions (FIG. 3 (c)). Par la suite, l'image des composantes et l'image des occlusions sont marquées. Finalement, les voisinages des contours des composantes et de leurs occlusions sont analysés afin d'extraire les relations d'inclusion (FIG. 3 (d)).

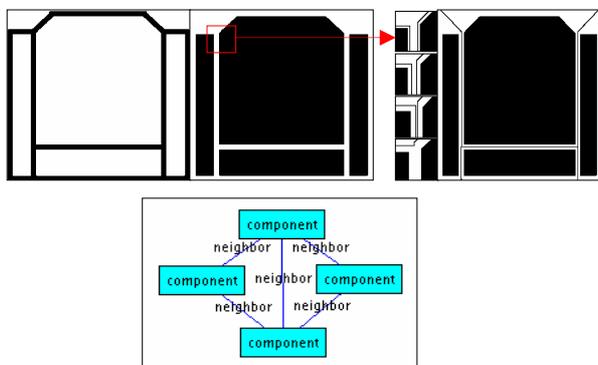


FIG. 4. - (a) symbole (b) extension des contours (c) graphe de voisinage

Ensuite, nous extrayons les relations de voisinage entre les composantes connexes. Ces relations sont non orientées, et permettent de construire une représentation graphe des composantes connexes. Notre méthode est basée sur l'extension des contours des composantes connexes. Durant cette extension, les contours étendus sont marqués progressivement. Le processus d'extension est bloqué lorsque les contours étendus rencontrent des zones déjà marquées (autres contours étendus, composantes connexes), ou des zones nulles (en dehors de l'image). La FIG. 4 (b gauche) donne un exemple de quatre étapes successives d'extension de contours de la FIG. 4 (a). Par la suite la méthode extrait les points frontières de l'image marquée des contours étendus. La FIG. 4 (b droit) donne une représentation image des points frontières extraits de la FIG. 4 (a). Finalement, ces points frontières sont analysés afin d'extraire les relations de voisinage entre les composantes connexes. La FIG. 4 (c) donne le graphe de voisinage de la FIG. 4 (a). Cette méthode semble plus complexe (complexité en θ^2)⁴ que les méthodes basées sur les diagrammes de Voronoi généralisé (complexité en θ) [BUR 95]. Cependant, cette méthode présente des avantages dans le

cas des problèmes nécessitant une forte précision. En effet, elle extrait la carte exacte de voisinage (avec une précision au niveau pixel), et donne ainsi pour certains cas des résultats plus complets que les méthodes basées sur les diagrammes de Voronoi généralisé [BUR 95].

Finalement, nous utilisons une dernière méthode dans le but de construire des graphes hybrides combinant les relations d'inclusion et de voisinage. Le caractère original de cette méthode réside dans l'enrichissement des modes de représentation à base de graphes de régions. La FIG. 5 (a) donne un exemple de graphe hybride de la FIG. 4 (a). Celle-ci combine les relations d'inclusion, et les relations de voisinage entre les occlusions (FIG. 4 (c)). Notre méthode utilise les graphes d'inclusion et de voisinage, avec les caractéristiques topologiques des composantes connexes fournies par la méthode de marquage. Les caractéristiques topologiques permettent la mise en correspondance des noeuds des deux graphes, et ainsi de les « fusionner » en un graphe hybride. Il est possible de construire trois types différents de graphe hybride : celui basé sur les composantes, celui basé sur les occlusions, ou un graphe dual (basé sur les composantes et les occlusions). La FIG. 5 (b), (c), et (d) donne des exemples de ces trois types de graphe hybride de la FIG. 3 (a).

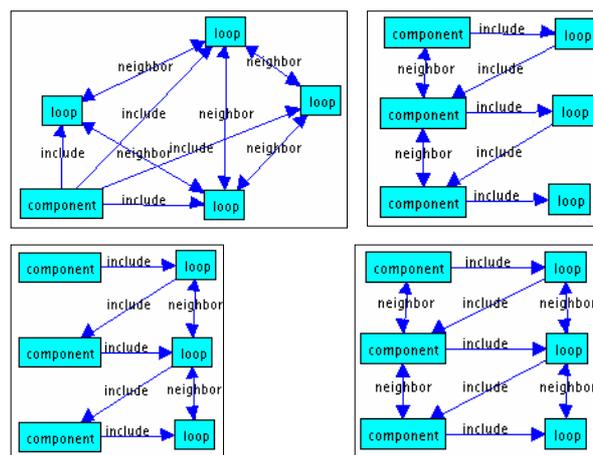


FIG. 5 - (a) graphe hybride (b) composantes (c) occlusions (d) dual

3.3 Prétraitement d'images

Les méthodes de construction de graphes présentées dans la dernière sous-section (3.2) sont basées sur l'extraction des composantes connexes. En conséquence de quoi, elles sont très sensibles au bruit impulsionnel (ajout de petites composantes et occlusions). La FIG. 6 (b) donne un exemple d'application d'un bruit impulsionnel sur la FIG. 6 (a).



FIG. 6 - (a) image modèle (b) bruit impulsionnel (c) image filtrée

⁴ 25.1 secondes pour un plan binaire non compressé de 364 ko « 2.9 Millions de pixels » (CPU 2 GHz, Système Windows).

Pour résoudre ce problème, nous utilisons une méthode de filtrage adaptative basée sur l'analyse des surfaces des composantes connexes (1). Premièrement, un tableau (S) des surfaces (non nulles) des composantes connexes est créé. Celui-ci peut être créé à partir de l'image des composantes, de l'image des occlusions, ou à partir des deux images. De ce tableau des surfaces, un tableau des rapports des surfaces (R) est créé. Le rapport maximum (r) est cherché, son indice permet par la suite de déterminer le seuil de surface pour le filtrage (s). La FIG. 6 (b) donne un exemple d'image filtrée de la FIG. 6 (b).

$$S = \bigcup_{i=1}^n s_i ; R = \bigcup_{i=1}^{n-1} \left(r_i = \frac{s_{i+1}}{s_i} \right) ; \quad (1)$$

$$r_j = \max(R) ; s = s_j .$$

4 Classification structurelle

Les graphes extraits (section 3) sont ensuite exploités par un algorithme de d'appariement de graphes [HER 00]⁵. Cet algorithme permet de calculer des critères de similarité entre graphes, basés sur le recouvrement entre un graphe candidat et un graphe modèle. Ce recouvrement correspond à leur plus grand sous-graphe commun. Cet algorithme permet ainsi de traiter des problèmes de graphes inexacts. Deux taux de similarité peuvent être calculés selon le nombre d'éléments communs sur les nœuds (δn) et sur les arcs (δe). Le taux global de similarité est obtenu par calcul de la variance entre (δn) et (δe). Le résultat de la classification correspond au label du graphe modèle de taux de similarité minimum.

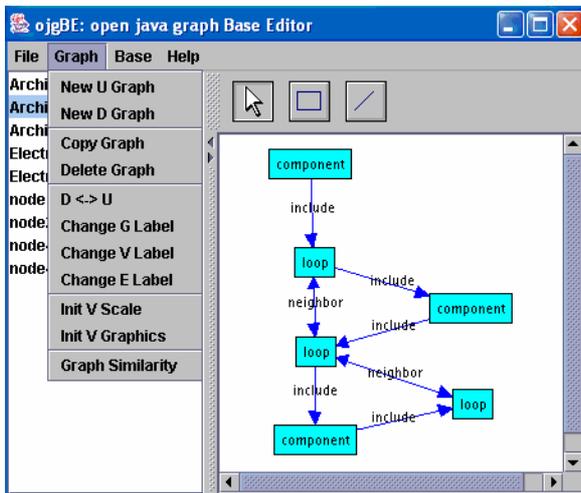


FIG. 7 – open java graph Base Editor

Les graphes modèles sont acquis dans le système via notre interface graphique utilisateur : ojgBE (FIG. 7). ojgBE est une interface Java basée sur l'utilisation de la librairie OpenJGraph⁶. L'utilisation de ojgBE permet d'adapter aisément notre système à de nouvelles

applications de reconnaissance par édition de l'utilisateur des bases de graphes modèles. Il ne s'agit donc ici que d'un apprentissage des graphes modèles idéaux. Cette interface permet d'éditer (FIG. 7 panel droit) des graphes labellisés (arcs et nœuds), orientés et/ou non orientés. Les graphes édités sont inclus dans une base de graphes. L'utilisateur peut par la suite naviguer dans cette base de graphes (FIG. 7 panel gauche) et effectuer différentes actions : étiqueter les graphes, dupliquer les graphes, trier la base, changer l'orientation des graphes, analyser les similarités entre graphes (section 6), etc. La base de graphes est stockée au format XGMML (section 5).

5 Utilisation de XML dans le système

Les différentes parties de notre système (sections 3 et 4) exploitent une représentation XML des données. L'utilisation de ce langage offre plusieurs avantages.

XML est un méta-langage car il est défini comme un langage racine, ce qui permet de définir et d'utiliser des sous-langages spécialisés. L'utilisation de ces langages nous permet de normaliser la représentation de nos données, et d'utiliser ainsi un ensemble d'outils informatiques existants. Nous utilisons SVG⁷ pour la représentation graphique des données superposées à l'image (FIG. 3 (b)), en combinaison avec la plate forme batik⁸. Nous utilisons également XGMML⁹ pour la représentation des graphes utilisés dans les différentes parties de notre système, en combinaison avec la librairie OpenJGraph⁶ via notre interface ojgBE (section 4). La FIG. 8 donne un exemple de code XGMML du graphe d'inclusion du symbole s6 sur la FIG. 9.

```
<?xml version="1.0"?>
<graph directed="1">
  <node id="1" label="component"/>
  <node id="2" label="loop"/>
  <edge label="include"
    source="1" target="2"/>
</graph>
```

FIG. 8 – XGMML

XML permet également l'utilisation de parseur, permettant d'accéder facilement aux données. Ceci permet d'interfacer aisément différentes sources de données dans un système, du moment qu'elles sont représentées en XML. Nous utilisons le parseur Xerces⁸ dans notre système de façon à évaluer les résultats de reconnaissance (section 6). Ceci nous permet de confronter les données vérité terrain des tests avec celles provenant des différentes parties de notre système (sections 3 et 4).

6 Expérimentation, résultats, et analyse

Nous présentons ici quelques résultats obtenus sur un jeu de symboles du concours de reconnaissance GREC2003¹⁰. Le problème de ce concours concerne la

⁵ Nous développons sommairement cet algorithme ici et reportons le lecteur à [HER 00].

⁶ <http://openjgraph.sourceforge.net/>

⁷ Scalable Vector Graphics

⁸ <http://xml.apache.org/>

⁹ eXtensible Graph Markup and Modeling Language

¹⁰ Graphics Recognition 2003: <http://www.cvc.uab.es/grec2003/>

reconnaissance de symboles architecturaux et électriques segmentés (FIG. 9). Différents jeux de tests sont disponibles¹⁰ selon : le nombre de classes de symbole, les dégradations vectorielles et binaires utilisées, etc. Ces jeux de tests sont fournis avec leurs fichiers modèles décrivant la vérité terrain, de façon à évaluer les résultats de reconnaissance. Nous avons testés notre système sur un jeu de test de 9 symboles (FIG. 9). Notre système nous permet d'adapter notre modèle de représentation en fonction du problème de reconnaissance. Nous avons utilisé ici le graphe hybride pour représenter nos symboles basés sur les relations d'inclusion entre composantes et occlusions, et les relations de voisinage entre occlusions (voir sous-section 3.2). En effet, les graphes de voisinage et d'inclusion seuls ne sont pas adaptés pour cette application de reconnaissance. Le graphe d'inclusion est similaire pour les symboles s8 et s9, et le graphe de voisinage des occlusions pour le symbole s1 et s7. Nous avons testé notre système sur 600 images de symboles, avec 6 jeux tests (de 100 images chacun) d'images binaires dégradées selon différents types et niveaux de bruit (degrad-level2-m1, à degrad-level2-m6¹¹). La FIG. 6 (a) donne un exemple de dégradation binaire de la FIG. 6 (b).

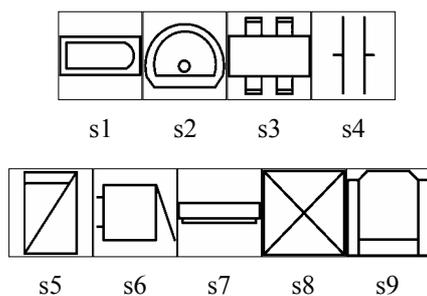


FIG. 9 - jeu de symboles

La FIG. 10 donne les résultats globaux obtenus sur les 6 tests. Nous obtenons des résultats parfaits sur tous les tests, excepté le test n°5. Dans le but d'analyser ces résultats, nous avons mesuré deux types de bruit. Le premier est une estimation du bruit de dilatation (2). Cette estimation (\hat{E}_d) est calculée entre les images tests (it) et les images modèles (im). Elle est basée sur la recherche du nombre de pixels de la forme communs (pnc) entre les images tests et modèles, et le nombre de pixels de la forme de l'image test (pn). Le deuxième est un bruit structurel. Ce bruit structurel donne le taux de graphes bruités parmi les graphes extraits. Ceux-ci sont détectés par comparaison avec leurs graphes modèles. Nous pouvons voir sur la FIG. 10 une corrélation partielle entre le bruit de dilatation et le bruit structurel. En effet, les composantes connexes sont des primitives graphiques robustes (les résultats parfaits le prouvent). Cependant, elles sont très sensibles au phénomène de dilatation, qui provoque des fermetures d'occlusions et des sur-connexions de composantes. Ce bruit de dilatation est la principale source de bruitage des graphes extraits.

$$\hat{E}_d(it, im) = 1 - pnc(it, im) / pn(it) . \quad (2)$$

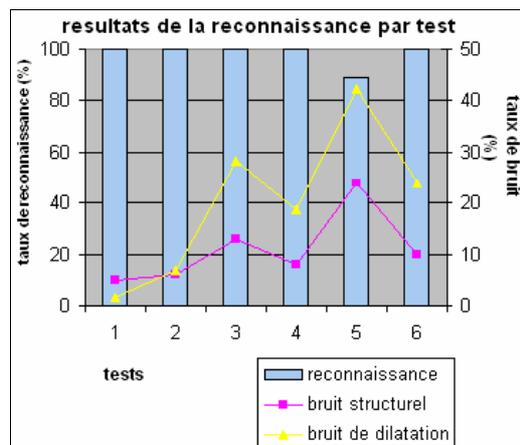


FIG. 10 – résultats de reconnaissance par test

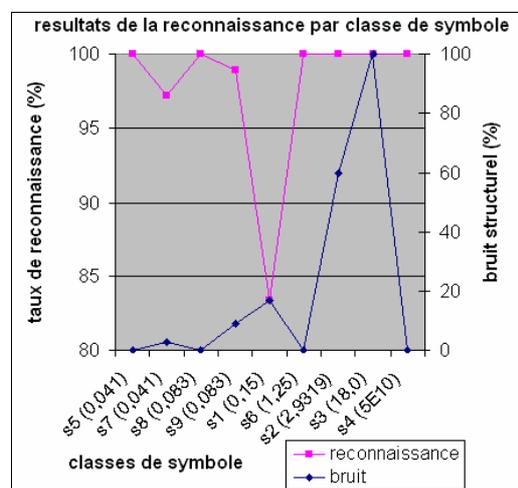


FIG. 11 – résultats de reconnaissance par classe

Dans le but d'analyser la distribution du bruit structurel en fonction des classes de symboles, nous avons calculé le taux de reconnaissance pour chaque classe de symbole sur les 6 tests (FIG. 11). Nous pouvons constater un haut niveau de bruit (>15%) pour les symboles s1, s2, et s3. La FIG. 12 donne des exemples de trois types de bruit structurel « classique ». Le premier est le bruit de sur-connexion de composantes (a), typiquement rencontré sur les symboles s1 et s2. Le deuxième est le bruit de fermeture des composantes (b). Ce bruit apparaît fréquemment sur les symboles s2, s3, et s7. Le dernier est un bruit de distorsion de voisinage (c). Dans certains cas, les distorsions des contours des formes perturbent l'extension des contours, et créent de fausses frontières (comme sur le symbole s7). Cependant, l'impact du bruit structurel sur la reconnaissance dépend de la classe du symbole. Dans le cas des symboles s7 et s1, le bruit structurel influe beaucoup sur la reconnaissance. Dans le cas des symboles s2 et s3, le bruit structurel n'a aucun effet sur les résultats de la reconnaissance. En effet, les graphes modèles de ces différents symboles n'ont pas le même

¹¹ Dans la suite, nous nommons ces tests test n°1, test n°2, etc.

« critère de robustesse ». Dans la FIG. 11, nous avons calculé (et trié) pour chaque modèle de graphe de symbole, les taux de similarité minimum avec les autres graphes de la base. Ce taux de similarité minimum est donc la distance entre le graphe modèle et son graphe le plus proche dans la base. Il peut être donc vu comme le critère de robustesse du graphe modèle. Ainsi, nous pouvons voir deux groupes principaux de graphes modèles dans la base. Les graphes modèles de faible robustesse (≤ 0.15) (s1, s5, s7, s8, et s9), et les graphes modèles de forte robustesse (≥ 1.25) (s2, s3, s4, et s6).

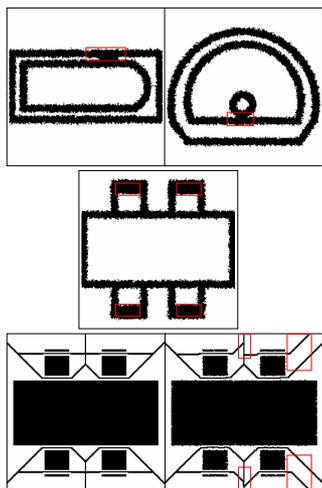


FIG. 12 - (a) sur-connexion
(b) fermeture
(c) distorsion de voisinage

7 Conclusion et perspectives

Dans cet article nous avons présenté un système de reconnaissance de symboles appliqué aux documents graphiques. Ce système exploite une combinaison d'approches structurelles pour la multi-représentation en graphes de régions des symboles. Il permet d'extraire les composantes et leurs occlusions, avec leurs relations d'inclusion et de voisinage. Il est ainsi possible de construire différents types de graphe selon le problème de reconnaissance envisagé. Ces graphes sont ensuite classés par un algorithme d'appariement de graphes. Ce système exploite une représentation XML des données, ce qui en permet une manipulation aisée. Nous présentons quelques résultats et leur analyse sur un jeu de symboles du concours de reconnaissance GREC2003. Ce système donne de très bons résultats. Toutefois il est sensible au bruit de dilatation. De plus, basé sur des graphes de régions, il ne permet que la reconnaissance des symboles segmentés, et de distinguer un faible nombre de classe.

En terme de perspectives à court terme [DEL 04], nous voulons tout d'abord réaliser une étape de prétraitement contextuel, afin d'adapter le prétraitement aux images dilatées. Par la suite, nous voulons enrichir nos graphes de régions avec des méthodes statistiques, afin de distinguer un plus grand nombre de classes. Nous souhaitons finalement superviser notre système par un pilotage automatique de programmes via une méthodologie de construction d'objets.

En terme de perspectives à long terme, nous voulons étendre l'apprentissage de notre système aux modèles de graphes bruités. Nous avons déjà réalisé un premier prototype d'interface basée sur l'apprentissage par l'exemple¹². Nous voulons ensuite étendre notre librairie d'analyse des composantes connexes avec une méthode de diagramme de Voronoi généralisée, de façon à réduire la complexité pour le traitement des plans de larges dimensions. Enfin, nous voulons combiner notre méthode d'appariement de graphes avec une méthode de grammaire de graphes, dans le but de corriger certains cas de bruit structurel.

Références

- [AGG 77] A.K. Aggarwal, A.V. Kulkarni, A Sequential Approach to the Extraction of Shape Features, *Computer Graphics and Image Processing*, 6(6) : 538-557, 1977.
- [BAD 02] O. El Badawy, M. Kamel, Shape Representation using Concavity Graphs, *International Conference on Pattern Recognition (ICPR)*, 2002.
- [BLO 95] D. Blostein, H. Fahmy, A. Grbavec, Issues in the Practical Use of Graph Rewriting, *Graphics Recognition (GREC)*, 1995.
- [BUR 95] M. Burge, G. Monagan, Using the Voronoi Tessellation for Grouping Words and Multi Part Symbols in Document, *Vision Geometry IV*, 1995.
- [DEL 03] M. Delalandre, E. Trupin, J.M. Ogier, Analyse Structurelle en Interprétation de Documents: Un Bref Survol, *International Conference on Image and Signal Processing (ICISP)*, 2003.
- [DEL 04] M. Delalandre, E. Trupin, J.M. Ogier, J. Labiche, Système Contextuel de Reconnaissance Structurelle de Symboles, Basé sur une Méthodologie de Construction d'Objets, *Conférence Internationales Francophone sur l'Ecrit et le Document (CIFED)*, 2004.
- [HAN 02] E. Hancock, R. Wilson, Graph-Based Methods for Vision: A Yorkist Manifesto, *Structural and Syntactical Pattern Recognition (SSPR)*, 2002.
- [HER 00] P. Héroux, S Diana, E. Trupin, Y. Lecourtier, A Structural Classification for Retrospective Conversion of Document, *Structural and Syntactical Pattern Recognition (SSPR)*, 2000.
- [KAS 02] R. Kasturi, L. O'Gorman, V. Govindaraju, Document Image Analysis: A Primer, *Sadhana*, 27(1) : 3-22, 2002.
- [LLA 01] J. Lladós, E. Valveny, G. Sánchez, E. Martí, Symbol Recognition : Current Advances and Perspectives, *Graphics Recognition (GREC)*, 2001.
- [LOO 01] P.K. Loo, C.L. Tan, Detection of Word Group Based on Irregular Pyramid, *International Conference on Document Analysis And Recognition (ICDAR)*, 2001.

¹² XML graphics modelling learning : <http://site.voila.fr/mdhws/>