

Reconnaissance de Symboles par Approche Structurale Globale-Locale, Basée sur l'Utilisation de Scénarios, et Exploitant une Représentation XML des Données

Delalandre Mathieu¹, Nicolas Stéphane¹, Trupin Eric¹, Ogier Jean-Marc²

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

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

¹{mathieu.delalandre;nicolas.stephane;eric.trupin}@univ-rouen.fr ; ²jmogier@univ-lr.fr

Résumé

Cet article aborde le problème de la reconnaissance de symboles sur les images de documents. Nous avons basé notre approche sur la combinaison des approches structurelles globale et locale. L'approche globale groupe les composantes connexes entre elles selon des contraintes de connexion et de voisinage. L'approche locale décompose une composante connexe en graphe d'objets géométriques (vecteurs, arcs, courbes, jonctions). Les graphes extraits sont exploités par un classifieur structurel qui permet l'appariement de graphes et de sous-graphes, de façon exacte et inexacte. L'adaptabilité du système est obtenue par l'utilisation de scénarios. Une représentation XML des données est utilisée dans le système permettant une transformation aisée des données, pour le contrôle des représentations des formes, et la création de représentations graphiques des résultats.

Abstract

This paper deals with the structural recognition of symbols on the documents. We have based our system on a combination of local and global structural approaches. The global approach groups the connected components together according to some closeness and connection constraints. The local approach splits up each connected component into a graph of geometrical objects (vectors, arcs, curves, junctions). The extracted graphs are matched thanks to a structural classifier, which permits graphs-subgraphs and exact-inexact matching. The system adaptability is obtained thanks to the scenarios use. A XML data representation is used, allowing the patterns models control and the graphic representations of results.

1. Introduction

Cet article aborde le problème de la reconnaissance de symboles sur les images de documents. Nous présentons ici un système basé sur la combinaison des approches structurelles globale et locale. L'objectif est d'obtenir un système adaptable, aussi nous avons basé notre système sur l'utilisation de scénarios. Dans la suite de cet article, nous présentons dans la section 2 le problème général de la reconnaissance de forme structurelle. Les sections 3,4,5, sont dédiées à la description des trois parties principales de notre système pour la reconnaissance de symboles : la classification structurelle dans la section 3, l'approche structurelle globale dans la section 4, et l'approche structurelle locale dans la section 5. Dans la section 6, nous présentons la construction et l'exploitation des scénarios. Dans la section 7, nous présentons des expérimentations et résultats obtenus sur une application de reconnaissance de symboles techniques. Dans la section 8, nous présentons l'utilisation de XML dans le système. Enfin, dans la section 9, nous concluons et donnons quelques perspectives.

2. Reconnaissance de forme structurelle

Un processus de reconnaissance de forme appliqué aux images est décomposé en deux étapes : une étape de traitement d'image et une étape de reconnaissance [13]. Dans le cas de la reconnaissance de forme structurelle, l'étape de traitement d'image extrait des graphes de l'image, reconnus par la suite par des méthodes d'appariement ou de grammaire de graphes [27]. En reconnaissance de documents, de nombreuses formes se prêtent à une description structurelle comme : les formulaires [12], les caractères orientaux [14], et les parties graphiques sur les documents [25]. Parmi ces parties graphiques, le symbole constitue un cas particulier, largement présent sur les documents techniques [25] ou sur les documents généraux [18]. Un état de l'art sur la reconnaissance de symboles peut être trouvé dans [16].

L'étape de reconnaissance structurelle est un problème de recherche de graphes. On emploie généralement deux grandes familles d'approches : soit les approches à base d'appariement de graphes [11], ou soit les approches à base de grammaire de graphes [2]. Les premières appariement les graphes candidats extraits à des graphes modèles : on parle de classification structurelle. Les secondes appliquent un ensemble de règles de transformation (ou grammaire) afin de faire converger les graphes candidats vers les graphes modèles. La recherche de graphes dépend de deux critères : recherche de graphes ou de sous-graphes, recherche exacte ou inexacte. Un sous-graphe est un sous-ensemble de nœuds et d'arcs appartenant à un graphe de taille plus importante. Lorsque les graphes candidats traités sont strictement équivalents aux graphes modèles, on dit que la recherche de graphes est exacte. Dans le cas de la reconnaissance structurelle de symboles, les symboles sont souvent connectés à d'autres éléments graphiques sur le document. C'est donc souvent un problème de recherche de sous-graphes [3]. De même, les graphes extraits lors de l'étape de traitement d'image sont souvent bruités, il s'agit donc de recherche inexacte [23].

L'étape de traitement d'image est un problème de construction de graphes. Les graphes construits sont non orientés, inexactes, et correspondent à un problème de recherche de graphes/sous-graphes. Dans [7] nous qualifions (dans un état de l'art que nous résumons ici) cette étape d'analyse structurelle des formes, et nous proposons une classification de cette analyse structurelle en approches globale et locale¹. La frontière entre ces deux approches est la composante connexe. La Figure 1 illustre la différence entre ces deux approches. L'approche locale (b) décompose une composante connexe (a) en arcs, jonctions et vecteurs. L'approche globale (c) groupe les composantes connexes selon des contraintes de distance et de voisinage.

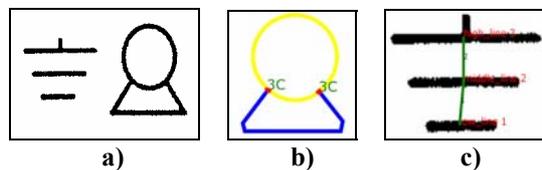


Figure 1: Approche globale et locale

L'approche globale analyse les relations spatiales entre les composantes connexes. On parle de graphes de voisinage. Des contraintes spatiales classiques (distance euclidienne, alignement, etc.) sont généralement utilisées, mais des approches comme les diagrammes de Voronoi peuvent être également employées. L'approche globale est généralement utilisée [18] pour la reconnaissance de formulaires, la reconstruction de chaînes de caractères ou de formules mathématiques, et le groupement de symboles. Elle utilise des informations de l'image pour construire les graphes, mais peut également exploiter des informations issues d'une étape de reconnaissance (voir section 4). L'approche locale emploie différentes techniques issues [25] de la reconnaissance de caractères, de la reconnaissance de symboles, et de la reconnaissance de parties graphiques. Généralement, il y a quatre étapes : une étape d'extraction des primitives (de type pixel, plage, région, maille, et objet géométrique), une étape d'approximation de ces primitives, une étape éventuelle de construction de primitives haut-niveaux, et une étape de correction des graphes de primitives. Notons que les graphes extraits par approche locale peuvent être traités ensuite par une approche globale, créant ainsi des descriptions multi-niveaux [10].

Nous avons sommairement présenté au cours de cette section le problème de la reconnaissance de forme structurelle appliquée aux documents, et plus spécialement aux symboles. Nous avons soulevé les problèmes de recherche de graphes que pose cette reconnaissance de forme structurelle. Nous avons présenté les deux grandes approches d'analyse structurelle (locale et globale), et expliqué qu'elles pouvaient être complémentaires. Nous exploitons cette complémentarité dans la suite de cet article pour la reconnaissance de symboles, essentiellement sur documents techniques.

3. Classification structurelle

La librairie de classification structurelle que nous exploitons inclue des méthodes pour l'appariement exact/inexact de graphes/sous-graphes. Les premiers algorithmes implémentés dans cette librairie étaient utilisés pour la reconnaissance de formulaires, ils permettent l'appariement exact/inexact de graphes [12]. Récemment, nous avons étendu cette librairie pour la reconnaissance structurelle de symboles en rajoutant des algorithmes pour l'appariement exact/inexact de sous-graphes tels que : des méthodes de recherche d'isomorphismes basées sur la recherche de cliques maximales, la méthode de Ullman [28], certaines méthodes de la VFLib² [5], et les méthodes de décomposition en réseau de Messmer [17] exacte et inexacte. Toutes ces méthodes ont été complétées avec une méthode pour la factorisation des sous-graphes détectés. Le but de cette

¹ Dans la suite de l'article, afin de clarifier le texte, nous qualifierons les analyses structurelles locale et globale d'approche locale et globale

² SIVA Laboratory, Università di Napoli Federico II, Italy, "VF Library Graph Matching Algorithm", <http://amalfi.dis.unina.it/graph>

factorisation est de réduire itérativement la taille des graphes d'entrée. Dans la suite de l'article, nous exploitons cette librairie avec les résultats des approches locales et globales pour la reconnaissance structurelle de symboles.

4. Approche globale

Dans [6] nous avons présenté une approche statistico-structurelle pour la reconnaissance de symboles. Cette approche est basée sur un traitement par approche globale des images des occlusions de plans techniques. Ces occlusions sont reconnues par l'utilisation de transformées mathématiques et d'un classifieur statistique de type k plus proche voisin(s) [1]. Par la suite, notre traitement structurel par approche globale est utilisé pour reconstruire les graphes d'occlusions correspondant aux symboles. Ces graphes sont ensuite appariés par la librairie de classification structurelle (voir section 3). Les bases d'apprentissage statistique et les bases de graphes modèles pour nos classifieurs sont obtenues par des interfaces d'acquisition (voir section 6). La Figure 2 donne un exemple de reconnaissance statistico-structurelle, avec l'étape (a-b) pour l'extraction des occlusions, et l'étape (b-c) pour : la classification statistique, l'approche globale, et la classification structurelle. Sur ces plans techniques, c'est donc l'étape d'extraction des occlusions qui nous permet d'appliquer l'approche globale.

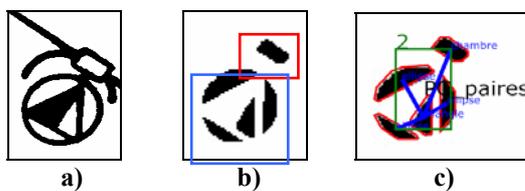


Figure 2: Reconnaitre statistico-structurelle

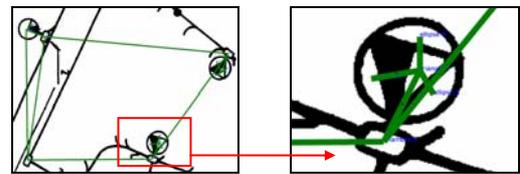


Figure 3: Stratégie de groupement

Le traitement par approche globale que nous utilisons construit des graphes à partir de l'image en utilisant des contraintes de connexion et de distance. Premièrement, toutes les connexions entre les composantes connexes sont créées. Les connexions sont ensuite sélectionnées selon une contrainte de distance euclidienne. Finalement, on applique des contraintes de connexions pour la construction finale des graphes des symboles. Ces contraintes de connexion définissent le nombre maximum de connexions autorisées par classe de composantes connexes (définies dans la base d'apprentissage statistique). Leur utilisation permet ainsi de diminuer la complexité des graphes créés. Evidemment, cette construction est très sensible aux résultats de la reconnaissance statistique, et les contraintes de connexion doivent donc être définies en conséquence. L'ordre d'application des contraintes peut être modifié, pour appliquer les contraintes de connexion d'abord et la contrainte de distance ensuite. Il est ainsi possible de contrôler les interconnexions entre les composantes connexes, et donc entre les graphes correspondant aux symboles, en suivant deux stratégies:

- i. Une construction par stratégie de segmentation isole chaque symbole sur l'image (Figure 2).
- ii. Une construction par stratégie de groupement crée un graphe correspondant à tous les symboles sur l'image (Figure 3).

Cette approche globale pose cependant des problèmes dans le cas des symboles trop proches. En effet, deux symboles proches seront reconstruits comme un seul graphe. Par exemple, dans la Figure 2 (b), nous avons deux symboles reconstruits comme un seul graphe (Figure 2 (c)). Le rectangle du haut représente le premier symbole (une occlusion), et le rectangle du bas le second (4 occlusions). Une voie pour résoudre ce problème est l'utilisation de l'approche locale dans le but de compléter les résultats de l'approche globale. Nous présentons cette utilisation dans la section suivante.

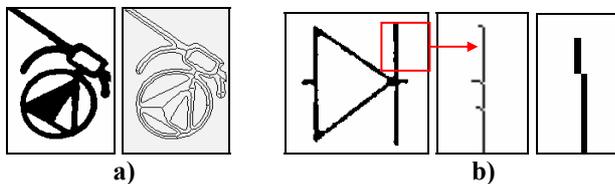
5. Approche locale

Notre approche locale est basée sur l'extraction de graphes de listes de pixels (détection de contour et squelettisation), et l'utilisation de fonctions d'approximation mathématique (vectorisation, recherche de courbes et d'arcs de cercle). Dans l'objectif d'exploiter une approche à base de scénarios (voir section 6), nous avons décomposé la chaîne classique de vectorisation en traitements granulaires. Nous parlons de scénarios de vectorisation. Nous exploitons une librairie décomposée en trois niveaux de traitements (Tableau 1) : un niveau données image, un niveau données structurées, et un niveau frontière entre les données image et les données structurées. Chaque niveau est composé de différents traitements granulaires. Nous présentons chacun de ces niveaux dans la suite de cette section. Une description plus complète de notre approche locale et de notre librairie sera publiée dans [8].

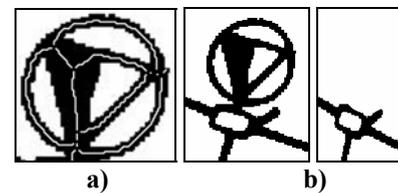
Tableau 1 : Niveau de traitement dans l'approche locale

Niveau Image	Adaptation à la Squelettisation, Détection de Contour, Squelettisation
Niveau Frontière	Simplification Progressive d'Objets, Extraction de Listes de Pixels
Niveau Données Structurées	Segmentation d'épaisseur, Reconstruction de Jonctions, Construction de Graphes, Vectorisation, Recherche de Courbes et d'Arcs, Post-corrections

Le niveau image utilise des traitements d'image classiques pour la détection de contour et la squelettisation. Pour la squelettisation, nous utilisons deux algorithmes basés sur l'utilisation de transformées de distance : les algorithmes de Di Baja [9] et Taconet [26]. Pour la détection de contour, nous utilisons un algorithme classique, basé sur un test de voisinage exploitant un élément structurant 3*3 de type diagonal 'X' ou vertical '+'. L'image des contours obtenue est ensuite affinée de façon à éliminer les pixels n-connexes superflus (3 et plus). La Figure 4 (a) donne un exemple de résultats superposés de squelettisation et de détection de contour. Le principal problème de la squelettisation est sa dépendance au bruit, fonction de l'épaisseur de la forme analysée. Dans le but d'adapter l'image à la squelettisation, nous utilisons la transformée 3-4 de Di Baja [9] en combinaison avec un algorithme classique de réduction d'image. La Figure 4 (b) donne un exemple de l'impact de la squelettisation sans réduction (image du milieu) et avec réduction (image de droite).



**Figure 4: a) Squelettisation et détection de contour
b) Adaptation à la squelettisation**



**Figure 5: a) Extraction des listes
b) Simplification d'un objet arc**

Notre niveau frontière utilise deux traitements : le premier permet de convertir l'image en données structurées, et le second permet d'exploiter ces données structurées sur l'image. Le premier traitement permet d'extraire les listes de pixels, afin de convertir les images de squelette et de contour en données structurées. Ce traitement détruit, dans un premier temps, tous les pixels 3-connexes. Par la suite, les pixels restants sont chaînés en listes de pixels. Chaque liste est donc composée de pixels 1-connexe (les extrémités), et de pixels 2-connexes (les pixels centraux de la chaîne). La Figure 5 (a) donne un exemple de données structurées extraites à la suite d'une squelettisation (et superposées à l'image initiale). Le second traitement permet d'exploiter les données structurées sur l'image. Le but est de simplifier itérativement l'image afin de réduire la complexité du processus de reconnaissance [24], c'est de la simplification d'objets. Actuellement, ce traitement ne permet d'effacer que les objets de type arc. La Figure 5 (b) donne un exemple de résultat d'effacement d'arc.

Le niveau données structurées est la partie centrale de notre approche à base d'utilisation de scénarios de vectorisation. En effet, tous les traitements utilisent le même format d'entrée/sortie. En pratique nous pouvons utiliser les traitements dans n'importe quel ordre. Cependant en théorie, certains traitements dépendent d'un certain niveau de structuration des données (par exemple notre traitement de recherche d'arcs exploite les résultats de la vectorisation). Notre format de donnée représente un graphe d'objets géométriques composé d'arcs, de courbes, de poly lignes, de listes de pixels, et de jonctions. Le but du processus de scénario de vectorisation est d'ajouter de l'information sémantique au graphe basique obtenu par le traitement d'extraction de listes de pixels. Pour cela, nous utilisons les différents traitements granulaires du niveau données structurées du Tableau 1. Pour la segmentation d'épaisseur nous fragmentons nos listes de pixels selon un simple test de variation des degrés d'intériorité entre pixels consécutifs. L'information des degrés d'intériorité des pixels est obtenue par la succession des traitements de squelettisation et d'extraction des listes de pixels. La Figure 6 donne un exemple de résultat de segmentation d'épaisseur, avec l'image originale (a), les représentations graphiques des listes de pixels sans segmentation (b), et après segmentation d'épaisseur (c). Pour la polygonisation (transformation des listes de pixels en poly lignes 'ou liste de vecteurs'), nous utilisons l'algorithme d'approximation polygonale de la corde [20] en combinaison avec la procédure de fusion de [19]. Pour chacune des poly lignes, nous calculons un certain nombre d'attributs comme : l'épaisseur des vecteurs (calculée sur la liste de pixels initiale), la longueur des vecteurs, les relations angulaires entre deux vecteurs consécutifs (fonction du parcours de la liste de vecteurs). Notre traitement de recherche d'arcs est une approche basique uniquement basée sur l'analyse des relations angulaires et de longueur entre les vecteurs de la poly lignes. Nous utilisons ici une propriété géométrique classique : un cercle parfait peut être approximé par un polygone régulier en angle et en longueur. Les arcs sont recherchés selon des critères de plages minima / maxima de proportion circulaire, et de

périmètre. Récemment, nous avons étendu nos traitements d'approximation avec un algorithme de recherche de courbes de Bernstein [15]. La Figure 7 donne des exemples de représentations graphiques appliquées à des images de contours, de la polygonisation et de la recherche d'arcs de cercle (a), et de l'approximation en courbes de Bernstein (b).

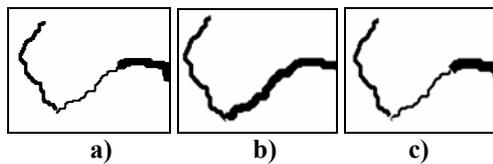


Figure 6: Segmentation d'épaisseur

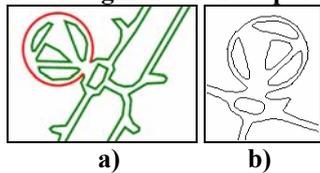


Figure 7: a) Vectorisation et recherche d'arc
b) Recherche de courbe

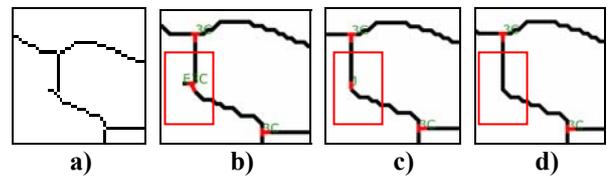


Figure 8: Ebarbulage & fusion

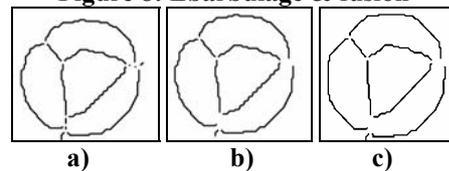


Figure 9: Filtrage & lissage

Un traitement important de l'approche locale est la détection des jonctions. Nous avons basé notre approche sur un algorithme de reconstruction des jonctions. Premièrement, cet algorithme construit toutes les connexions entre les extrémités des objets, en utilisant une contrainte de distance euclidienne. Ensuite, il analyse toutes les connexions de façon à trouver les interconnexions entre les extrémités, chacune de ces interconnexions correspond à une jonction. Durant cette analyse, les deux extrémités appartenant à un même objet ne peuvent être affectées à une même connexion. Une approche similaire à la notre peut être trouvée dans [14]. Avec l'information sur les jonctions, nous utilisons un algorithme de construction de graphes. Cet algorithme exploite les informations sur les jonctions et sur les objets extraits de façon à construire des graphes d'objets géométriques composés de listes de pixels, d'arcs, de courbes, et de poly lignes. L'analyse du graphe permet d'exploiter différents traitements de post-corrections structurelles comme l'ébarbulage et la fusion. La Figure 8 donne un exemple de reconstruction de jonctions (a, b), suivie d'un traitement d'ébarbulage (b, c), puis d'un traitement de fusion des listes (c, d). Nous exploitons également d'autres traitements de post-corrections au cours du scénario de vectorisation. La Figure 9 (a, b) donne un exemple de filtrage des segments. Le seuil appliqué peut être défini manuellement ou automatiquement (basé sur une recherche du rapport proportionnel maximum des longueurs). Ce filtrage permet de diminuer la complexité du traitement de détection des jonctions. La Figure 9 (b, c) donne un exemple de lissage des chaînes de pixels (basée sur l'analyse des variations des directions de Freeman), afin d'accroître la qualité des données d'entrée des étapes de vectorisation, et de recherche d'arcs et de courbes.

6. Construction et exploitation des scénarios

Avec les traitements de la classification structurelle, de l'approche globale et locale, nous construisons différentes chaînes de traitements. Nous parlons de construction de scénarios. Pour chacun de ces scénarios, nous acquérons également les bases de modèles statistiques et structurels adaptées. Le but est de tester différentes stratégies pour un même objectif de reconnaissance ; ou d'adapter nos traitements à de nouvelles applications. Pour ce faire, nous exploitons un système d'interprétation de documents. Ce système se veut adaptable et évolutif, nous l'avons donc basé sur l'exploitation d'une base de connaissances. Ce système est composé de trois parties:

- Un ensemble d'interface : Ces interfaces permettent l'acquisition et l'exécution des scénarios [21], et l'acquisition des bases d'apprentissage statistique [22] et des bases de graphes modèles (les graphes modèles acquis sont idéaux, le système ne réalise donc pas d'apprentissage structurel).
- Une base de connaissance : Cette base de connaissance est en cours de réalisation. Elle contient l'ensemble des informations manipulées par notre système (corpus d'images, scénarios, bases de modèles, résultats d'exécution). Nous avons choisi de représenter cette base en XML (voir section 8). La recherche des connaissances dans la base se fait via des requêtes et/ou via nos méthodes d'appariement de graphes. Cette base est pour le moment faiblement structurée, seuls nos scénarios [21] lient nos différentes connaissances entre elles.
- Une librairie de reconnaissance de forme : Cette librairie contient tous les traitements présentés dans cet article, plus des traitements standards de traitement d'image et de reconnaissance de forme statistique. Une brève présentation de cette librairie pourra être trouvée dans [4].

7. Expérimentations et résultats

Nous présentons ici une application de reconnaissance exploitant les approches : globales, globales/locales, et locales. Cette application de reconnaissance concerne des symboles techniques représentant des équipements de raccordement téléphonique sur les plans de réseaux (Figure 3) de l'opérateur Français de Télécommunication³ (FT). Comme présenté dans la section 6, notre système est adaptable. Il a déjà été exploité sur d'autres applications comme la reconnaissance de symboles météorologiques [6], et est en cours de test sur des symboles électriques et architecturaux [8]. Sur les symboles FT, nous avons de 1 à 4 occlusions par symbole, et de 3 à 5 classes de symboles. Le problème majeur de cette application est la proximité des symboles sur les plans techniques (voir section 4). Nous avons testé 5 stratégies de traitement différentes. Pour cela, nous avons créé environ 65 scénarios en combinant nos différents traitements. Nous avons testé ces 65 scénarios sur 10 extraits de plan de réseaux, constitués d'une cinquantaine de symboles et de 110 occlusions. Pour l'approche globale, ces occlusions ont été traitées avec deux bases d'apprentissage statistique différentes (une complète et une altérée), et 7 règles différentes de construction. Nous avons ainsi créé une base de test de graphes d'environ 700 graphes de symbole, composés d'environ 1400 nœuds. Nous avons exploité cette base de graphes avec toutes les méthodes d'appariement de graphes/sous-graphes de notre librairie de classification structurelle (voir section 3). Nous ne traitons pas dans cet article de l'étape de reconnaissance statistique utilisée dans les approches globales, celle-ci a été abordée dans [1] et [6]. Nous détaillons chacune des stratégies dans la suite de cette section. Nous comparons d'abord les stratégies exploitant les approches globales (les stratégies 1-2-3) et globales/locales (la stratégie 4). Le Tableau 2 donne les résultats comparatifs de ces quatre stratégies. Les résultats statistiques de 100% de reconnaissance (première base) sont notés '0% de bruit', et les résultats statistiques de 49% de reconnaissance (deuxième base) sont notés '51% de bruit'. Nous présenterons ensuite une stratégie exploitant une approche purement locale (la stratégie 5).

Les première et seconde stratégies sont des approches globales par segmentation, avec une reconnaissance de graphes (stratégie 1) et de sous-graphes (stratégie 2). Nous avons déjà présenté la première stratégie dans [6]. Dans cette première stratégie, nous avons considéré les symboles trop proches comme des nouveaux symboles uniques dans les bases de graphes modèles (de trois à cinq classes de symbole). Nous avons appliqué deux règles de construction (contraintes de distance et de connexion, contrainte de distance seule) sur les résultats de la reconnaissance des occlusions (avec nos deux bases d'apprentissage).

La troisième stratégie utilise une approche globale par groupement, avec une reconnaissance de sous-graphes. Nous avons testé cette stratégie avec nos deux bases d'apprentissage statistique, et 5 règles de construction différentes : aucune contrainte (construction de graphes complets), contrainte de distance élevée, 3 règles de contraintes de connexion différentes.

La quatrième stratégie utilise une approche locale avec une approche globale. L'approche locale détecte les arcs des symboles sur les images résultats de la détection de contour. Notre intention est d'effacer les zones des symboles délimitées par leurs arcs, de façon à réduire la complexité de l'approche globale. En effet avec cette stratégie, nous pouvons effacer les symboles proches (voir section 4). La Figure 10 (a) et (b) présentent les résultats de l'extraction des occlusions de la Figure 5 (b gauche) et (b droite). La Figure 10 (c) est le résultat de l'extraction de l'image non commune entre la Figure 10 (a) et (b). Suite à cette approche locale, l'approche globale présentée dans la deuxième stratégie est utilisée sur les images résultats d'extraction non commune (Figure 10 (c)). Le graphe correspondant au symbole effacé (rectangle du haut de la Figure 2 (b)) a été supprimé de la base de modèle de graphes.

Si les graphes ne sont pas bruités, nous obtenons des résultats parfaits avec toutes les stratégies. Les contraintes de connexion employées par notre approche globale permettent de contrôler la structure des graphes créés (voir section 4). Elles permettent ainsi d'en diminuer la complexité (en matière de nombre d'arcs). Elles permettent également de segmenter les symboles proches avec une approche graphe (stratégie 1), et donc de donner les mêmes résultats qu'une approche sous-graphe (stratégies 2 et 3). Pour un haut niveau de bruit, nous obtenons des résultats incohérents avec la stratégie de groupement (stratégie 3). En effet, les graphes correspondant aux symboles sont de faibles tailles (1 à 5 nœuds). Une fausse reconnaissance statistique sur 1 nœud a des effets importants sur la reconnaissance complète du symbole, et rend la détection d'un sous-graphe difficile. Les stratégies globales par segmentation donnent de meilleurs résultats (stratégies 1,2), et l'approche graphe (stratégie 1) est plus efficace que l'approche sous-graphe (stratégie 2). En effet, on obtient un problème de graphe moins complexe quand on peut obtenir un graphe pour un symbole. Pour un haut niveau de bruit,

³ France Télécom R&D: <http://www.rd.francetelecom.fr/>

seules les contraintes de distance sont exploitables. En effet, les contraintes de connexion sont très sensibles aux résultats statistiques. Ceci empêche leur utilisation (pour un haut niveau de bruit) avec la stratégie 1, pour la détection des symboles proches (Figure 10 (b)). La stratégie globale-locale donne les résultats les plus intéressants, avec un bon taux de reconnaissance et le traitement des symboles proches. En effet, la détection d'arcs de cercle corrige d'une certaine manière les résultats de la reconnaissance statistique (voir Figure 10). Avec notre traitement basique de détection d'arcs de cercle, nous avons reconnu environ 95% des arcs des symboles (21/22 arcs). De plus, sur nos images de contour, l'utilisation des critères de plages (voir section 5) n'autorise pas de fausses détections d'arcs de cercle. Ceci évite les altérations des images de plan de réseaux que produirait la simplification d'objets suite à de fausses détections d'arcs. L'inconvénient de l'approche globale-locale est la dépendance très locale (voir stratégie 4) de la reconnaissance des symboles proches. Les occlusions extraites de l'image simplifiée (Figure 10 (b)) sont automatiquement assimilées à ces symboles. L'efficacité de cette approche est donc très sensible aux performances de la détection d'arcs de cercle. Sur toutes ces stratégies, le choix de l'algorithme d'appariement (voir section 3) employé pour la reconnaissance de graphes ou de sous-graphes n'a que peu d'importance. Les graphes sont de tailles trop faibles pour qu'une méthode d'appariement puisse démontrer sa supériorité.

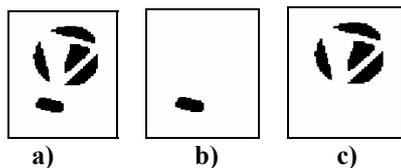


Figure 10: Extraction des occlusions après une détection d'arc

Stratégies :	1	2	3	4
0% de bruit	100%	100%	100%	100%
51% de bruit	86.6%	46.8%	~	87.5%

Tableau 2 : Comparaison des stratégies globale/globale-locale

Nous avons testé une cinquième stratégie purement locale. Cette stratégie a pour but la reconnaissance structurale de symboles directement à partir du graphe local. Elle permet également la recherche de zones d'intérêt pour la détection des caractères connectés aux réseaux. Après l'effacement des zones d'arcs de cercle des symboles (Figure 5 (b)), on réalise une extraction du graphe de listes de pixels (jonction, ébarbulage, fusion) (Figure 8). Ce graphe de pixels est ensuite exploité pour la recherche des symboles connectés aux réseaux (rectangle du haut de la Figure 2 (b)). Pour cela, on recherche les cycles dans les graphes candidats afin de détecter cette classe de symbole. Pour la détection des caractères collés, un graphe est construit avec les informations de longueur sur les listes de pixels. Cette possibilité de construction multi-représentation est obtenue grâce à l'utilisation de XML (voir section 8). Nous cherchons dans ce nouveau graphe les groupes de nœuds du graphe reliés et labellisés 'court' selon un seuil. La Figure 11 donne un exemple de résultat. Sur cette image, nous avons détecté le symbole et la zone d'intérêt pour le caractère collé au réseau. Des tests plus complets de stratégies purement locales sont en cours de réalisation et seront présentés dans [8].

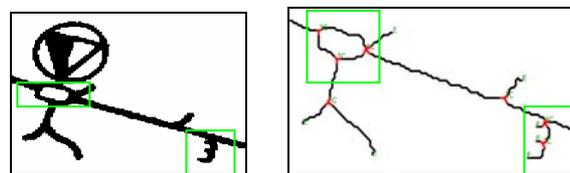


Figure 11: Reconnaissance structurale locale

8. Utilisation de XML dans le système

Nous utilisons XML⁴ pour représenter toutes les connaissances manipulées par notre système (voir section 6). L'utilisation de ce langage offre plusieurs avantages :

- Premièrement, XML dispose de sous langages spécialisés. Ces langages servent de normes de représentation, et permettent d'exploiter un ensemble d'outils informatiques associés existants [6]. Nous utilisons SVG⁵ pour les représentations graphiques de nos résultats, et XGMML⁶ pour les descriptions des graphes manipulés par notre librairie de classification structurale.

⁴ eXtensible Markup Language : <http://www.w3.org/>

⁵ Scalable Vector Graphics

⁶ eXtensible Graph Markup and Modeling Language

- Deuxièmement, XML permet d'utiliser des processeurs de transformation. Ils transforment les flots de données XML à l'aide de scripts XSLT⁷. Ceci permet l'adaptation des formats d'entrée de nos traitements (nous exploitons nos scénarios XML par nos méthodes d'appariement de graphes (voir section 6)). Ceci permet également le contrôle de nos modèles de représentation des formes (nous créons plusieurs graphes différents à partir d'un même flot de données XML (voir stratégie 5, section 7)). Ceci permet enfin la création de nos représentations graphiques (à partir de nos différents résultats XML nous créons nos représentations graphiques en SVG (voir Figure 2)).
- Enfin, nous utilisons Quilt, un langage candidat XML-QL⁸, pour exploiter un mécanisme de requête sur notre base de connaissance (voir section 6). Nous combinons également Quilt avec XSLT pour rechercher les informations communes dans les différents flots XML résultats, et les réorganiser de façon à produire un document XML final cohérent.

9. Conclusion et perspectives

Dans cet article, nous avons présenté un système pour la reconnaissance structurelle, utilisant une combinaison des approches locale et globale, et basé sur l'utilisation de scénarios. Avec l'approche globale, une mauvaise reconnaissance statistique sur des petits symboles impliquera une tâche difficile de reconnaissance structurelle. L'utilisation de l'approche locale réduit la complexité de l'approche globale, et augmente ainsi les résultats de la reconnaissance. De plus, cette approche locale peut permettre la détection de zones d'intérêt pour les caractères collés à des parties graphiques. L'utilisation de scénarios permet de tester différentes stratégies selon les objectifs de reconnaissance, et d'adapter le système à de nouvelles applications. Cette utilisation permet de plus, pour une même stratégie, de tester l'influence des paramètres des traitements, et d'utiliser plusieurs modèles de représentation de la forme. L'utilisation de XML dans le système permet un contrôle aisé des données. En terme de perspectives, nous approfondissons actuellement notre approche purement locale [8], et souhaitons faire évoluer nos interfaces pour l'apprentissage structurel. De plus, notre système actuel n'analyse pas la cohérence des résultats entre les différentes stratégies. L'analyse de cette cohérence permettrait, plus qu'une combinaison, une réelle interaction entre les approches globale et locale. Dans cette optique, nous étudions actuellement les possibilités de RuleML⁹ [29] pour inférer directement nos flots résultats XML. Enfin, notre base de connaissance actuelle est faiblement structurée. Sa structuration permettrait d'accroître les possibilités d'adaptation de notre système. Dans ce sens, nous étudions actuellement les possibilités de RDF¹⁰ afin de structurer notre base de connaissances.

10. Références

- [1] S. Adam, J.M. Ogier, C. Cariou, J. Gardes, Y. Lecourtier, "Combination of Invariant Pattern Recognition Primitive on Technical Documents", *Graphics Recognition (GREC)*, 1999.
- [2] D. Blostein, H. Fahmy, A. Grbavec, "Issues in the Practical Use of Graph Rewriting", *Graphics Recognition (GREC)*, 1995.
- [3] H. Bunke, P. Foggia, C. Guidobaldi, C. Sansone, M. Vento, "A Comparison of Algorithms for Maximum Common Subgraph on Randomly Connected Graphs", *Structural and Syntactical Pattern Recognition (SSPR)*, 2002.
- [4] E. Clavier, G. Masini, M. Delalandre, M. Rigamonti, J.M. Ogier, J. Gardes, "DocMining: A Cooperative Platform for Heterogeneous Document Interpretation According to User-Defined Scenarios", *Graphics Recognition (GREC)*, 2003.
- [5] L. P. Cordella, P. Foggia, C. Sansone, M. Vento, "An Improved Algorithm for Matching Large Graphs", *Graph Based Representation (GBR)*, 2001.
- [6] M. Delalandre, P. Héroux, S. Adam, E. Trupin, J.M. Ogier, "Une Approche Statistico-Structurelle pour la Reconnaissance de Symboles Exploitant une Représentation XML des Données", *Colloque International Francophone sur l'Écrit et le Document (CIFED)*, 2002.
- [7] 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.

⁷ eXtensible Stylesheet Transform Language

⁸ XML Query Language

⁹ Rule Markup Language

¹⁰ Ressource Description FrameWork

- [8] M. Delalandre, Y. Saidali, J.M. Ogier, E. Trupin, "Adaptable Vectorisation System Based on Strategic Knowledge and XML Representation Use", *Graphics Recognition (GREC)*, 2003.
- [9] G.B Di Baja, "Well shaped, Stable, and Reversible Skeletons from the 3-4 Distance Transform", *Journal of Visual Communication and Image Representation*, 5(1) : 107-115, 1992.
- [10] O. El Badawy, M. Kamel, "Shape Representation using Concavity Graphs", *International Conference on Pattern Recognition (ICPR)*, 2002.
- [11] E. Hancock, R. Wilson, "Graph-Based Methods for Vision: A Yorkist Manifesto", *Structural and Syntactical Pattern Recognition (SSPR)*, 2002.
- [12] P. Hérroux, S Diana, E. Trupin, Y. Lecourtier, "A Structural Classification for Retrospective Conversion of Document", *Structural and Syntactical Pattern Recognition (SSPR)*, 2000.
- [13] R. Kasturi, L. O'Gorman, V. Govindaraju, "Document Image Analysis : A Primer", *Sadhana*, 27(1) : 3-22, 2002.
- [14] K.K. Lau, P.C. Yuen, Y.Y. Tang, "Stroke Extraction and Stroke Sequence Estimation On Signatures", *International Conference on Pattern Recognition (ICPR)*, 2002.
- [15] C. W Liao, J. S. Huang, "Stroke Segmentation by Bernstein-Bezier Curve Fitting", *Pattern Recognition (PR)*, 23(5) : 475-484, 1990.
- [16] J. Lladós, E. Valveny, G. Sánchez, E. Martí, "Symbol Recognition : Current Advances an Perspectives", *Graphics Recognition (GREC)*, 2001.
- [17] B.T Messmer, "Efficient Graph-matching Algorithms for Preprocessed Model Graphs", *Phd Thesis of Bern University, Switzerland*, 1995.
- [18] G. Nagy, "Twenty Years of Document Image Analysis in PAMI", *Pattern Analysis and Machine Intelligence (PAMI)*, 22(1) : 38-62, 2000.
- [19] T. Palvidis, "A Vectorizer and Feature Extractor for Document Recognition", *Computer Vision Graphics and Image Processing*, 35 : 111-127, 1986.
- [20] V. Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves", *Computer Vision Graphics and Image Processing*, 1(3) : 244-246, 1972.
- [21] Y. Saidali, N. Beaudoin, E. Trupin, M. Holzen, J. Labiche, "ACTI_VA : Plate Forme Interactive pour l'Acquisition de Connaissance Traiteur d'Images de Document", *Colloque International sur le Document Electronique (CIDE)*, 2002.
- [22] Y. Saidali, S. Adam, J.M. Ogier, E. Trupin, J. Labiche, "Knowledge Representation and Acquisition for Engineering Document Analysis", *Graphics Recognition (GREC)*, 2003.
- [23] G. Sánchez, Joseph Lladós, K. Tombre, "An Error-Correction Graph Grammar to Recognize Texture Symbols", *Graphics Recognition (GREC)*, 2001.
- [24] J. Song, F. Su, C. Tai, S. Cai, "An Object-Oriented Progressive-Simplification based Vectorisation System for Engineering Drawings: Model, Algorithm and Performance", *Pattern Analysis and Machine Intelligence (PAMI)*, 24(8) : 1048-1060, 2002.
- [25] J. Song, M.R. Lyu, M. Cai, S. Cai, "Graphic Object Recognition from Binary Images: a Survey and an Integrated Paradigm", *Transactions on Systems Man and Cybernetics part C : Applications and Reviews (TSMCC)*, under review.
- [26] B. Taconet, A. Zahour, S. Zhang, A. Faure, "Deux Algorithmes de Squelettisation", *Reconnaissance Automatique de l'Ecriture (RAE)*, 1990.
- [27] K.Tombre, "Structural and Syntactic Methods in Line Drawing Analysis : To Which Extent do they Work ?", *Structural and Syntactical Pattern Recognition (SSPR)*, 1996.
- [28] J.R Ullman, "An Algorithm for Subgraph Isomorphism", *Journal Association for Computer Machinery*, 23 : 31-42, 1976.
- [29] G. Wagner, "How to Design a General Rule Markup Language", *XML Technologien für das Semantic Web (XML TSW)*, 2002.