



Système contextuel de reconnaissance structurelle de symboles, basé sur une méthodologie de construction d'objets

Mathieu Delalandre, Eric Trupin, Jean-Marc Ogier, Jaques Labiche

► To cite this version:

Mathieu Delalandre, Eric Trupin, Jean-Marc Ogier, Jaques Labiche. Système contextuel de reconnaissance structurelle de symboles, basé sur une méthodologie de construction d'objets. Jun 2004, 2004. <sic_00001185>

HAL Id: sic_00001185

https://archivesic.ccsd.cnrs.fr/sic_00001185

Submitted on 6 Dec 2004

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

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

Système contextuel de reconnaissance structurelle de symboles, basé sur une méthodologie de construction d'objets

Mathieu Delalandre¹ – Eric Trupin¹ – Jean-Marc Ogier² – Jaques Labiche¹

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 appliqué aux documents graphiques. Celui-ci est basé sur une approche contextuelle de reconnaissance structurelle de symboles exploitant sur une Méthodologie de Construction d'Objets. Il exploite une librairie de traitements composée d'extracteurs de primitives graphiques essentiellement structurelles, et de traitements d'évaluation contextuelle. Ces traitements sont automatiquement contrôlés par un moteur d'inférence durant le processus de reconnaissance. Les connaissances factuelles, stratégiques, et descriptives du système sont structurées selon une Méthodologie de Construction d'Objets fonction de l'application de reconnaissance. Cette structuration originale des connaissances permet au système d'accroître ses performances de reconnaissance. Nous illustrons l'intérêt de cette approche à travers des résultats sur un jeu test de symboles du concours GREC2003.

Mots-clés : reconnaissance de symboles, graphe, pilotage de traitements, connaissances, méthodologie de construction d'objets.

1 Introduction

Cet article aborde le problème de la reconnaissance de symboles appliquée aux documents graphiques. Nous présentons ici un système contextuel de reconnaissance structurelle de symboles basé sur une Méthodologie de Construction d'Objets (OPM¹)². Dans la suite de l'article, nous présentons dans la section (2) le problème général de la reconnaissance de symboles appliquée aux documents graphiques. Les deux sections suivantes sont dédiées aux deux parties principales de notre système : la librairie de traitements dans la section (3), et le système contextuel à base d'OPM dans la section (4). Dans la section (5) nous présentons un cas d'usage du système, sur un jeu test de symboles du concours GREC2003. Finalement, dans la section (6), nous concluons et donnons nos perspectives.

2 Système de reconnaissance de symboles

La reconnaissance de symboles [LLA 01] est une thématique prépondérante du domaine de l'interprétation de documents graphiques [ABL 00]. Au même titre que les systèmes d'interprétation de documents graphiques [ABL 00], les systèmes de reconnaissance de symboles font appel à trois entités différentes : une chaîne de reconnaissance de formes, un ensemble de connaissances, et un système de contrôle. Nous présentons chacune d'entre elles dans les sous-sections (2.1), (2.2), et (2.3). Dans la sous-section (2.4), nous introduisons et positionnons notre système.

2.1 Chaîne de reconnaissance de formes

La chaîne de reconnaissance de formes est décomposée en deux étapes principales [OGI 00] : une étape d'analyse, et une étape de reconnaissance. Deux approches principales existent : statistique et structurelle. Cet article traite majoritairement de l'approche structurelle. Celle-ci utilise des représentations graphes des objets du document. En effet, dans les documents graphiques, de nombreux objets peuvent être représentés par des graphes [ABL 00], et spécialement les symboles [LLA 01]. Ainsi, dans une chaîne de reconnaissance structurelle de symboles, l'étape d'analyse extrait des graphes représentant les symboles, et l'étape de reconnaissance structurelle les exploite.

L'étape de reconnaissance utilise deux grandes familles de méthodes : à base d'appariement de graphes [HAN 02], et à base de grammaire de graphes [BLO 95]. Les premières appartiennent aux graphes extraits avec des graphes modèles. Les secondes appliquent différentes règles afin de transformer les graphes extraits en graphes modèles. L'étape d'analyse extrait de l'image des graphes correspondant aux symboles. Cette étape fait appel à des approches variées [LLA 01] comme les approches à base de : de graphes de régions, de squelettes/contours, etc.

¹ Object-Process Methodology [DOR 02]

² Nous adoptons ici cette traduction plus apte à exprimer l'application des OPM à un processus de reconnaissance de formes.

2.2 Connaissances

Les connaissances dans un système sont soit dynamiques (temporaires) ou statiques. Dans le dernier cas, la base de connaissances regroupe et structure de façon déclarative les connaissances utilisées, ou générées [ABL 00]. Dans un système de reconnaissance, on en distingue trois types : factuelle, descriptive, et stratégique.

Les connaissances factuelles incluent les données à traiter (images et/ou des données structurées) et leurs informations contextuelles (qualité, type, etc.). Les données à traiter [WOR 01] sont soit des données d'entrée soit produites par les traitements du système. Les informations contextuelles peuvent être obtenues par le biais d'interfaces, ou par le biais de traitements d'évaluation contextuelle [ROS 98].

Les connaissances descriptives décrivent les objets susceptibles de figurer sur les documents. Elles sont exploitées par les traitements de reconnaissance. Deux formalismes de représentation sont donc couramment utilisés : les règles [BLO 95] et les graphes modèles [HAN 02]. D'autres formalismes de représentation peuvent être simultanément utilisés [OGI 00] comme les approches statistique/structurelle, matching/structurelle, etc. Ces connaissances descriptives peuvent être acquises par des interfaces d'édition [DEL 04] ou par des techniques d'apprentissage [SAI 03].

Les connaissances stratégiques [SAI 03] décrivent les graphes correspondant aux séquençement des traitements : les scénarios. On distingue deux types de système de construction : les systèmes de programmation graphique du scénario par les utilisateurs [SAI 03], et les systèmes contextuels [HEN 03]. Dans le dernier cas, les systèmes auto-génèrent tout ou partie leurs scénarios en fonction du contexte.

2.3 Le système de contrôle

Le système de contrôle [OGI 00] supervise le processus de reconnaissance. Il existe deux grandes familles de systèmes : planifiés a priori et contextuels. Les premiers utilisent un scénario figé de traitements. Ils sont « historiques » et largement utilisés dans les systèmes de reconnaissance de symboles [LLA 01]. Ils deviennent obsolètes au profit des systèmes contextuels [OGI 00]. Ces derniers auto-génèrent leurs scénarios en fonction du contexte. Plusieurs approches existent, cependant aucune d'entre-elles n'a su démontrer sa supériorité :

Les systèmes à base de tableau noir (*blackboard*) [ABL 00] constituent les ancêtres des SMA³. Dans ces systèmes chaque traitement (ou chaîne de traitements) et encapsulé dans un processus. Un mécanisme de contrôle supervise l'accès à la mémoire commune (le *blackboard*) et active les processus en fonction de l'état de cette mémoire. Cette approche est largement utilisée par les systèmes de traitement de documents graphiques [ABL 00].

Les SMA³ sont une alternative aux systèmes à base de tableau noir. Ce sont des systèmes très largement étudiés en intelligence artificielle [HEN 03]. Ils sont dits à contrôle décentralisé, car le contrôle est réparti entre les différents agents du système. Leur utilisation reste cependant marginale dans les systèmes de traitement de documents graphiques au profit des systèmes à base de tableau noir [HEN 03].

Les systèmes de pilotage automatique de traitements [THO 95] sont basés sur l'utilisation de connaissances stratégiques combinées avec l'utilisation de règles de production. Ces règles de production permettent la modification de la stratégie en fonction de l'évaluation des résultats obtenus [THO 95]. Certains travaux ont adoptés cette approche pour le traitement des documents graphiques [CLA 03].

2.4 Introduction et positionnement de notre système

Dans cet article nous présentons un système contextuel de reconnaissance structurelle de symboles appliqué aux documents graphiques. Ce système est basé sur une approche de pilotage automatique de traitements par inférence. Il exploite une librairie de traitements composés d'extracteurs de primitives graphiques, et de traitements d'évaluation contextuelle. Les connaissances factuelles, stratégiques, et descriptives dans ce système ont été structurées selon une OPM. Cette structuration originale des connaissances permet au système d'accroître ses performances de reconnaissance. Nous présentons ce système dans les deux sections suivantes. Nous l'illustrons par la suite à travers le cas d'usage.

3 Librairie de traitements

Nous présentons sommairement dans cette section les différents traitements manipulés par notre système [ADA 99] [HER 00] [DEL 03] [DEL 04]⁴. Ces traitements se distinguent selon leur fonction : « pré-traitements / extraction / reconnaissance » ou « évaluation ». De même, ils se répartissent en trois catégories selon le type des données manipulées : images, primitives graphiques structurelles, primitives graphiques statistiques. Nous les présentons dans les trois sous-sections suivantes.

3.1 Images

Nous utilisons différents pré-traitements d'images, et traitements associés d'évaluation contextuelle du bruit. Nos pré-traitements se répartissent en deux catégories : pré-traitements morphologiques [DEL 03] (érosion, dilatation, ouverture, fermeture), et filtres de composantes connexes [DEL 04] (utilisateur ou automatique, sur la forme et/ou le fond). Nous utilisons deux traitements associés d'évaluation contextuelle pour le bruit morphologique (rupture, fermeture et sur-connexion de composantes) et le bruit impulsif (ajout de composantes formes et fonds). Nous utilisons un estimateur d'épaisseur moyenne des composantes connexes, basé sur l'analyse de la carte d'intériorité

³ Système(s) Multi Agent

⁴ Nous reportons le lecteur à ces références pour une présentation détaillée de ces traitements.

produite par un squelettiseur [DEL 03]. Cet estimateur nous permet de détecter, sous contraintes d'échelles, les éventuels bruits d'érosions et de dilatations des images. Nous utilisons aussi une méthode de détection du bruit impulsionnel [DEL 04]. Celle-ci est basée sur la détection de rapports de surface disproportionnés entre composantes connexes.

3.2 Primitives structurelles

Nous utilisons différents extracteurs de primitives structurelles de types régions et de type ligne. Ces différents extracteurs sont basés sur l'utilisation de notre librairie de modélisation de primitives graphiques⁵.

Nous utilisons une librairie d'extracteurs permettant la multi-représentation en graphes de régions [DEL 04]. Celle-ci permet l'extraction des composantes et des occlusions des symboles, ainsi que leurs relations d'inclusion et de voisinage. Les occlusions correspondent aux composantes « fonds » des symboles. Il est ainsi possible d'extraire différents types de graphes selon le problème de reconnaissance envisagé. La FIG. 1 présente un exemple de symbole (a), avec ses occlusions (b), et son graphe de régions (c). Celui-ci représente la composante et ses occlusions, avec leurs relations d'inclusion combinées avec les relations de voisinage entre occlusions.

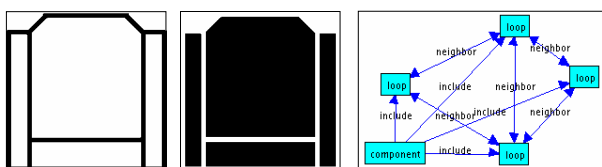


FIG. 1 - (a) symbole (b) occlusions (c) graphe de régions

Nous utilisons une librairie d'extracteurs permettant la multi-représentation en graphes de lignes [DEL 03]. Celle-ci permet la combinaison de différentes approches (squelettisation, détection de contours, segmentation en plages et régions, vectorisation directe, simplification d'objets). Dans l'objectif d'exploiter cette librairie dans un système à base d'OPM, nous l'avons rendue la plus granulaire possible. La FIG. 2 présente un exemple de symbole (a), et la représentation graphique de son graphe de squelette (b) après simplification des contours circulaires [DEL 03].

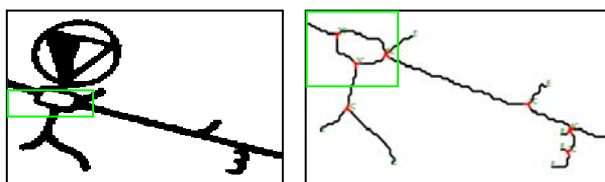


FIG. 2 - (a) symbole (b) graphe de squelette

Les graphes extraits sont ensuite exploités par un algorithme d'appariement de graphes inexacts [HER 00]. Cet algorithme calcule un critère de similarité basé sur le recouvrement entre le graphe candidat et un graphe modèle, correspondant à leur plus grand sous-graphe

commun. Le résultat de la classification correspond au label du graphe modèle de similarité maximum. Nous utilisons également ce classifieur comme traitement d'évaluation contextuelle. La similarité maximum est alors utilisée comme mesure de qualité du graphe extrait.

3.3 Primitives statistiques

Nous utilisons différents extracteurs de primitives statistiques (Fourier-Mellin, sondes circulaires, moments de Zernike, caractéristiques géométriques) [ADA 99] [DEL 04] afin de décrire les composantes des symboles et/ou leurs occlusions. Ces primitives sont par la suite utilisées par un classifieur k plus proches voisins pour la reconnaissance. Celui-ci permet l'utilisation de différentes distances, et la normalisation des primitives. Ce classifieur est également utilisé comme traitement d'évaluation contextuelle. La distance de classification est utilisée comme mesure de qualité des primitives extraites.

4 Système contextuel à base d'OPM

Nous présentons dans cette section notre système. Nous présentons premièrement dans la sous-section (4.1) notre système contextuel de contrôle par mécanisme inférence. Dans la sous-section (4.2), nous présentons ses aspects connaissances. Finalement, dans la sous-section (4.3), nous présentons la structuration OPM employée.

4.1 Système de contrôle par mécanisme d'inférence

Notre système de contrôle baptisé rsOPM⁶ permet le pilotage automatique de traitements par mécanisme d'inférence à partir d'une base de règles. De cette façon rsOPM est générique et adaptatif, la définition et la mise en œuvre d'une application de reconnaissance se fait entièrement via sa base de règles et son exploitation dans rsOPM. Au cours de ce processus, un OPD⁷ est construit formalisant l'OPM de l'application (section 4.3). rsOPM est basé sur l'utilisation de deux types d'opérateurs, l'opérateur de contrôle, et l'opérateur d'interface.

L'opérateur de contrôle constitue l'élément central de rsOPM. Il permet d'inférer la base de règles et de charger dynamiquement les traitements via leurs opérateurs d'interface. Il exploite le moteur Mandarax⁸ basé sur l'utilisation du langage RuleML⁹ [WAG 02]. Ce langage est basé sur l'utilisation de différents types de règles [WAG 02], dont les prédicats (ou règle de déduction). Le moteur est donc régi selon un triplet $\{F, P, Q\}$ ou (F) constitue les faits, (P) les prédicats, et (Q) la requête. (F) constitue ici les connaissances contextuelles du système extraites par les différents traitements d'évaluation, et donc les variables d'entrée des prédicats (P) . (P) constitue la stratégie et les règles de production employées dans le système (section 2.3). Le résultat de l'évaluation des prédicats (par le moteur d'inférence) suite à la requête (Q) détermine le traitement à exécuter. Au cours de ce processus

⁶ Rules based System for OPM : <http://site.voila.fr/mdhws/>

⁷ Object-Process Diagram [DOR 02]

⁸ <http://sourceforge.net/projects/mandarax>

⁹ Rule Markup Language

⁵ Graphics Object Modelling Library : <http://site.voila.fr/mdhws/>

d'inférence, un OPD est construit formalisant l'OPM de l'application (section 4.3). La priorité entre les règles est définie selon leur ordonnancement dans la base. Les prédicats utilisés dans rsOPM sont définis selon un quadruplet modulaire $\{P, O, S, R\}$, avec $\{P\}$ le traitement à exécuter, $\{O\}$ l'ensemble des objets sur lesquels s'exécute le traitement, $\{S\}$ l'ensemble des paramètres, et $\{R\}$ l'ensemble des règles inter-opérables avec ce traitement pour la mise à jour de la base.

Les opérateurs d'interface encapsulent (selon l'application visée) différents traitements (ou chaîne de traitements). Ils les mettent ainsi en œuvre, les rendant inter-opérables. Ils contrôlent également selon les triplets $\{O, S, R\}$ définis précédemment les objets produits et utilisés par les traitements, ainsi que les interactions de ces traitements avec la base de règles. Enfin, ces opérateurs d'interface régissent l'interaction du traitement avec l'OPD construit par l'opérateur de contrôle.

4.2 Représentation et acquisition des connaissances

Nous présentons dans cette sous-section les aspects connaissances dans notre système d'un point de vue représentation et acquisition.

Notre système utilise pour une représentation homogène à tous niveaux (traitements et système de contrôle) le langage XML. Nous utilisons deux sous-langages XML en plus de la représentation XML racine : RuleML⁹ dans notre moteur de contrôle, et XGMML¹⁰ pour la description des graphes utilisés par nos traitements.

Pour l'acquisition nous utilisons premièrement deux interfaces pour l'apprentissage de nos outils de reconnaissance. La première baptisée XMLgml¹¹ permet un apprentissage par l'exemple, basé sur différents modèles graphiques statistiques et/ou structurels (section 3). L'utilisateur peut rechercher (et labelliser) les formes similaires sur une image. Elle est basée sur un principe d'apprentissage incrémental, l'utilisateur peut à tout moment analyser statistiquement son apprentissage et le remettre en cause. Cependant XMLgml ne permet pas la combinaison de modèles graphiques. Afin d'assurer cette combinaison, nous utilisons une interface complémentaire, ojgBE¹² [DEL 04]. Cette interface permet l'édition de base de graphes par l'utilisateur. Il est ainsi possible de modifier les graphes acquis en première instance par XMLgml, pour les combiner entre eux ou les enrichir à l'aide d'informations statistiques.

Nous utilisons finalement l'interface modulaire TextPad¹³ pour l'acquisition des bases de règles de notre moteur de contrôle rsOPM (section 4.1). En effet, nous avons conçu rsOPM comme outil de développement, de façon à faciliter la conception d'application à travers l'exécution du moteur d'inférence et le chargement automatique de traitements.

4.3 Structuration OPM

L'OPM¹ [DOR 02] est une approche pour la représentation de connaissances liées à la construction d'objets via des traitements. Cette approche se base sur un formalisme de graphes attribués : les OPD⁷. Différentes relations structurelles peuvent être exprimées entre objets et/ou traitements comme : l'attribution, l'agrégation, l'héritage, etc. De ce point de vue, toute application de contrôle de chaîne de reconnaissance de forme peut être formalisée selon un OPD. Cependant, seules les applications, présentant une stratégie de construction « réelle », peuvent se prévaloir de cette approche. Les OPD permettent une structuration unifiée des connaissances d'un système de reconnaissance. Les objets et traitements associés représentent les connaissances descriptives et stratégiques. Les connaissances factuelles sont représentées via les différents stades de construction des objets. Cette représentation unifiée offre un avantage : l'organisation stratégique des connaissances descriptives. Ceci permet une reconnaissance progressive, à travers les différents stades de construction des objets, procédant ainsi du plus simple au plus complexe. Les connaissances factuelles sont alors utilisées pour piloter la reconnaissance et le processus de construction des objets.

Les connaissances utilisées dans notre système sont implicitement formalisées selon un OPD global lié à l'application de reconnaissance. Cette formalisation se traduit à travers la fragmentation de nos bases d'apprentissage, et l'interaction entre traitements via la base de règles utilisée. Notre système de contrôle rsOPM restitue localement cette OPD pour chaque cas de reconnaissance du système. La construction de l'OPD est définie dans chaque opérateur d'interface (section 4.1). Nous illustrons la formalisation OPM d'une application de reconnaissance dans le cas d'usage de la section (5) suivante.

5 Cas d'usage

Nous présentons ici des résultats obtenus sur un jeu test de symboles du concours de reconnaissance GREC2003¹⁴, afin d'illustrer l'intérêt de notre approche contextuelle à base d'OPM. Le problème de ce concours concerne la reconnaissance de symboles architecturaux (A) et électriques (E) segmentés. Différents jeux de tests sont disponibles¹⁴ 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 test de 20 classes de symboles (FIG. 3). Nous avons utilisé une base de 900 images réparties en 9 tests (de 100 images chacun) d'images binaires dégradées selon différents types et niveaux de bruit (degrad-level2-m1 à degrad-level2-m9¹⁵). Nous avons estimé (1) deux types de bruit morphologique sur ces images¹⁶ (FIG. 4). Cette

¹⁰ eXtensible Graph Markup and Modeling Language

¹¹ XML graphics model learning : <http://site.voila.fr/mdhws/>

¹² open java graph Base Editor : <http://site.voila.fr/mdhws/>

¹³ <http://www.textpad.com/>

¹⁴ Graphics Recognition 2003 : <http://www.cvc.uab.es/grec2003/>

¹⁵ Dans la suite, nous nommons ces tests degrad1, degrad2, etc.

¹⁶ Nous reportons le lecteur à la page du Contest pour une présentation détaillée des modèles de bruit employés.

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 m communs (pmc) entre les images tests et modèles, et le nombre de pixels m de l'image test (pm). m représente les pixels « formes » pour le bruit de dilatation, et les pixels « fonds » pour le bruit d'érosion. Ainsi les 9 tests appartiennent à trois catégories, les tests sur des images non bruitées morphologiquement (1 et 2), et bruitées par dilatation (3 à 6) et par érosion (7 à 9).

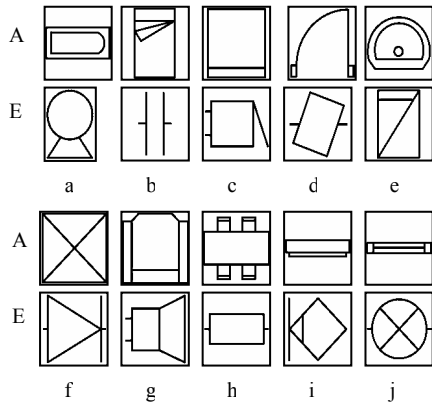


FIG. 3 - Jeu de symboles

$$\hat{E}_d(it, im) = 1 - pmc(it, im) / pm(it) \quad (1)$$

Dans une première partie nous avons mis en œuvre notre système afin de réaliser un pré-traitement contextuel de nos images. Celui-ci est basé sur des traitements d'évaluation de deux types de bruit : impulsif et morphologique (section 3.1). Ces évaluations sont réalisées par des opérateurs et retranscrites sous forme de faits. Le système exploite ces faits et auto-génère selon une base de règles de pré-traitement son scénario. Il exploite pour cela différents opérateurs basés sur des traitements morphologiques et des traitements de filtrage de composantes (section 3.1).

Dans une deuxième partie nous exploitons ces images pré traitées pour la reconnaissance des symboles. Nous utilisons une approche statistico-structurale basée sur le graphe de régions de la FIG. 1. Ce graphe de régions est enrichi par la suite par les résultats de la reconnaissance statistique des occlusions des symboles. Cette reconnaissance statistique est basée sur des caractéristiques géométriques 'surface, périmètre, compacité' (section 3.3). Nous avons utilisé nos différentes interfaces d'apprentissage de façon à constituer nos bases statistiques et structurales. Nous avons réalisé un apprentissage « exact » (sur les images modèles seules). La FIG. 4 illustre l'évaluation de la reconnaissance contextuelle à base d'OPM par comparaison avec une méthode directe. La reconnaissance directe exécute séquentiellement les différentes étapes de reconnaissance (extraction du graphe de régions, reconnaissance statistique puis structurale). La reconnaissance contextuelle suit l'OPD partiellement décrit sur la FIG. 5. Elle procède d'abord à une reconnaissance structurale des symboles. A ce stade du processus de construction, les graphes reconnus sont

répartis en quatre objets : Sym1 (Aa, Ae, Ah, Eb, Ee), Node01 (Ec, Ed, Ef, Eh), Node02 (Ac, Ad, Ai, Ea, Eg, Ei), Node04 (Ab, Af, Ag, Aj, Ej). L'objet Sym1 regroupe les graphes de symboles directement reconnaissables structurellement. Les autres objets regroupent les graphes de symboles (composés respectivement de 1, 2, et 4 occlusions) enrichis statistiquement. Cette reconnaissance statistique correspond dans l'OPD à une spécialisation (∇) des graphes extraits en graphes statistico-structuraux. Les résultats de la reconnaissance structurale sont utilisés pour piloter la construction de l'objet à travers l'OPD.

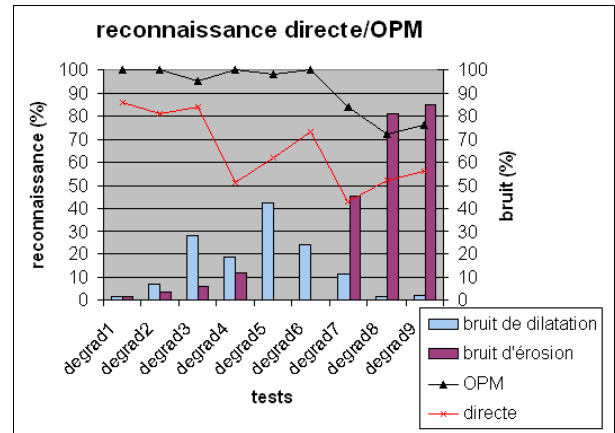


FIG. 4 - Reconnaissance directe/OPM

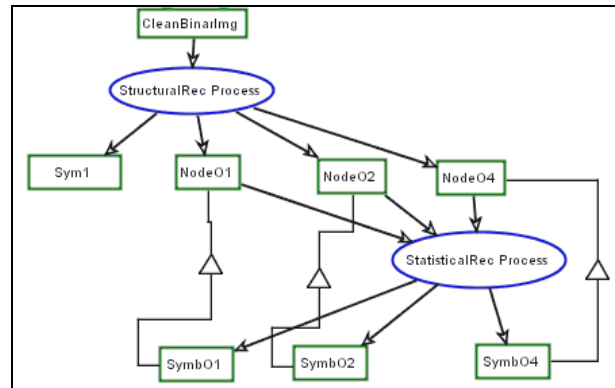


FIG. 5 - OPD partiel du processus de reconnaissance¹⁷

A travers cet OPD, les connaissances descriptives ont été organisées stratégiquement en différentes bases structurales et statistiques. Cette organisation stratégique fragmente la connaissance descriptive dans le système, simplifiant le problème de reconnaissance (FIG. 4). Pour les tests 1 à 6, les résultats sont de 98.8% alors que les primitives statistiques utilisées sont élémentaires et leur l'apprentissage exact. Cette approche est cependant bornée par la précision de la première étape de reconnaissance (ou construction) des graphes de régions [DEL 04]. Pour les tests 1 à 6, cette reconnaissance est de 99.7%. En effet, les composantes connexes et leurs occlusions sont des primitives graphiques simples à extraire. Cependant, celles-ci sont sensibles au bruit d'érosion important. Cette sensibilité est la principale cause de chute des résultats pour les tests 7 à 9.

¹⁷ Nous respectons ici la convention graphique anglaise de [DOR 02].

6 Conclusion et perspectives

Dans cet article, nous avons présenté un système contextuel de reconnaissance structurelle de symboles basée sur une OPM. Celui-ci exploite une librairie de traitements composée d'extracteurs de primitives graphiques essentiellement structurelles, et de traitements d'évaluation contextuelle. Un système de contrôle basé sur un moteur d'inférence pilote automatiquement ces traitements durant le processus de reconnaissance, s'adaptant ainsi au contexte. Les connaissances factuelles, stratégiques, et descriptives du système sont structurées selon une OPM fonction l'application de reconnaissance. Cette structuration originale des connaissances permet au système d'accroître ses performances de reconnaissance. Nous illustrons cette approche et son intérêt à travers des résultats sur un jeu test de symboles du concours GREC2003.

En terme de perspectives nous souhaitons premièrement démontrer l'adaptabilité de notre système en l'appliquant à d'autres cas de reconnaissance de symboles, et à des cas de traitement de documents graphiques. Nous souhaitons pour cela utiliser des primitives graphiques de type ligne afin d'étendre les possibilités de combinaison de notre système. Finalement, nous souhaitons exploiter l'OPD actuellement généré par notre système afin d'évaluer le processus global de construction d'objets. Cette évaluation permettrait de remettre en cause les décisions prises par le système à la vue du résultat global, et de redémarrer le processus de construction à une étape antérieure.

Références

- [ADA 99] S. Adam, J.M. Ogier, C. Cariou, J. Gardes, Y. Lecourtier, Combination of Invariant Pattern Recognition Primitive on Technical Documents, *Graphics Recognition (GREC)*, 1999.
- [ABL 00] S. Ablameyko, T.P. Pridmore, Machine Interpretation of Line Drawing Images, *Springer-Verlag*, 2000.
- [BLO 95] D. Blostein, H. Fahmy, A. Grbavec, Issues in the Practical Use of Graph Rewriting, *Graphics Recognition (GREC)*, 1995.
- [CLA 03] E. Clavier, G. Masini, M. Delalandre, M. Rigamonti, K. Tombre, J. Gardes, DocMining: A Cooperative Platform for Heterogeneous Document Interpretation According to User-Defined Scenarios, *Graphics Recognition (GREC)*, 2003.
- [DEL 03] M. Delalandre, Y. Saidali, J.M. Ogier, E. Trupin, Adaptable Vectorisation System Based on Strategic Knowledge and XML Representation Use, *Graphics Recognition (GREC)*, 2003.
- [DEL 04] M. Delalandre, E. Trupin, J.M. Ogier, Système de Reconnaissance Structurelle de Symboles, Basé sur une Multi Représentation en Graphes de Régions, *Conférence Internationales Francophone sur l'Écrit et le Document (CIFED)*, 2004.
- [DOR 02] D. Dori, Object-Process Methodology, a Holistic Systems Paradigm, *Springer-Verlag*, 2002.
- [HAN 02] E. Hancock, R. Wilson, Graph-Based Methods for Vision: A Yorkist Manifesto, *Structural and Syntactical Pattern Recognition (SSPR)*, 2002.
- [HEN 03] T.C. Henderson, L. Swaminathan, Agent Based Engineering Drawing Analysis, *Symposium on Document Image Understanding Technology (SDIUT)*, 2003.
- [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.
- [LLA 01] J. Lladós, E. Martí, J.J. Villuanueva, Symbol Recognition by Error Subgraph Matching Between Region Adjacency Graphs, *Pattern Analysis and Machine Intelligence (PAMI)*, 23(10) : 1137-1143, 2001.
- [OGI 00] J. M. Ogier, De l'Image au Document Technique, Problèmes d'Interprétation, *Habilitation à Diriger les Recherches de l'Université de Rouen*, 2000.
- [ROS 98] C. Rosenberg, J.M. Ogier, C. Cariou, K. Chehdi, A Statistical Solution to Evaluate Processing Techniques, *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, 1998.
- [SAI 03] Y. Saidali, S. Adam, J.M. Ogier, E. Trupin, J. Labiche, Knowledge Representation and Acquisition for Engineering Document Analysis, *Graphics Recognition (GREC)*, 2003.
- [THO 95] M. Thommat, S. Moisan, Knowledge Based System for Program Supervision, *Knowledge based Systems for the re-Use Program Libraries*, 1995.
- [WOR 01] M. Worring, B. Wielinga, F. Verster, A. Anjewierden, L. Todoran, S. Kabel, R. de Hoog, Automatic Indexing of Text and Graphics in Technical Manuals, *International Conference on Multimedia and Expo (ICME)*, 2001.
- [WAG 02] G. Wagner, How to Design a General Rule Markup Language, *XML Technologien für das Semantic Web (XSW)*, 2002.