

Analyse des documents graphiques :
une approche par reconstruction d'objets

15 juin 2007

Chapitre 1

Cas d'usage

1.1 Introduction

Au cours des chapitres précédents nous avons présenté notre approche de reconstruction d'objets. Nous avons introduit tout d'abord, dans le chapitre (??), nos différents opérateurs d'extraction de primitives graphiques. Ceux-ci sont basés sur les approches contour, région, et squelette. De cette façon, ils permettent de multiples cas de reconstruction d'objets en raison de leurs différences de niveaux d'extraction. Dans le chapitre (??), nous avons présenté notre système de reconstruction d'objets basé sur l'utilisation de ces opérateurs. Au sein de ce système, nos différentes connaissances sont structurées selon une méthodologie de construction d'objets (formalisée sous la forme d'OPD) en fonction du cas d'usage envisagé.

Dans ce chapitre nous présentons quelques cas d'usage de notre système de reconstruction d'objets. Le but de cette présentation est d'illustrer comment notre approche permet de simplifier le problème d'analyse des documents, et d'accroître les performances de traitement. Cette présentation a également pour but d'illustrer les propriétés de généricité et d'adaptabilité de notre système. Nos cas d'usage concernent, plus particulièrement, la reconnaissance de symboles sur des images de documents graphiques. Nous présentons un premier cas d'usage dans la section (1.2) de reconnaissance de symboles sur des plans d'itinéraire France Télécom. Nous y illustrons comment, la combinaison d'opérateurs par approche région/contour, est particulièrement adaptée pour la reconnaissance des symboles sur ce type de plan. Le deuxième cas d'usage, présenté dans la section (1.3), concerne le concours de reconnaissance de symboles GREC2003. Nous montrons comment, une taxinomie de représentations, permet d'assurer la reconnaissance de ces symboles à l'aide de simples primitives géométriques. Finalement, dans la section (1.4) nous concluons sur l'intérêt de notre approche sur ces cas d'usage.

Afin de traiter ces parties graphiques nous procédons tout d'abord à leur pré-traitement. Ce dernier est basé sur les travaux de [Lassaulzais 97] et [Adam 01]⁷. Il se déroule en quatre étapes : scannérisation, tuilage, binarisation et séparation texte/graphique. Les plans ont tout d'abord été scannés en 256 niveaux de gris avec une résolution de 300 ppp (points par pouce⁸)⁹. Les images des plans A0 ont ensuite été tuilées de manière à obtenir des extraits de plan de dimension 30 cm × 30 cm (figure (1.1)). Ceux-ci ont par la suite été prétraités par deux algorithmes : binarisation de [Otsu 79] puis filtrage de composantes connexes pour la séparation texte/graphique. La figure (1.2) donne un exemple de résultat de binarisation (a) de la figure (1.1) puis de filtrage de composantes connexes (b).

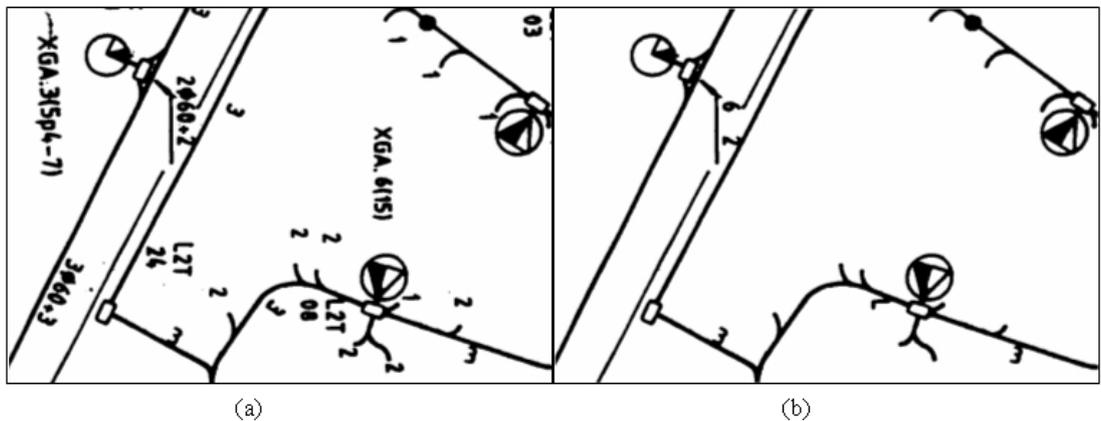


FIG. 1.2 – (a) binarisation (b) filtrage de composantes connexes

Nous procédons ensuite à la reconnaissance des symboles des plans d'itinéraire à partir des images prétraitées. Il y a trois classes de symboles sur ces plans [Adam 98] : la classe (*chambre*) et deux classes de Point de Concentration (*PC*). La chambre est un élément fournissant un accès aux câbles du réseau (les parties linéaires), tandis que les PC permettent la distribution des lignes chez les abonnés. La figure (1.3) donne des exemples de symboles appartenant à ces trois classes.

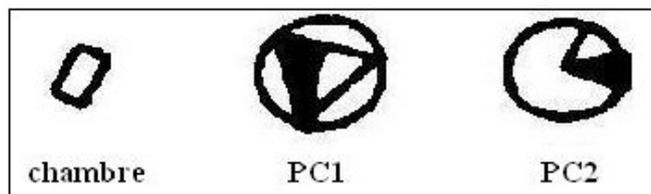


FIG. 1.3 – Symboles FT

⁷ Nous les introduisons sommairement ici et reportons à [Lassaulzais 97] et [Adam 01].

⁸ 1 pouce = 2.54 cm

⁹ dpi (dot per inch)

Pour la reconnaissance de ces symboles nous avons choisi d'exploiter une approche région. Cette dernière est basée sur l'extraction des graphes de voisinage des occlusions des symboles. En effet, ces symboles présentent des structures d'occlusions remarquables ce qui les rend particulièrement adaptés à ce type d'approche. De plus, l'utilisation de cette approche permet de s'affranchir du problème de connexion des symboles aux parties linéaires sur les plans. La figure (1.4) donne un exemple de résultat de notre approche avec l'extraction des occlusions (b) d'une image (a), et la construction du graphe de voisinage de ses occlusions (c). Notre reconnaissance de symboles se décompose alors en trois étapes principales : l'extraction et le filtrage des occlusions, la construction et l'exploitation des graphes de voisinage des occlusions, puis l'appariement des graphes extraits. Nous présentons préalablement dans la sous-section (1.2.2) la base d'images tests utilisée pour évaluer notre approche. Nous présentons ensuite chacune de nos trois étapes dans les sous-sections (1.2.3), (1.2.4) et (1.2.5). Nous présentons alors dans la sous-section (1.2.6) comment nous mettons en oeuvre ces différentes étapes au travers de trois stratégies de reconstruction d'objets. Finalement dans la sous-section (1.2.7) nous concluons.

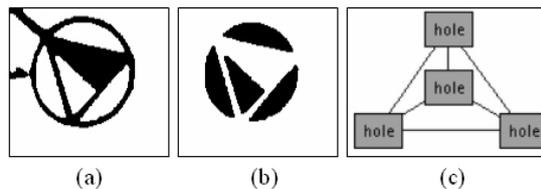


FIG. 1.4 – (a) image (b) occlusions (c) graphe de voisinage

1.2.2 Base d'images tests

Pour évaluer notre approche nous avons constitué une base d'images tests. Celle-ci se compose d'une quinzaine d'extraits de plan d'itinéraire. Les images de cette base ont d'abord été prétraitées. Nous avons ensuite utilisé notre interface d'annotation¹⁰ afin de saisir la vérité terrain. Le tableau (1.1) résume les caractéristiques de cette base : taille et distribution des classes de symbole $\{chambre, PC1, PC2\}$. À travers ces caractéristiques on voit que les plans d'itinéraire sont majoritairement composés des symboles (*chambre*) et (*PC1*).

Plans	Symboles	Occlusions
14	103	246
Chambres	PC1	PC2
54.4%	41.7%	3.9%

TAB. 1.1 – Base d'images tests

¹⁰XMLibi présentée en Annexe D de ce manuscrit.

Dans notre base nous n'avons considéré que les extraits de plan sans fragmentation de symboles. En effet, durant l'étape de tuilage des images des symboles peuvent se retrouver fragmentés entre différentes tuiles. La figure (1.5) donne un exemple de fragmentation d'un symbole (*PC1*). Dans le cadre de notre approche nous n'avons pas abordé les problématiques liées à ces cas de fragmentation [Samet 94]¹¹.

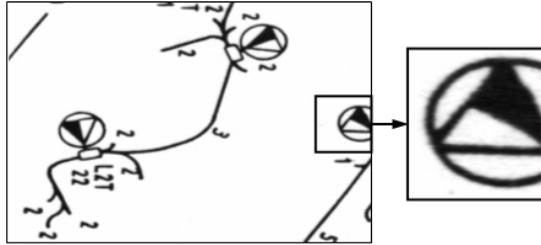


FIG. 1.5 – Cas de fragmentation d'un symbole

1.2.3 Extraction et filtrage des occlusions

L'extraction des occlusions des images des plans d'itinéraire s'appuie sur nos algorithmes de marquage de composantes connexes et d'extraction d'occlusions (chapitre (??)). Cependant, les images d'occlusions des plans d'itinéraire ne peuvent être directement utilisées pour la construction des graphes de voisinage. En effet, certaines parties graphiques sur ces images génèrent des occlusions parasites (occlusions n'appartenant pas aux symboles). Ces occlusions parasites appartiennent à différentes catégories (figure (1.6)) : occlusions de parallélisme d'un symbole (*chambre*) à une partie linéaire (a), occlusions de caractères non segmentés (b), occlusions de bouclage d'une zone d'annotation¹² (c).

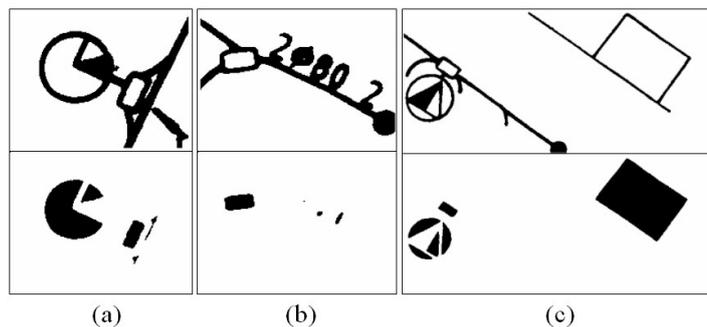


FIG. 1.6 – Cas d'occlusions parasites
(a) parallélisme chambre (b) caractères connectés (c) zone d'annotation

¹¹Nous reportons le lecteur à [Samet 94] sur ces aspects.

¹²Zone d'annotation textuelle, nous reportons le lecteur à [Adam 01] pour plus de précisions.

Cependant ces occlusions parasites peuvent être aisément distinguées des occlusions appartenant aux symboles. En effet, ces dernières ont des caractéristiques topologiques proches (figure (1.7)). De plus, les surfaces des occlusions parasites (figure (1.6)) sont soit nettement inférieures à celles des occlusions appartenant aux symboles (a) (b), ou soit nettement supérieures (c). De cette façon, les caractéristiques topologiques des composantes connexes permettent de distinguer ces différentes occlusions.

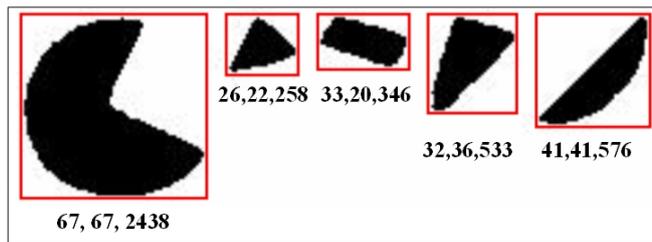


FIG. 1.7 – largeur, hauteur et surface des occlusions appartenant aux symboles

Afin de supprimer les occlusions parasites nous avons développé un opérateur spécifique de filtrage de composantes connexes. Nous avons pour cela utilisé notre algorithme de marquage de composantes connexes à partir d'un filtre¹³ passe-bande. Les paramètres de ce filtre correspondent aux plages $\{min, max\}$ utilisées pour le filtrage de caractéristiques topologiques. Ces paramètres ont été déterminés par rapport aux caractéristiques topologiques moyennes des occlusions appartenant aux symboles. Ils sont donc applicables sous contrainte d'échelle. Si au moins une caractéristique topologique d'une composante connexe n'est pas comprise dans sa plage correspondante, cette composante est alors supprimée de l'image. La figure (1.8) donne un exemple de résultat de filtrage d'occlusions (c) de l'image (b). Les zones rectangulaires sur l'image (b) indiquent les occlusions effacées de l'image durant le filtrage.

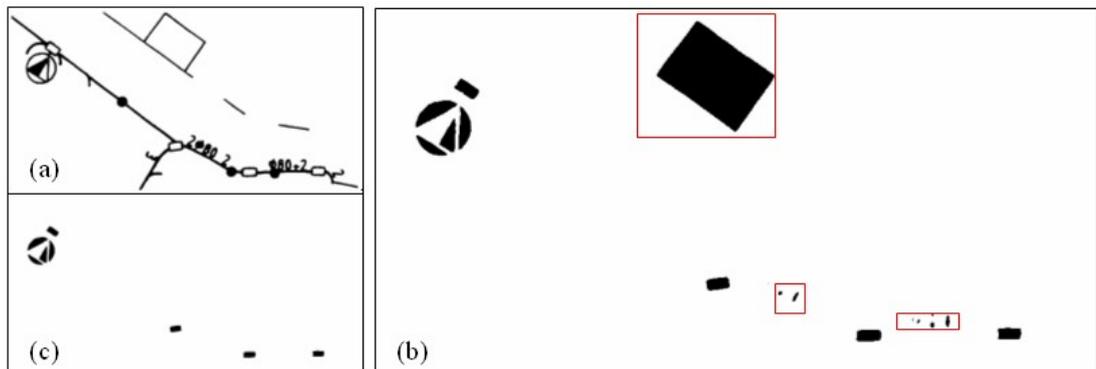


FIG. 1.8 – (a) image (b) occlusions extraites (c) occlusions filtrées

¹³L'algorithme de ce filtre est proche de celui présenté page ??, nous ne le développerons pas ici.

1.2.4 Construction et exploitation des graphes de voisinage

Une fois les occlusions appartenant aux symboles extraites nous procédons à l'étape de construction des graphes de voisinage et à leur exploitation. Nous avons utilisé pour cela notre opérateur de construction de graphes de voisinage à partir des images des occlusions. Le problème majeur de l'utilisation de cette approche pour ce cas d'usage est la proximité des symboles sur les plans d'itinéraire. La figure (1.9) présente les images et graphes de voisinage de deux cas de proximité (a) (b). On voit sur ces deux cas que la localisation des graphes de symboles est impossible. Dans le premier cas (a), de multiples configurations d'emplacement apparaissent pour la paire de symboles $\{chambre, PC2\}$. Dans le second cas (b), de multiples configurations d'emplacement apparaissent pour la paire de symboles $\{PC1, PC1\}$, mais aussi pour le triplet de symboles $\{PC1, PC1, chambre\}$.

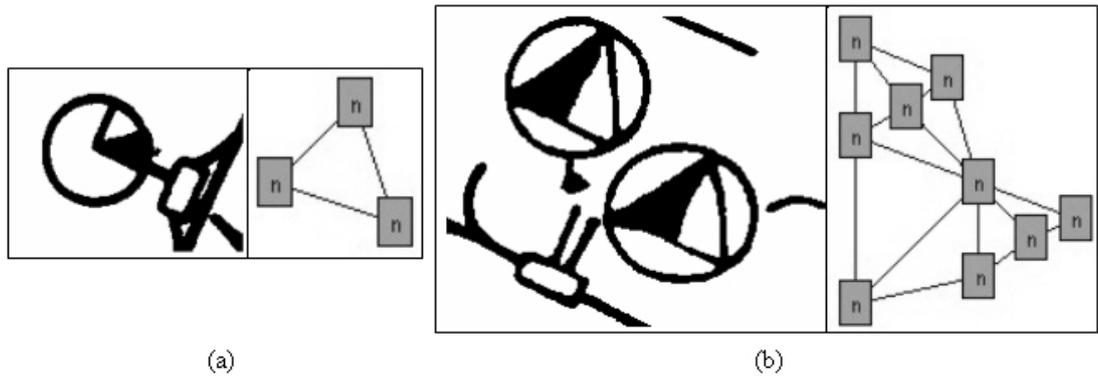


FIG. 1.9 – Cas de proximité¹⁴(a) *chambre* – *PC2* (b) *chambre* – *PC1* – *PC1*

Le tableau (1.2) présente la distribution des cas de proximité dans notre base d'images tests en fonction des classes de symbole. À travers ce tableau, on voit que ces cas de proximité affectent une large partie des symboles sur les plans (86.4%). En effet, la totalité des symboles (*PC1*) et (*PC2*) sont proches d'autres symboles, ils sont systématiquement regroupés à un symbole (*chambre*) comme le montre les exemples (a) (b) de la figure (1.10). Les symboles (*chambre*) sont les seuls symboles pouvant être isolés sur ces plans (c).

Chambres	PC1	PC2	Total
75%	100 %	100 %	86.4 %

TAB. 1.2 – Distribution des cas de proximité selon les classes de symbole

¹⁴Représentation simplifiée des graphes de voisinage : arcs non labellisés, noeuds labellisés *n*.

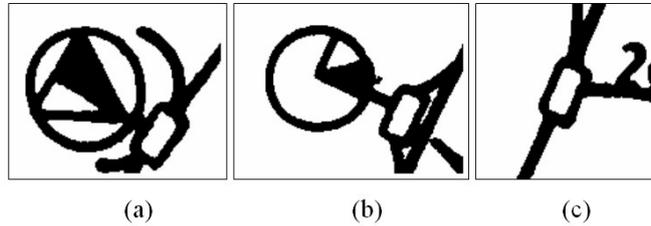


FIG. 1.10 – (a) *chambre* – PC1 (b) *chambre* – PC2 (c) *chambre* seule

Dans ce contexte, il devient impossible de reconnaître les symboles uniquement avec une approche employant les graphes de voisinage des occlusions. De façon à pouvoir traiter ces cas de proximité, nous avons employé une reconnaissance statistico-structurale des graphes de voisinage. La figure (1.11) en illustre le principe. Les deux premières étapes correspondent à la partie structurale de cette approche : extraction des occlusions (b) et construction du graphe de voisinage (c). La troisième étape correspond à la partie statistique. Durant cette étape les occlusions sont reconnues statistiquement (ici en trois classes $\{e, t, c\}$). Les résultats de cette reconnaissance sont ensuite ajoutés (d) au graphe de voisinage extrait (c).

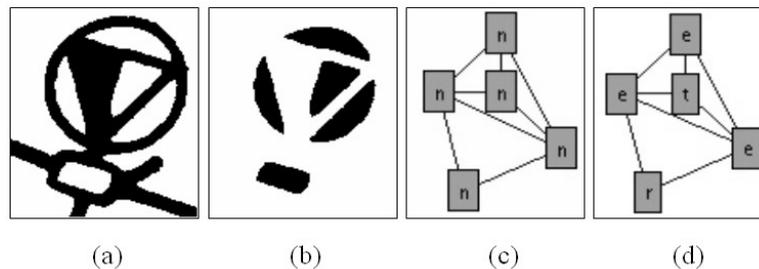


FIG. 1.11 – (a) symbole (b) occlusions (c) graphe de composantes (d) graphe de labels

La reconnaissance statistique des symboles FT concerne 5 classes d'occlusions (figure (1.12)). Pour cette reconnaissance nous nous sommes basés sur les travaux de [Adam 01]¹⁵. Nous avons utilisé un ensemble d'opérateurs¹⁶ d'extraction de caractéristiques basés sur différentes approches : moments de Zernike, invariants de Fourier-Mellin, et sondes circulaires. Ces caractéristiques permettent de décrire les occlusions de façon invariante aux changements d'échelle et d'orientation. Nous avons équitablement réparti les vecteurs extraits entre une base de test et une base d'apprentissage à l'aide de notre interface¹⁷ d'apprentissage. Ensuite, à l'aide d'un classifieur¹⁶ de type k plus proches voisins (kppv avec $k=1$) utilisant la distance euclidienne, nous avons obtenu les résultats de reconnaissance présentés dans le tableau (1.3).

¹⁵Nous introduisons sommairement ces aspects ici et reportons le lecteur à [Adam 01].

¹⁶Ces opérateurs sont présentés en Annexe C de ce manuscrit.

¹⁷XMLgml présentée en Annexe D de ce manuscrit.

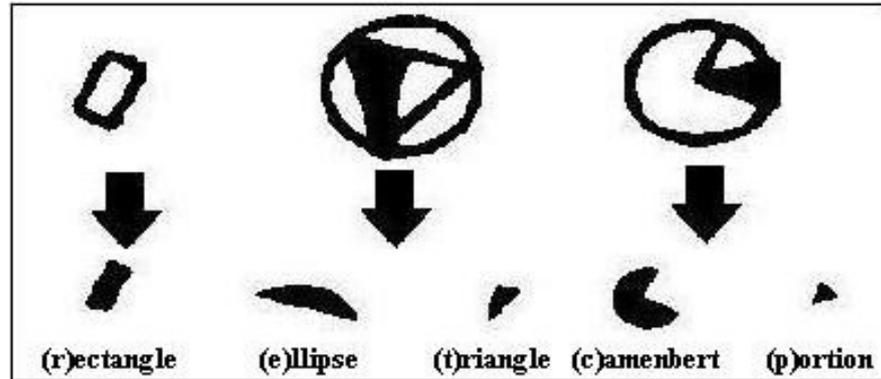


FIG. 1.12 – Classes d'occlusions des symboles

Zernike	Fourier-Mellin	Sondes
97.8 %	86.66 %	86.66 %

TAB. 1.3 – Résultats de la reconnaissance statistique des occlusions

Ces résultats démontrent que les moments de Zernike semblent les mieux placés¹⁸ pour la reconnaissance de ce type d'occlusions. Le problème de reconnaissance statistique semble donc aisé avec cette approche comme le montre les résultats élevés de reconnaissance de 97.8%.

Afin d'analyser plus précisément les résultats de reconnaissance nous avons utilisé notre interface¹⁹ d'apprentissage afin de dresser la carte des distances entre les classes d'occlusions (tableau (1.4)). Cette carte a été calculée à partir de l'extraction des moments de Zernike sur un jeu d'occlusions, et l'utilisation du classifieur kppv. Pour chaque classe d'occlusion cette carte indique les distances minimum (D_{min}), moyenne (D_{moy}) et maximum (D_{max}). Les classes d'occlusions avec lesquelles les distances sont minimum et maximum sont également indiquées. Les distances sont exprimées en % par rapport à la distance maximum calculée sur le jeu.

Classe	D_{min}	D_{moy}	D_{max}
r	60 % (t)	76 %	90 % (c)
t	60 % (r)	82 %	100 % (e)
e	53 % (c)	81 %	100 % (t)
c	53 % (e)	80 %	99 % (t)
p	67 % (t)	75 %	85 % (e)

TAB. 1.4 – Carte des distances des classes d'occlusion

¹⁸Nous reportons le lecteur à [Adam 01] pour une comparaison de ces différentes approches.

¹⁹XMLgml présentée en Annexe D de ce manuscrit.

À travers cette carte des distances, on peut identifier les cas de confusion les plus probables (D_{min}), et les moins probables (D_{max}), pour la reconnaissance des occlusions. Il s'agit des couples $\{r,t\}$ $\{e,c\}$ pour les plus probables et $\{e,t\}$ $\{t,c\}$ pour les moins probables. À la vue de la répartition des symboles sur les plans (tableau (1.1)), et la distribution des cas de proximité (tableau (1.2)), la source la plus importante de confusion concerne donc le couple d'occlusions $\{r,t\}$. En effet, les symboles $\{chambre, PC1\}$ sont systématiquement associés sur ces plans. Une confusion des occlusions $\{r,t\}$ peut donc entraîner une erreur de reconnaissance structurelle de ces deux symboles.

1.2.5 Reconnaissance structurelle de symboles

Nous exploitons les graphes précédemment construits à l'aide d'une méthode d'appariement afin de reconnaître les symboles. Cet appariement concerne la recherche de sous-graphes dans le graphe de voisinage des occlusions de l'image. Il s'agit donc d'un appariement de sous-graphe vers graphe. Les sous-graphes correspondent aux graphes modèles des symboles. Chacun des symboles comporte de 1 à 4 occlusions, et les extraits de plans comporte en moyenne une vingtaine d'occlusions (tableau (1.1)). Le problème d'appariement correspond donc à la recherche de sous-graphes modèles de tailles de 4 noeuds maximum dans des graphes d'une dimension moyenne de 20 noeuds. Les étiquettes des noeuds des graphes de voisinage des occlusions résultent des résultats de la reconnaissance statistique. Comme ces résultats peuvent être entachés d'erreurs, le problème d'appariement est inexact.

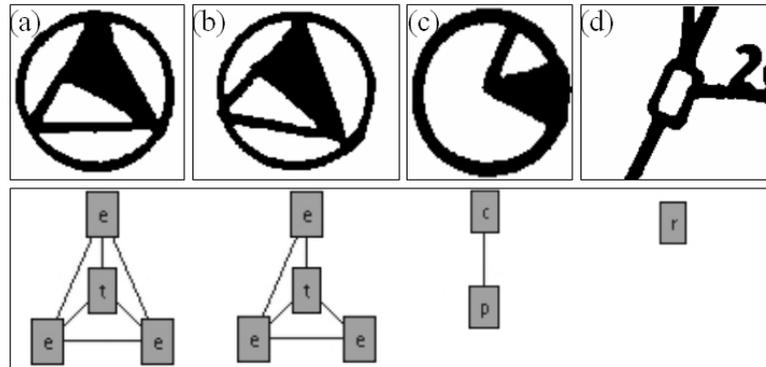
Pour notre appariement nous avons choisi d'utiliser une méthode basée sur l'algorithme²⁰ A* [Nilsson 80]. L'implémentation de cette méthode est issue des travaux de [Nicolas 02]²¹. Celle-ci permet l'appariement inexact de sous-graphes. Elle utilise des coûts d'édition, la mesure de dissimilarité entre un sous-graphe candidat et un graphe modèle se mesure par la somme de coûts d'édition. Ces coûts correspondent aux opérations nécessaires à la transformation d'un sous-graphe candidat vers un graphe modèle. Cette méthode emploie également une factorisation des sous-graphes détectés. Le but est de factoriser itérativement des sous-graphes reconnus dans un graphe plus large, afin de produire un graphe des résultats de reconnaissance.

La figure (1.13) présente la base de graphes modèles que nous utilisons pour l'appariement. Chacun des graphes modèles est présenté avec son symbole correspondant. Cette base a été constituée à l'aide de notre interface²² d'apprentissage. Dans cette base, le symbole (*PC1*) possède deux graphes modèles. En effet, ce symbole existe en deux représentations graphiques (a) (b) sur les plans. La relation de voisinage entre le couple d'occlusion $\{t, e\}$ peut être parfois présente, parfois absente. Afin de résoudre ce problème nous avons défini les deux graphes modèles correspondants.

²⁰Nous l'introduisons sommairement ici et reportons le lecteur à [Nicolas 02].

²¹Cet algorithme est inclu dans la librairie PSI présentée en Annexe C.

²²XMLgml présentée en Annexe D de ce manuscrit.

FIG. 1.13 – (a) $PC1_a$ (b) $PC1_b$ (c) $PC2$ (d) *chambre*

1.2.6 Stratégies de reconstruction d'objets

Au cours des sous-sections précédentes nous avons présenté les trois étapes principales de notre approche : extraction et filtrage des occlusions, construction et exploitation des graphes de voisinage, puis reconnaissance structurale des symboles. Au cours de cette section nous présentons comment nous mettons en oeuvre ces différentes étapes aux travers de trois stratégies évolutives de reconstruction d'objets. Les deux premières sont purement basées sur une approche région par reconnaissance statistico-structurale des symboles. La troisième étend la reconnaissance statistico-structurale à l'aide d'une approche contour. Nous y illustrons alors comment la combinaison d'opérateurs par approche région/contour est particulièrement adapté pour la reconnaissance des symboles sur ce type de plan.

Dans le cadre de la reconnaissance statistico-structurale la reconnaissance structurale des symboles dépendra en large partie de la reconnaissance statistique des occlusions. Concernant nos stratégies de reconstruction d'objets nous nous sommes plus particulièrement intéressés aux interactions existantes entre la reconnaissance statistique et structurale. Nous avons, pour cela, étudié la capacité de la reconnaissance structurale à pallier aux résultats de la reconnaissance statistique. Pour ce faire, nous avons tout d'abord volontairement bruité notre base d'apprentissage statistique. Ce bruitage a été effectué de façon à réduire les résultats de reconnaissance d'occlusions à hauteur de 50 % environ. Il a été réalisé en supprimant aléatoirement des vecteurs modèles dans la base d'apprentissage statistique. Ensuite, nous avons défini les coûts d'édition de notre algorithme d'appariement de façon à privilégier la topologie des graphes durant le calcul de la dissimilarité. Nous avons défini²³, pour cela, un faible coût de substitution pour les noeuds par rapport aux coûts de suppression et d'insertion des noeuds et des arcs [Nicolas 02].

²³Nous reportons le lecteur à [Nicolas 02] pour plus de précisions sur ces aspects.

Le tableau (1.5) synthétise les résultats de nos trois stratégies de reconstruction à partir des bases d'apprentissage statistique non bruitée et bruitée. Dans ce tableau, les résultats de reconnaissance statistique non bruités sont notés 97.8% de reconnaissance, et ceux bruités sont notés 49% de reconnaissance. Nous détaillons nos trois stratégies de reconstruction d'objets dans la suite de cette sous-section.

Stratégies	groupement	segmentation	simplification
97.8 % de reconnaissance (non bruité)	100 %	100 %	100 %
49 % de reconnaissance (bruité)	<i>nul</i> ²⁴	54.4 %	87.5 %

TAB. 1.5 – Résultats de la reconnaissance de symboles des différentes stratégies

Notre première stratégie de reconstruction est décrite par l'OPD de la figure (1.14) (a). Celui-ci enchaîne séquentiellement à partir des images des occlusions la construction du graphe de voisinage, la reconnaissance statistique puis l'appariement de graphes. Nous qualifions cette première stratégie de groupement car elle extrait un graphe de voisinage des occlusions global à l'image comme le montre la figure (1.15) (a).

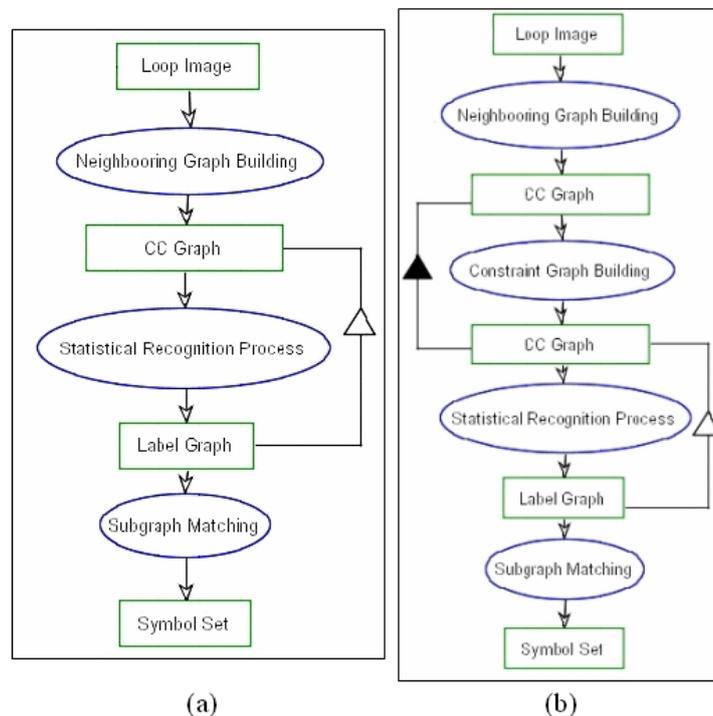


FIG. 1.14 – OPD par stratégie (a) de groupement (b) de segmentation

²⁴ non évaluable

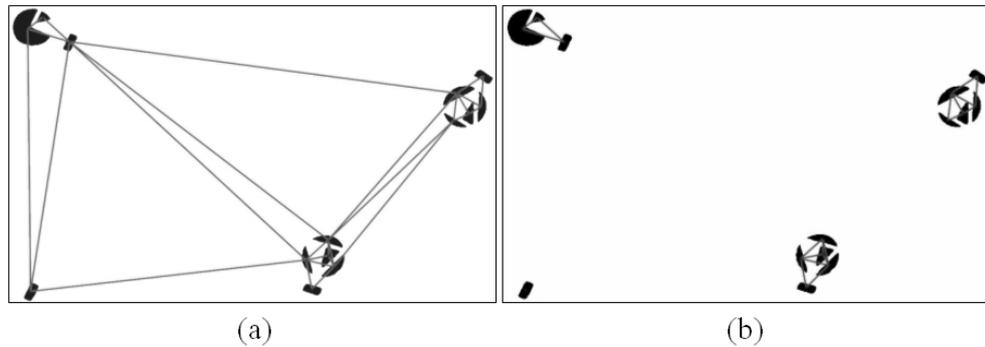


FIG. 1.15 – Résultats SVG des stratégies par (a) groupement (b) segmentation

Cette stratégie par groupement donne des résultats de reconnaissance de symboles exacts dans le cas d'une reconnaissance statistique non bruitée (tableau (1.5)). En effet, les uniques erreurs de reconnaissance statistique concernent la confusion des occlusions $\{c,e\}$ (tableau (1.4)). Ces erreurs n'affectent jamais deux fois les occlusions d'un même symbole : elles sont réparties de façon disjointe entre les différents symboles sur les plans d'itinéraire. Elles n'ont donc pas d'impact sur les résultats d'appariement de graphes. Dans le cas d'une reconnaissance statistique bruitée, la stratégie par groupement donne des résultats de reconnaissance de symboles incohérents (tableau (1.5)). Le taux de reconnaissance de 49 % rend l'appariement de sous-graphes inexploitable à cause des relations de voisinage dites distantes. Ces dernières sont illustrées sur la figure (1.16). Nous les détaillons par la suite.

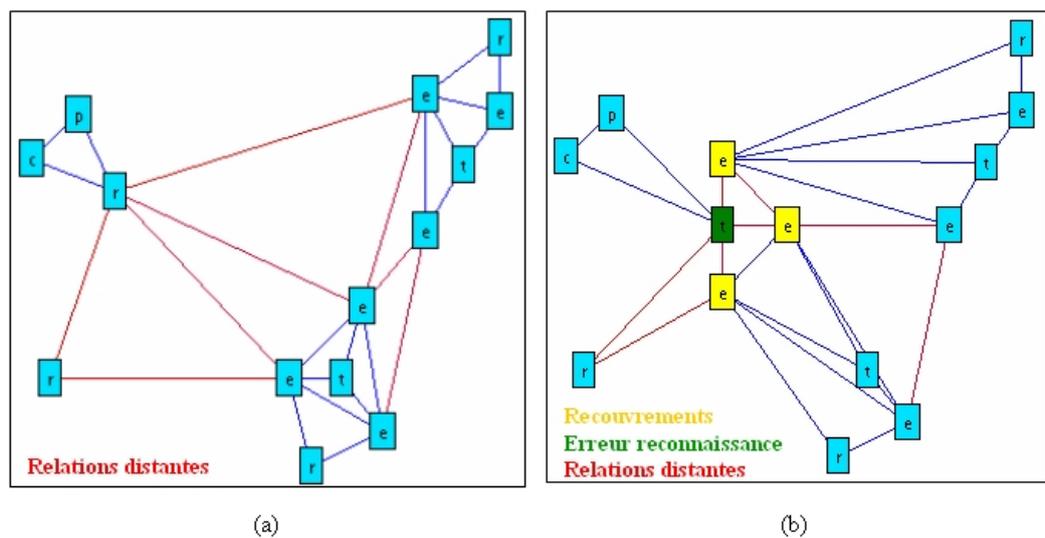


FIG. 1.16 – (a) relations de voisinage distantes (b) recouvrement de sous-graphes²⁵

²⁵Cette figure donne une représentation graphe de la figure (1.15) (a).

Ces relations de voisinage distantes peuvent provoquer des recouvrements entre sous-graphes de symboles en cas de mauvaises reconnaissances statistiques des occlusions. Ces recouvrements perturbent alors l'appariement des sous-graphes, la figure (1.16) (b) en donne un exemple. Sur cet exemple une erreur de reconnaissance statistique d'une occlusion $\{r,t\}$ provoque un recouvrement entre les sous-graphes des symboles $\{chambre, PC1, PC1\}$. Le sous-graphe de recouvrement de ces symboles correspond à un symbole $PC1$. Son appariement provoque son association aux noeuds appartenant aux autres sous-graphes. Cette association ampute alors les autres sous-graphes pouvant provoquer des erreurs d'appariement par la suite. Ce phénomène amplifié à un taux de reconnaissance de 49 % de reconnaissance statistique des occlusions rend alors l'appariement inexploitable.

Afin de résoudre ce problème nous avons développé une seconde stratégie de reconstruction d'objets (figure (1.14) (b)). Cette dernière est étendue de la stratégie par groupement via la construction d'un graphe sous contraintes de distance basée sur l'utilisation de notre opérateur. Cette stratégie est, par opposition, qualifiée de stratégie par segmentation. En effet, elle extrait différents graphes de voisinage des occlusions pour une même image de plan d'itinéraire comme le montre la figure (1.15) (b). Dans l'OPD de cette stratégie (figure (1.14) (b)), une relation d'agrégation (\blacktriangle) relie les deux objets graphe de composantes. En effet, le graphe de composantes sous contrainte de distance correspond à une partie du graphe de voisinage extrait dépourvu des relations de voisinage distantes (figure (1.16)).

Au même titre que la stratégie par groupement, cette stratégie par segmentation donne des résultats de reconnaissance de symboles exacts dans le cas d'une reconnaissance statistique non bruitée (tableau (1.5)). Dans le cas d'une reconnaissance statistique bruitée, elle donne des résultats de reconnaissance cohérents contrairement à la stratégie par groupement. Elle permet, en effet, de résoudre le problème de recouvrement des sous-graphes de symboles (figure (1.16)) par filtrage des relations de voisinage distantes. De cette façon, elle assure la reconnaissance des chambres isolés sur les plans d'itinéraire. Cependant, cette stratégie ne permet pas la segmentation des symboles dans les cas de proximité (figure (1.9)). Ceux-ci affectent une large partie (75 %) des symboles sur les plans (tableau (1.2)). En effet, la totalité des symboles ($PC1$) et ($PC2$) sont systématiquement regroupés à un symbole ($chambre$) (figure (1.10)). Ces regroupements combinés aux confusions de reconnaissance statistique (tableau (1.4)) sont la principale cause d'erreur de reconnaissance des symboles.

De façon à résoudre les problèmes de reconnaissance dans les cas de proximité nous avons développé une dernière stratégie de reconstruction d'objets dite de simplification. Cette dernière vise à effacer les symboles $\{PC1, PC2\}$ sur les images de plan de façon à simplifier la reconnaissance des symboles ($chambre$). Elle emploie pour cela une approche contour complétant la reconnaissance statistico-structurale. Elle est décrite à travers l'OPD de la figure (1.17). Elle met en oeuvre six opérateurs $\{1, 2, 3, 4, 5, 6\}$ que nous détaillons dans la suite de cette sous-section.

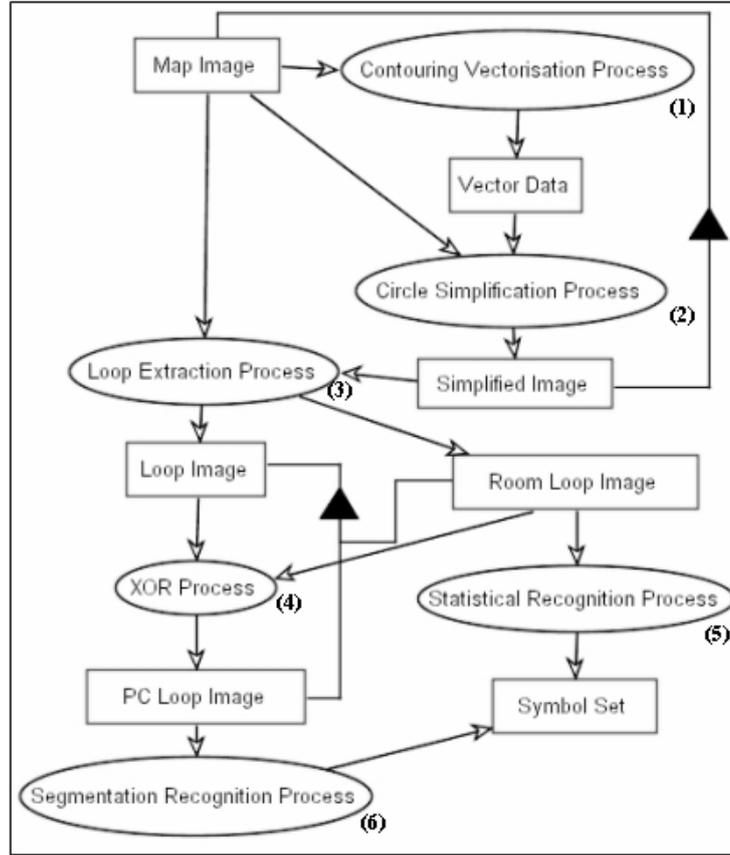


FIG. 1.17 – OPD par stratégie de simplification

Le coeur de notre stratégie de simplification est basée sur une approche de vectorisation des contours des plans d'itinéraire pré-traités. Nous avons défini pour cela un opérateur (1) de vectorisation. Ce dernier enchaîne successivement les différents opérateurs de détection de contours, d'approximation polygonale et de détection d'arcs de cercle. Celui de détection de contours est basé sur notre méthode présentée dans le chapitre (??). Les autres²⁶ sont basés sur les travaux²⁷ de [Hameau 02] et [Perin 04]. Ils emploient une polygonalisation dit de la corde [Ramer 72] suivie d'un post-traitement de type split & merge [Pavlidis 74]. La détection d'arcs de cercle est basée sur l'analyse de la variance des angles entre vecteurs deux à deux. Cette détection est paramétrée selon des plages de contraintes sur les angles (α) et longueurs (β) des arcs à détecter. Pour ce cas d'usage nous recherchons les arcs d'angles larges ($\alpha > \frac{3}{2} \times \Pi$). Les plages de longueur (β) sont déterminées, elles, en fonction de la taille moyenne des symboles rencontrés sur les plans. La figure (1.18) (b) donne un résultat de vectorisation au format SVG de l'image (a).

²⁶Ces opérateurs sont inclus dans la librairie PSI présentée en Annexe C.

²⁷Nous les présentons sommairement et reportons le lecteur à [Perin 04] pour plus de précisions.

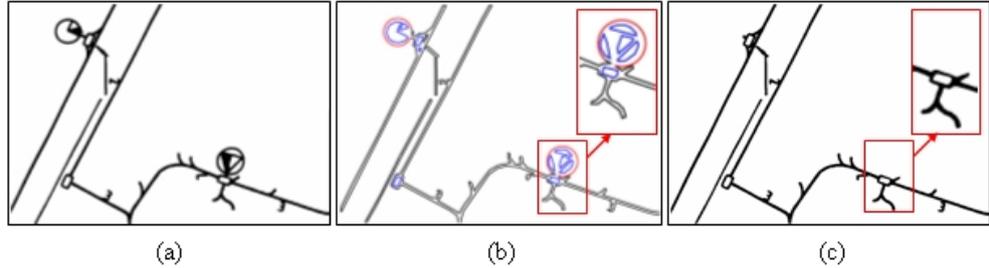


FIG. 1.18 – (a) image (b) vectorisation (c) simplification

Nous utilisons par la suite les résultats de vectorisation afin de simplifier les images de plan. Nous avons développé pour cela un opérateur spécifique **(2)** d'effacement des arcs de cercle. La méthode employée par cet opérateur est présentée dans le pseudo-algorithme (1.2.1). Elle est illustrée sur les figures (1.19) (a), (b) et (c). Elle utilise en entrée une variable (*image*) ainsi que les différentes coordonnées de l'arc détecté : centre (x_c, y_c) et point de début (x_1, y_1) et de fin (x_2, y_2) . Elle parcourt alors le rectangle englobant $\{x_0, y_0, d\}$ du cercle dans (*image*) à la recherche des pixels formes (a). Pour chacun des pixels formes elle teste si celui-ci est à une distance inférieure à (r) du centre de l'arc (x_c, y_c) . Elle recherche alors, à l'aide d'une méthode (*inclu*), si le point forme testé (x, y) est compris dans la zone d'effacement. La figure (b) donne deux exemples de point (x_a, y_a) et (x_b, y_b) respectivement inclu et non inclu. Elle recherche ensuite si un point d'intersection (x_i, y_i) existe, entre la droite (l_1) décrite par $\{x_c, y_c, x, y\}$, et (l_2) décrite par $\{x_1, y_1, x_2, y_2\}$. Elle calcule²⁸ pour cela les tangentes (a_1, a_2) ainsi que les ordonnées à l'origine (b_1, b_2) de ces deux droites. Les coordonnées (x_i, y_i) obtenues sont ensuite testées par une méthode de recouvrement afin de savoir si elles appartiennent à (l_1) . La figure (c) illustre ce test, il est basé sur la recherche de valeurs opposées des d_x et d_y entre (x_i, y_i) et les extrémités $\{x_c, y_c, x, y\}$ de (l_1) . Si (x_i, y_i) est non inclu cela signifie que (l_1) et (l_2) ne s'intersectent pas, et donc que le point (x, y) doit être effacé. La figure (1.18) (c) donne un exemple de résultat d'effacement des cercles.

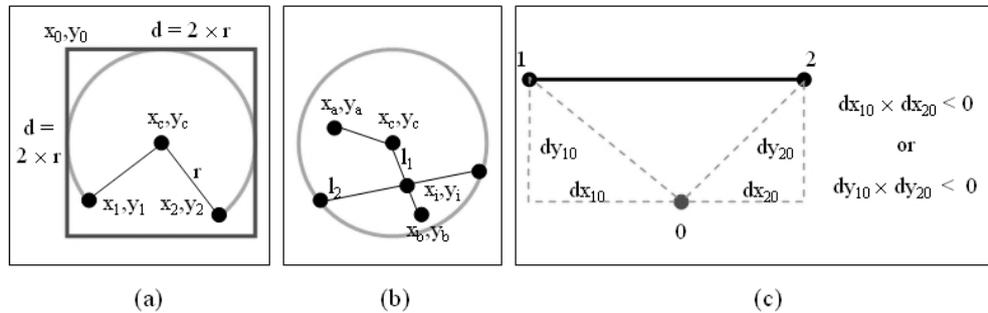


FIG. 1.19 – (a) cercle (b) zone d'effacement (c) recouvrement

²⁸Nous ne présentons pas ici les cas de parallélisme afin de clarifier la présentation de notre méthode.

Pseudo-algorithme 1.2.1: EFFACER($image, x_c, y_c, x_1, y_1, x_2, y_2$)

fonction RECOUVRE($x_0, y_0, x_1, y_1, x_2, y_2$)

```

 $dx_{10} \leftarrow x_1 - x_0$      $dy_{10} \leftarrow y_1 - y_0$ 
 $dx_{20} \leftarrow x_2 - x_0$      $dy_{20} \leftarrow y_2 - y_0$ 
si  $dx_{10} \times dx_{20} < 0$  ou  $dy_{10} \times dy_{20} < 0$ 
  alors retourner ( vrai )
retourner ( faux )

```

fonction INCLU($x_0, y_0, x_1, y_2, x_2, y_2, x_3, y_3$)

```

 $a_1 = \frac{y_1 - y_0}{x_1 - x_0}$      $b_1 = y_0 - a_1 \times x_0$ 
 $a_2 = \frac{y_3 - y_2}{x_3 - x_2}$      $b_2 = y_2 - a_2 \times x_2$ 
 $x_i = \frac{b_2 - b_1}{a_1 - a_2}$      $y_i = \frac{a_2 \times b_1 - a_1 \times b_2}{a_2 - a_1}$ 
si RECOUVRE( $x_i, y_i, x_c, y_c, x_1, y_2$ )
  alors retourner ( faux )
retourner ( vrai )

```

principal

```

 $r \leftarrow \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2}$ 
 $x_0 \leftarrow x_c - r$      $y_0 \leftarrow y_c - r$      $d \leftarrow 2 \times r$ 
pour  $y$  variant de  $y_0$  à  $y_0 + d$ 
  faire pour  $x$  variant de  $x_0$  à  $x_0 + d$ 
    faire si LIRE( $image, x, y$ ) = 1
      alors  $\begin{cases} \text{si } \sqrt{(x - x_c)^2 + (y - y_c)^2} \leq r \\ \text{alors si INCLU}(x_c, y_c, x, y, x_1, y_1, x_2, y_2) \\ \text{alors ECRIRE}(image, x, y, 0) \end{cases}$ 

```

Une fois les images simplifiées obtenues nous les exploitons à l'aide de notre opérateur d'extraction des occlusions (3). Celui-ci est utilisé en combinaison d'un opérateur²⁹ de soustraction logique (4). Nous illustrons l'action de cet opérateur sur les figures (1.20) (a), (b), (c), (d) et (e). Il est basé sur un test logique de type *ou exclusif* entre les images des occlusions (b) et (d). Ces dernières sont respectivement extraites à partir des plans initiaux (a) et des images simplifiées (c). La figure (e) donne un exemple de résultat de soustraction logique entre ces images (b) et (d).

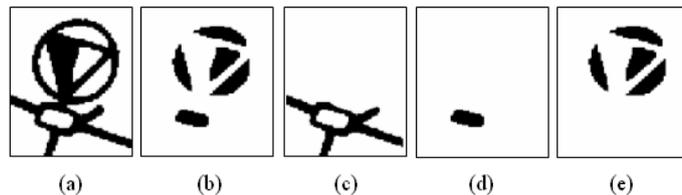


FIG. 1.20 – (a) (b) non simplifié (c) (d) simplifié (e) soustraction

²⁹Exploité de notre librairie PSI présentée en Annexe C.

Nous traitons ensuite les images des occlusions simplifiées (d) et soustraites (e) à l'aide de deux opérateurs (5) et (6). En effet ces images (d) et (e) correspondent respectivement aux occlusions des symboles (*chambre*) et (*PC*). Nous traitons les images des symboles (*PC*) à l'aide d'un opérateur (6) correspondant employant notre stratégie par segmentation (figure (1.14) (b)). Les images des symboles (*chambre*) sont traitées elles par reconnaissance statistique seule (5) : c'est à dire extraction des moments de Zernike et classification par k plus proches voisins. Ces symboles (*chambre*) sont en effet composés d'une seule occlusion ce qui n'impose pas l'utilisation d'une reconnaissance structurelle.

Notre stratégie par simplification permet ainsi une séparation des symboles (*chambre*) sur les plans par détection et suppression des cercles décrivant les symboles (*PC*). Elle utilise pour cela des opérateurs par approche contour qui viennent se combiner aux opérateurs par approche région. De cette façon il est possible de reconnaître ces deux classes de symbole par deux processus de reconnaissance des formes indépendants : un statistique et un statistico-structurel. Cette reconnaissance en deux temps permet ainsi d'accroître les résultats de reconnaissance des symboles, à partir des bases d'apprentissage statistique bruitées, de 54.4 % à 87.5 % (tableau (1.5)). Cette stratégie ne permet par contre de parer intégralement aux erreurs de reconnaissance statistique. Ce problème n'est pas inhérent à notre stratégie de reconstruction d'objets mais au pré-traitement des plans. En effet, différents caractères connectés aux symboles (*PC*) à l'issue du pré-traitement, altèrent la vectorisation des cercles et donc leur suppression. La figure (1.21) en donne un exemple.



FIG. 1.21 – Exemple de caractère connecté à un symbole

1.2.7 Conclusion

Dans cette section nous avons présenté un premier cas d'usage pour notre approche de reconstruction d'objets. Celui-ci concerne la reconnaissance de symboles sur plan d'itinéraire. Nous avons mis en oeuvre trois stratégies évolutives de reconstruction d'objets : par groupement, segmentation et simplification. Les deux premières emploient une approche région pour la reconnaissance statistico-structurelle des symboles. La troisième étend cette reconnaissance par une approche contour pour la simplification des symboles sur les images de plan. Nous illustrons alors, comment la combinaison d'opérateurs par approche région/contour, est particulièrement adaptée pour la reconnaissance de symboles sur ce type de plan. Dans la section suivante nous présentons un autre cas d'usage de reconnaissance de symboles du concours GREC2003.

1.3 Concours de reconnaissance de symboles GREC2003

1.3.1 Introduction

Nous présentons ici un second cas d'usage de notre approche de reconstruction d'objets : le concours de reconnaissance de symboles de GREC2003³⁰ [Valveny 04]. Celui-ci concerne la reconnaissance d'images de symboles architecturaux (A) et électriques (E) segmentés (figure (1.22)). Différents tests sont disponibles³¹, selon le jeu de symboles utilisé (15, 20, 40), et les types et niveaux de bruit appliqués à partir des images modèles. Plus particulièrement deux types principaux de bruit ont été utilisés³² (figure (1.23)) : binaire (a) et vectoriel (b). Les différents tests sont fournis avec des fichiers modèles décrivant la vérité terrain de façon à évaluer les résultats de reconnaissance.

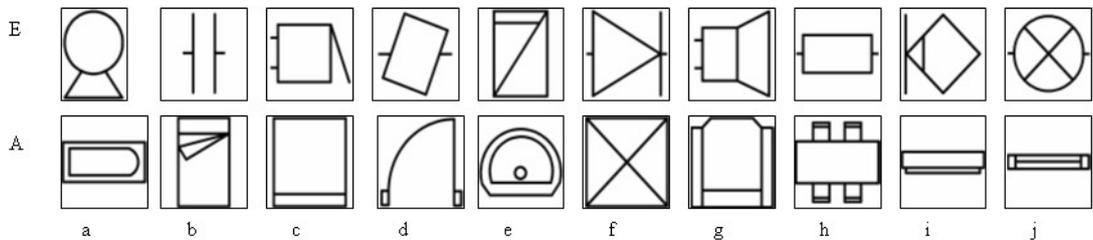


FIG. 1.22 – Jeu de symboles (20)

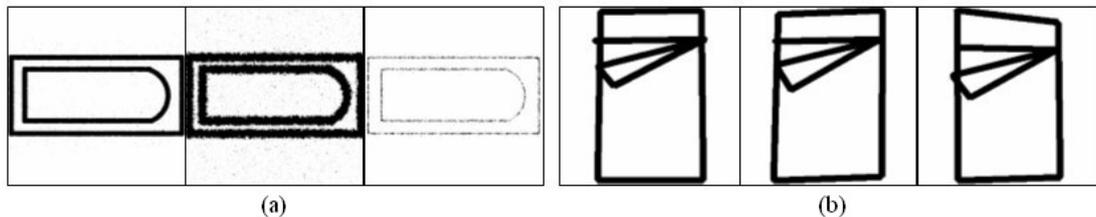


FIG. 1.23 – 3 niveaux de bruit de type (a) binaire (b) vectoriel

Pour ce cas d'usage nous avons tout d'abord basé notre approche sur une étape de pré-traitement contextuel des images bruitées, nous la présentons dans la sous-section (1.3.2). Dans la sous-section (1.3.3) nous présentons et évaluons la représentation que nous avons choisie pour décrire les symboles de ce cas d'usage. Dans la sous-section (1.3.4) nous présentons alors comment nous exploitons cette représentation au sein de différentes stratégies de reconstruction d'objets. Finalement dans la sous-section (1.3.5) nous concluons sur ce cas d'usage.

³⁰Workshop on Graphics Recognition 2003.

³¹<http://www.cvc.uab.es/grec2003/>

³²Nous reportons le lecteur à [Valveny 04] sur ces aspects.

1.3.2 Pré-traitement contextuel

Nous employons une première étape de pré-traitement des images bruitées avant de procéder à la reconnaissance des symboles. Nous qualifions ce pré-traitement de contextuel car il est basé sur l'évaluation des types et niveaux de bruit des images. Ceci permet ainsi d'adapter le pré-traitement d'une image en fonction de son contexte. Plus particulièrement, nous avons basé ce pré-traitement sur l'évaluation de deux types de bruit [Ablameyko 00]³³ : impulsionnel et morphologique. La figure (1.24) suivante en donne des exemples. Le bruit impulsionnel³⁴ (a) correspond à l'ajout de petites composantes de la forme et du fond sur l'image. Le bruit morphologique (b) correspond à la déformation morphologique d'une image par érosion et/ou dilatation. Pour la mise en oeuvre de ce pré-traitement contextuel, nous avons développé deux opérateurs spécifiques d'évaluation du bruit. Nous présentons chacun d'entre eux dans la suite de cette sous-section.

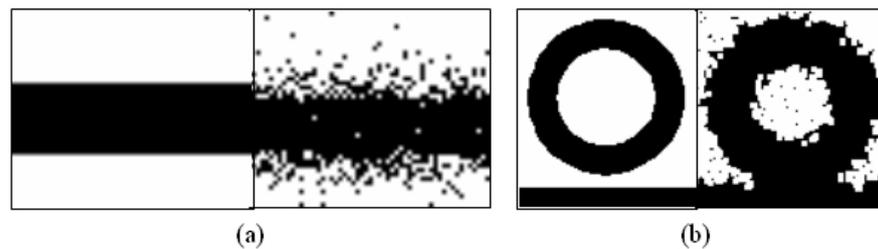


FIG. 1.24 – (a) bruit impulsionnel (b) bruit morphologique

Notre premier opérateur permet la détection du bruit impulsionnel. La méthode employée par cet opérateur est basée sur l'utilisation de notre méthode de marquage de composantes connexes. Suite au marquage, cette méthode teste l'existence de disproportions entre les surfaces des composantes (et/ou occlusions), afin de déterminer si du bruit impulsionnel est présent sur l'image. Elle crée et analyse pour cela un histogramme des surfaces des composantes. Le principe de cette analyse est alors similaire^{35,36} à celui de notre méthode de filtrage des frontières minoritaires. Dans l'histogramme des surfaces, les abscisses représentent les surfaces des composantes et les ordonnées les nombres de composante pour chacune de ces surfaces. Un tableau des rapports des surfaces est ensuite calculé à partir des différentes plages de cet histogramme. Le rapport maximum est alors recherché, il correspond à la disproportion de surface la plus importante entre deux composantes. Celui-ci est comparé à un seuil³⁷ utilisateur de façon à déterminer si du bruit impulsionnel est présent, ou pas, sur l'image.

³³Nous introduisons sommairement ces types ici et reportons le lecteur à [Ablameyko 00].

³⁴salt & pepper

³⁵Nous l'introduisons sommairement ici et reportons le lecteur page ??.

³⁶Le pseudo-algorithme en étant très proche nous ne le développerons pas ici.

³⁷Défini à $2^6 = 64$ pour ce cas d'usage.

Notre second opérateur permet la détection du bruit morphologique. Celui-ci exploite notre squelettiseur³⁸ basé sur la méthode de [Baja 94]. Nous utilisons la méthode de calcul de la carte des distances à partir d'une transformée ($d_{1,2}$). La carte des distances obtenue est ensuite moyennée afin d'obtenir une estimation d'épaisseur (\hat{e}_p) de l'image (équation (1.1)). Ce moyennage ne s'effectue que sur les (n) pixels formes de valeur (v_i) de la carte des distances. L'estimation (\hat{e}_p) obtenue est ensuite comparée à un ensemble de seuils, connus a priori, et définis en fonction des images traitées. Les écarts entre (\hat{e}_p) et ces seuils sont alors interprétés par l'opérateur d'évaluation afin de déterminer les niveaux d'érosion et/ou de dilatation des images.

$$\hat{e}_p = \frac{\sum_{i=1}^n v_i}{n} \quad (1.1)$$

Les évaluations faites, des bruits impulsif et morphologique, sont ensuite transcrites en faits par nos opérateurs. Ceux-ci sont fonction des types et niveaux de bruit détecté. Nous avons défini quatre catégories de faits concernant le bruit morphologique $\{\text{fortement érodé (se), érodé (e), valide (v), dilaté (d)}\}$ et deux pour le bruit impulsif $\{\text{impulsion (s), sans impulsion (us)}\}$. Notre système rsOPM exploite alors ces faits de façon à déterminer le pré-traitement à appliquer. Il utilise pour cela une base de prédicats définissant la mise en oeuvre d'opérateurs de pré-traitement. Ces derniers sont basés sur des traitements de morphologie mathématique³⁸ et notre méthode de filtrage de composantes connexes³⁹, ils sont au nombre de cinq : $\{\text{restauration (lr), filtrage morphologique (lf), restitution (r), filtrage de composantes (f), segmentation (s)}\}$. Le pré-traitement est stoppé lorsque l'image est évaluée valide et sans impulsion : $(v \wedge us)$. La figure (1.25) donne des exemples de résultats de pré-traitement contextuel. Le premier (a) est un cas de bruit impulsif ($v \wedge s$) qui déclenche une étape de filtrage de composantes $\{f\}$. Le second (b) est un cas de bruit d'érosion et impulsif ($e \wedge s$) qui déclenche un pré-traitement en deux étapes. Tout d'abord le système applique un filtrage morphologique $\{lf\}$ afin de restaurer en partie les cassures des composantes connexes, et de filtrer le bruit impulsif. La suppression du bruit impulsif permet ensuite d'appliquer un processus de restauration $\{r\}$ sur l'image encore érodée ($e \wedge us$). Ce traitement en deux étapes permet ainsi de restaurer les cassures des composantes connexes sans risquer d'amplifier le bruit impulsif.

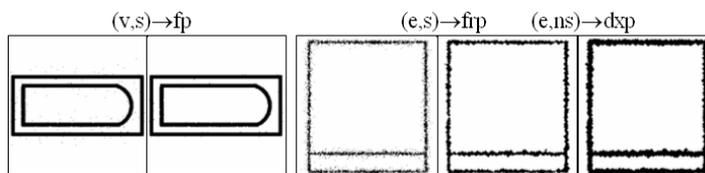


FIG. 1.25 – pré-traitement contextuel (a) une étape (b) deux étapes

³⁸ Exploité(s) de notre librairie PSI présentée en Annexe C.

³⁹ Présentée page 5.

1.3.3 Une représentation des symboles et son évaluation

À partir des images pré-traitées nous procédons à l'extraction de primitives graphiques pour la représentation des symboles. Nous nous sommes plus particulièrement basés sur une représentation par nos graphes hybrides. Les graphes hybrides que nous employons décrivent, des relations d'inclusion entre composantes et occlusions, et des relations de voisinage entre occlusions. Ce type de graphe hybride est en effet adapté pour la représentation des symboles de ce cas d'usage où les structures d'occlusions sont remarquables. La figure (1.26) donne un exemple de symbole et de ses occlusions (a), et son graphe hybride correspondant (b).

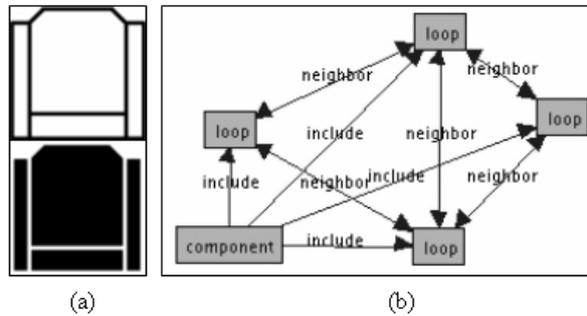


FIG. 1.26 – (a) symbole & occlusions (b) graphe hybride

Le but de notre démarche est plus précisément ici d'évaluer notre représentation par graphes hybrides pour ce cas d'usage. Afin d'estimer la qualité de cette représentation nous avons conçu une application d'évaluation. Celle-ci se déroule en trois étapes : construction des graphes hybrides (1), classification structurale (2) et évaluation des résultats (3). L'étape (1) est basé sur notre opérateur de construction de graphes hybrides. L'étape (2) exploite notre méthode d'appariement de graphes⁴⁰ ainsi que notre interface⁴¹ d'apprentissage. Enfin la dernière étape (3) ne fait⁴² que confronter les fichiers vérité terrain avec les résultats de reconnaissance [Valveny 04].

Pour cette évaluation nous avons utilisé les tests du jeu de symboles (20) (figure (1.22)) [Valveny 04]. Ces tests sont au nombre de 9 et se composent de 100 images chacun. Chacun d'entre eux correspond⁴² à un paramétrage différent des méthodes de bruitage. La figure (1.27) donne nos résultats sur ces tests : pour les six premiers nous obtenons des résultats parfaits excepté pour le n°5, soit 99.5% de reconnaissance au total. Nos résultats de reconnaissance s'écroulent par contre en ce qui concerne les tests 7 à 9 pour une moyenne de 82.3% de reconnaissance sur ces trois tests.

⁴⁰Exploitée de notre Librairie PSI présentée en Annexe C.

⁴¹XMLgml présentée en Annexe D de ce manuscrit.

⁴²Nous reportons le lecteur à [Valveny 04] sur ces aspects.

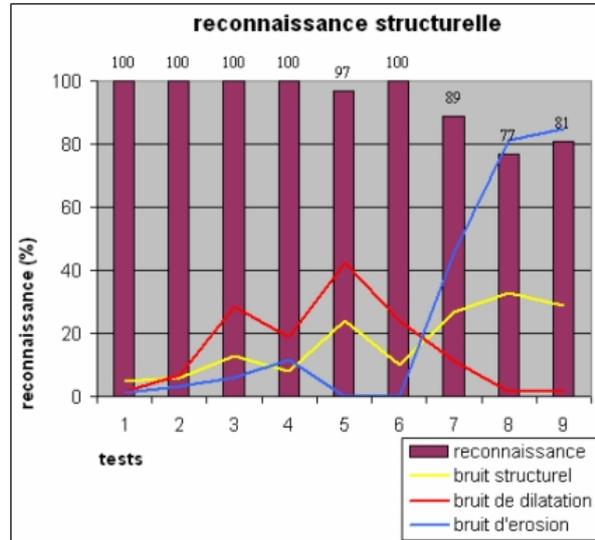


FIG. 1.27 – Résultats de la reconnaissance structurelle

Dans le but d'évaluer plus précisément nos résultats de reconnaissance nous avons mesuré différents types de bruit, à la fois sur les images traitées, mais aussi sur les graphes extraits. Il s'agit des bruits morphologiques (de dilatation et d'érosion) et du bruit structurel. La mesure des bruits morphologiques est détaillé dans l'équation (1.2). Celle-ci s'effectue entre une image test (it) et son image image modèle (im) non bruitée. Elle est basée sur la recherche du nombre de pixels communs (ctp) entre ces deux images, et le nombre de pixels (tp) de l'image test. (t) représente ici un type de pixel : respectivement forme et fond pour les bruits de dilatation et d'érosion. Le bruit structurel, quant à lui, donne le taux de graphes bruités parmi les graphes extraits. Ceux-ci sont détectés par appariement avec leurs graphes modèles : une distance non nulle indique la présence de bruit dans le graphe extrait.

$$\widehat{E}_m = 1 - \frac{ctp(it, im)}{tp(it)} \quad (1.2)$$

La figure (1.27) donne les résultats des mesures des bruits morphologiques et du bruit structurel. On peut constater une corrélation partielle entre ces bruits. En effet, les graphes hybrides sont basés sur l'extraction d'occlusions et de composantes connexes. Ces dernières sont des primitives graphiques robustes : les bons résultats le prouvent. Elles sont cependant très sensibles aux phénomènes de dilatation, mais surtout d'érosion qui déforment les composantes et occlusions. La corrélation entre les bruits morphologiques et celui structurel prouve que la dilatation et l'érosion sont les principales sources de bruitage des graphes hybrides extraits.

Basé sur ce constat nous avons cherché à déterminer la distribution du bruit structurel en fonction des classes de symbole. Le but de notre démarche est alors d'évaluer, et de catégoriser, le bruit structurel en fonction des bruits morphologiques. Nous avons calculé pour cela les résultats de reconnaissance pour chacune des classes de symbole. Nous nous sommes limités⁴³ aux classes de symbole suivantes parmi celles du jeu de symboles (20) : $\{A_a, A_e, A_f, A_g, A_h, A_i, E_b, E_c, E_e\}$. Nous présentons ces résultats de reconnaissance sur la figure (1.28).

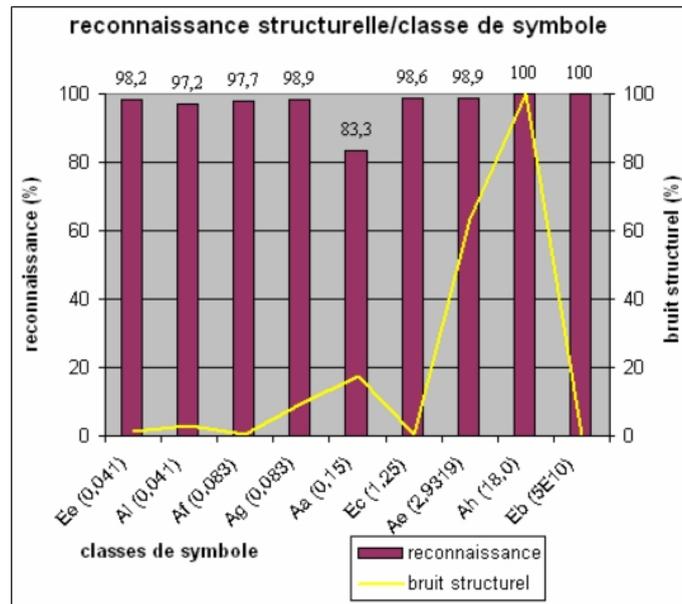


FIG. 1.28 – Résultats de la reconnaissance structurelle/classe de symbole

Ces résultats permettent de constater, dans un premier temps, que le bruit structurel n'a pas le même impact sur les résultats de reconnaissance selon la classe de symbole considérée. Dans le cas des classes de symbole $\{A_a, A_i\}$ le bruit structurel influe beaucoup sur les résultats de reconnaissance. Dans le cas des classes $\{A_e, A_h\}$ il n'a par contre aucun effet sur les résultats. En effet, les graphes modèles de ces différentes classes de symbole n'ont pas le même "critère de robustesse". Dans la figure (1.27) (b) nous avons calculé et trié⁴⁴, pour chaque graphe modèle, les taux de similarité minimum avec les autres graphes modèles de la base. Un tau de similarité minimum est donc la distance entre un graphe modèle et son graphe modèle le plus proche dans la base. Il peut être donc considéré comme un critère de robustesse du graphe modèle. Ainsi, nous pouvons voir deux groupes principaux de graphes modèles dans la base : ceux de faible robustesse (≤ 0.15) $\{A_a, E_e, A_i, A_f, A_g\}$ et ceux de forte robustesse (≥ 1.25) $\{A_e, A_h, E_b, E_c\}$.

⁴³Nous justifions cet aspect dans la sous-section suivante.

⁴⁴À l'aide de notre interface d'apprentissage XMLgml présentée en Annexe D.

Ces résultats nous ont permis dans un second temps de catégoriser le bruit structurel apparaissant sur les images de symbole. Nous avons catégorisé ce bruit en quatre types que nous présentons sur la figure (1.29). Nous les détaillons par la suite.

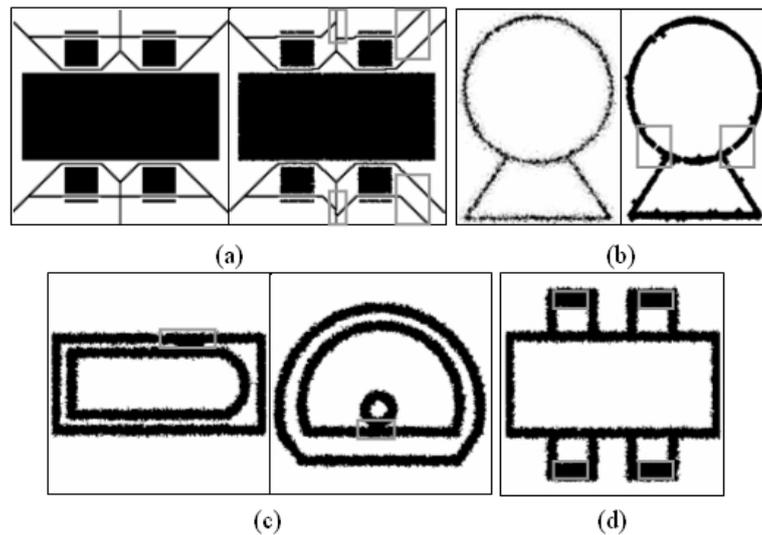


FIG. 1.29 – Bruit de (a) distorsion (b) rupture (c) sur-connexion (d) fermeture

Le premier est le bruit de distorsion de voisinage (a). Celui-ci est lié au bruitage des contours des occlusions. Le bruitage de ces contours perturbent alors leur extension durant la construction des relations de voisinage et créent de fausses frontières comme sur le symbole $\{A_h\}$. Cependant, ce bruit n'a que peu d'incidence sur les noeuds des graphes extraits (seulement sur les arcs). Ce bruit a alors un plus faible impact sur les résultats de reconnaissance. En effet, les composantes et/ou d'occlusions ont d'avantage de poids dans la représentation du symbole que leurs relations.

Le second est le bruit de rupture (b). Celui-ci est caractéristique des érosions trop importantes des images. Dans ce cas, le pré-traitement ne permet pas de restauration des traits des composantes ce qui introduit des cassures. Celles-ci rendent impossible a posteriori l'extraction des occlusions. L'épaisseur des traits étant homogène sur tout les symboles ce bruit affecte indifféremment l'ensemble des classes.

Enfin, les deux derniers sont liés aux bruits de dilatation des images. Le premier est le bruit de sur-connexion (c). Celui-ci apparaît dans les cas de composantes trop proches sur l'image : elles se retrouvent alors connectées à l'issue d'une dilatation trop importante. Il est typiquement rencontré sur les symboles $\{A_a, A_e\}$. Le deuxième est le bruit de fermeture (d). Celui-ci apparaît lors de la présence d'occlusions de faibles tailles au sein des composantes. Ces occlusions se retrouvent alors bouchées à l'issue d'une dilatation trop importante de l'image. Ce type de bruit apparaît fréquemment sur les symboles $\{A_e, A_h, A_i\}$.

1.3.4 Stratégies de reconstruction d'objets

Au cours des sous-sections précédentes nous avons présenté, tout d'abord, notre étape de pré-traitement contextuel des images bruitées. Nous avons présenté ensuite comment nous représentons les symboles (à partir des images pré-traitées) par des graphes hybrides. Nous avons alors évalué cette représentation et montré sa robustesse. Elle reste cependant sensible aux bruits morphologiques (de dilatation et d'érosion). Dans cette sous-section nous présentons comment nous exploitons cette représentation au sein de différentes stratégies de reconstruction d'objets.

Notre représentation par graphes hybrides n'est pas suffisante pour distinguer l'ensemble des symboles du jeu (20) [Valveny 04]. En effet, différents symboles présentent des graphes hybrides non univoques : c'est à dire avec plusieurs correspondances dans le jeu de symboles. La figure (1.30) en donne quelques exemples.

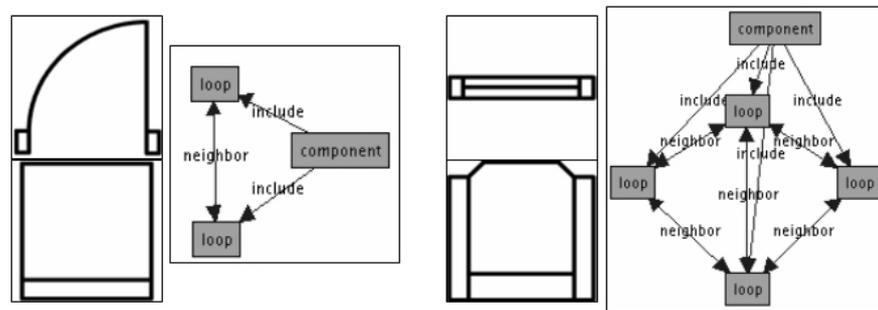


FIG. 1.30 – Exemples de graphes hybrides non univoques

De façon à pouvoir reconnaître l'intégralité des symboles nous avons spécialisé notre représentation par graphes hybrides en graphes de labels. Pour cela nous avons utilisé une approche statistique pour la reconnaissance des occlusions des symboles. La figure (1.31) (a) donne un exemple de symbole et de ses occlusions, son graphe hybride correspondant (b), et sa spécialisation en graphe de labels (c).

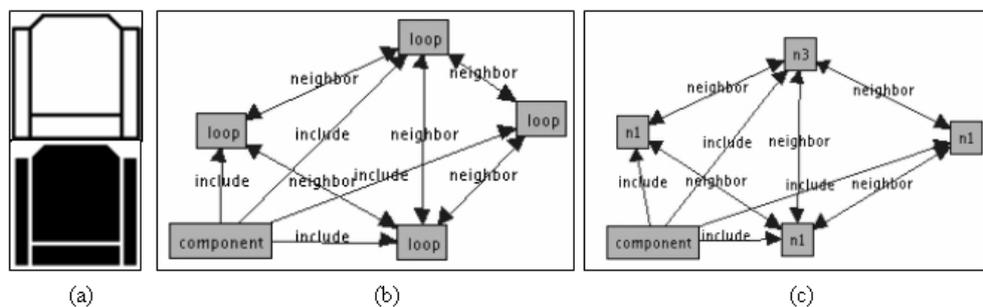


FIG. 1.31 – (a) symbole & occlusions (b) graphe hybride (c) graphe de labels

Pour ce cas d'usage nous avons volontairement choisi d'utiliser une approche statistique élémentaire pour la reconnaissance des occlusions des symboles. Le but de notre démarche est ici d'illustrer, comment notre approche par reconstruction d'objets, permet de résoudre un problème de reconnaissance des formes complexe malgré la simplicité de la chaîne utilisée. Pour ce faire, nous avons extrait des primitives géométriques $\{surface, périmètre, compacité\}$ pour décrire les occlusions des symboles. Cette extraction est réalisée à l'aide de nos méthodes combinées de marquage de composantes connexes et de détection de contours. Nous avons défini pour cela un opérateur spécifique combinant ces deux méthodes. Les primitives extraites sont ensuite traitées par un classifieur⁴⁵ de type k plus proches voisins. La base d'apprentissage employée par ce classifieur a été constituée⁴⁶ uniquement à partir des images idéales des symboles. Nous avons donc réalisé un apprentissage exact : c'est à dire sans tenir compte des types et niveaux de bruit rencontrés sur les images traitées. Ce choix a également pour but de montrer comment une approche par reconstruction d'objets peut parer aux éventuelles carences de la chaîne de reconnaissance des formes utilisée.

Notre approche repose donc sur l'utilisation de quatre opérateurs : extraction des graphes hybrides (1), extraction des caractéristiques géométriques (2), classification statistique (3) et classification structurale (4). Nous avons mis en oeuvre ces quatre opérateurs au sein de deux stratégies de reconstruction d'objets que nous présentons sur les OPD des figures (1.32) (a) et (b). Nous détaillons ces stratégies par la suite.

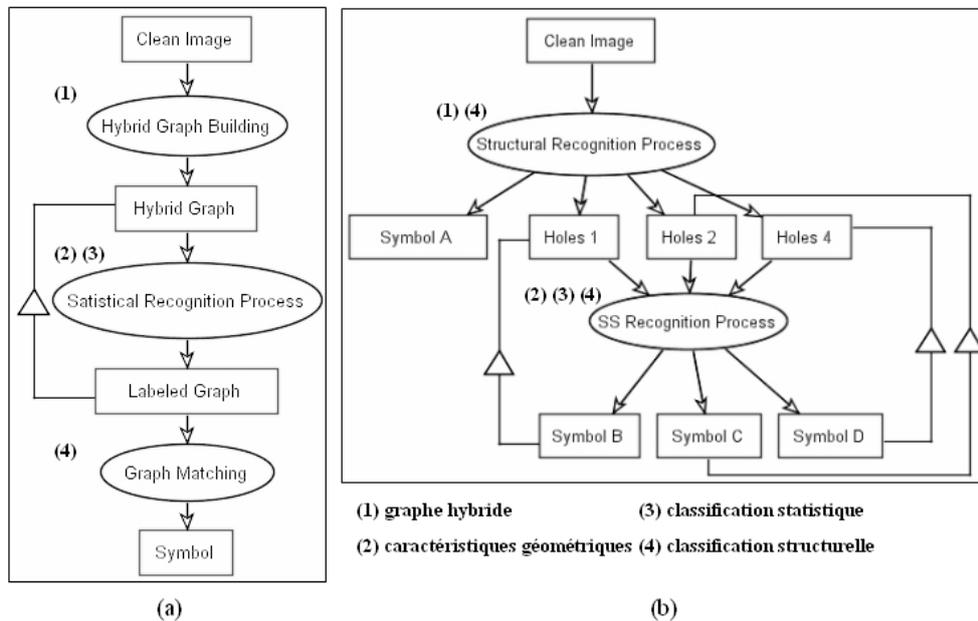


FIG. 1.32 – OPD par stratégie (a) séquentielle (b) taxinomique

⁴⁵ Exploité de notre librairie PSI présentée en Annexe C.

⁴⁶ À l'aide de notre interface XMLgml présentée en Annexe D.

La stratégie (a) ne fait qu'enchaîner séquentiellement les différents opérateurs $\{1, 2, 3, 4\}$. Ce type de stratégie est classiquement utilisé en reconnaissance des formes. La stratégie (b) met en oeuvre elle une taxinomie de représentations afin de les contrôler. Le principe en est présenté sur la figure (1.33). Elle procède tout d'abord à l'extraction des graphes hybrides et à leur classification à l'aide des opérateurs $\{1, 4\}$. Les graphes hybrides reconnus sont alors répartis en quatre objets : $\{Symbol A, Holes 1, Holes 2, Holes 4\}$. L'objet (*Symbol A*) regroupe les graphes hybrides univoques. Les autres objets $\{Holes 1, Holes 2, Holes 4\}$ regroupent les graphes hybrides non univoques composés respectivement de 1, 2, et 4 occlusions. Ces derniers nécessitent donc une spécialisation en graphes de labels pour assurer la reconnaissance de leurs symboles associés. Les opérateurs $\{2, 3, 4\}$ sont alors utilisés afin, de spécialiser les graphes hybrides par reconnaissance statistique des occlusions, puis de reconnaître les graphes de labels résultants. Des bases d'apprentissage spécifiques⁴⁷ sont utilisées pour chacun des objets $\{Holes 1, Holes 2, Holes 4\}$. Les résultats de la reconnaissance sont ensuite répartis en trois objets $\{Symbol B, Symbol C, Symbol D\}$ respectivement spécialisés de $\{Holes 1, Holes 2, Holes 4\}$. Tout au long du processus les résultats de reconnaissance du premier niveau de représentation (les graphes hybrides) sont retranscrits sous forme de faits et utilisés afin piloter la reconstruction d'objets.

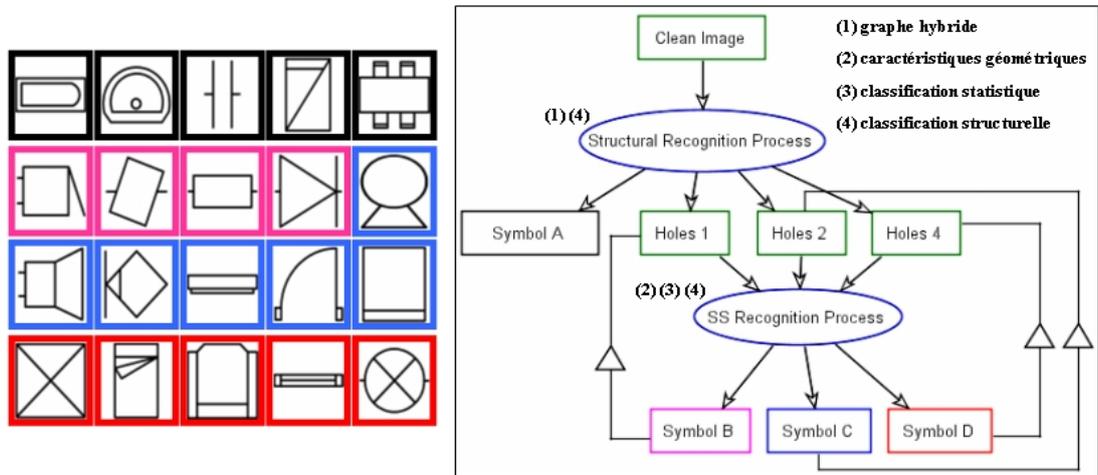


FIG. 1.33 – Mise en oeuvre de la taxinomie de représentations (1)

Afin d'illustrer plus en détails la taxinomie de représentations, nous donnons des exemples de reconnaissance de symboles sur la figure (1.34). Le symbole (a) est reconnu en une seule étape. Il est représenté en effet par un graphe hybride univoque, ce qui permet sa reconnaissance à l'aide d'un seul niveau de représentation. Les symboles (b) et (c) sont reconnus eux en deux étapes. Dans un première étape ils sont représentés par un même graphe hybride non univoque. Ce dernier est ensuite spécialisé en deux graphes de labels par reconnaissance des occlusions de chacun des symboles.

⁴⁷ Constituées à l'aide de notre interface XMLgml présentée en Annexe D.

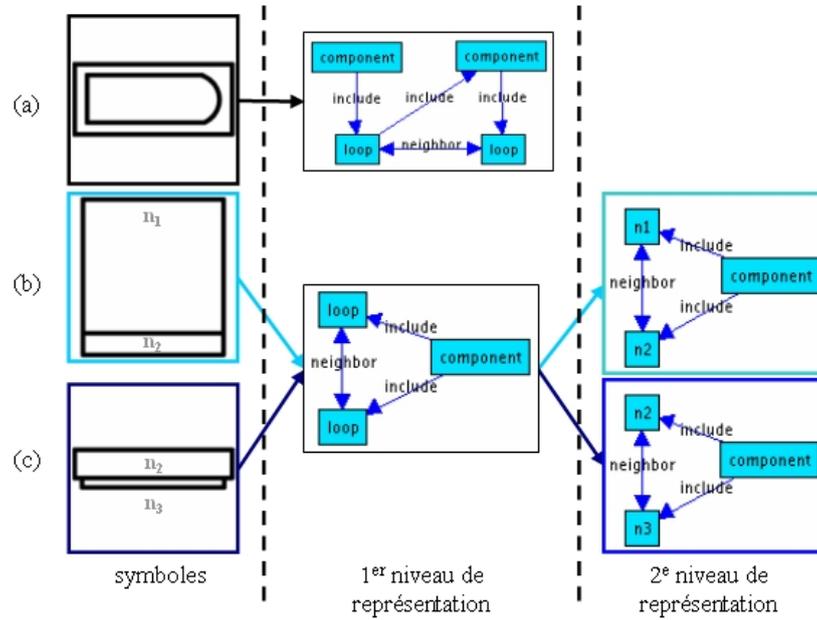


FIG. 1.34 – Mise en oeuvre de la taxinomie de représentations (2)

Dans la figure (1.35) nous comparons les résultats de reconnaissance de nos deux stratégies : séquentielle et taxinomique. Ces résultats ont été obtenus à partir des tests du jeu de symboles (20) (figure (1.22)) [Valveny 04].

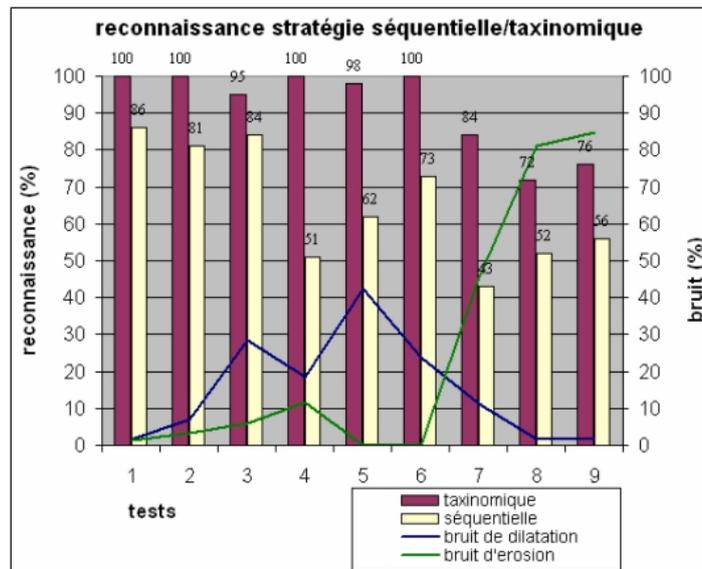


FIG. 1.35 – Résultats de reconnaissance par stratégie séquentielle vs taxinomique

Ces résultats illustrent l'intérêt de la stratégie taxinomique par rapport à la stratégie séquentielle. En effet les connaissances graphiques, apprises par le système, ont été réparties en différentes bases d'apprentissage statistique et structurel. Cette répartition fragmente ainsi les connaissances graphiques au sein du système simplifiant (fortement) le problème de reconnaissance. Nos résultats le prouvent : nous obtenons 98.8% de reconnaissance pour les tests 1 à 6 alors que les primitives statistiques utilisées sont élémentaires $\{surface, périmètre, compacité\}$ et que leur apprentissage est exact (réalisé à partir des images modèles seules). Évidemment, les résultats de cette stratégie sont dépendants de la première représentation utilisée par la taxinomie : celle par graphes hybrides. Celle-ci est sensible au bruit morphologique et plus particulièrement l'érosion des images. Cette sensibilité est la principale cause de la chute de nos résultats pour les tests 7 à 9.

La figure (1.36) compare les résultats de notre approche avec deux approches ayant été utilisées durant le concours GREC2003 [Valveny 04] : celles de [Yan 04] et [Yang 05]. Les résultats de notre approche sont proches des leur en ce qui concerne les tests 1 à 6. Ils sont cependant inférieurs sur les tests 7 à 9 à cause du fort bruit d'érosion présents sur les images de ces tests. Notre approche semble donc "pertinente" en ce qui concerne une partie des tests. Évidemment, nos résultats ont été obtenus hors conditions du concours ce qui relativise leur comparaison.

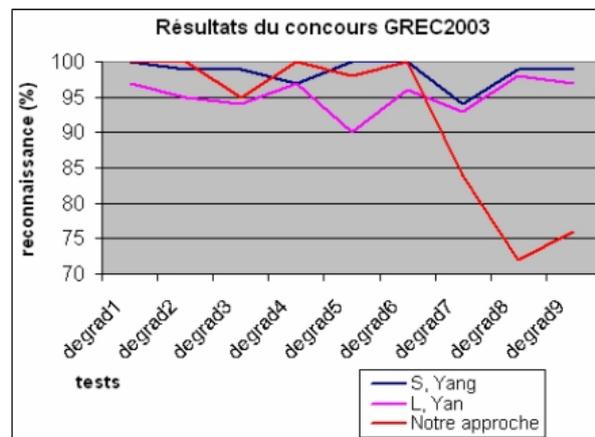


FIG. 1.36 – Comparaison avec les approches du concours

Ces résultats de reconnaissance ne constituent pas cependant l'unique critère de comparaison avec ces approches. Nous présentons ici quelques considérations sur la nature de ces approches :

- L'approche de [Yang 05] est de nature statistique, ce qui soulève la question de son utilisation pour des symboles non segmentés sur des images de plan technique.
- L'approche de [Yan 04] est basée sur une représentation vectorielle, ce qui soulève la question de son utilisation pour des symboles présentant des "formes pleines".

Nous avons également appliqué notre stratégie taxinomique aux tests du jeu de symboles (15) [Valveny 04]. Ce dernier est un sous-ensemble du jeu de symboles (20) présenté sur la figure (1.22). Il correspond à tous ses symboles à l'exception de A_a , A_d , A_e , E_a , et E_j . Nous avons alors redéfini un OPD taxinomique⁴⁸ pour cette nouvelle distribution de symboles. Nous avons également réalisé⁴⁹ un nouvel apprentissage pour la constitution des bases relatives à cet OPD. Nous présentons dans la figure (1.37) nos résultats de reconnaissance sur les tests de ce jeu de symboles (15). Ces tests mettent en oeuvre à la fois du bruit de type binaire mais aussi vectoriel. Ils sont au nombre de 27, et se répartissent selon 9 niveaux de bruit de type binaire et 3 vectoriel. Chaque test comprend 90 images pour un total de 2430 images pour l'ensemble des tests. Nos résultats sur ces tests sont (légèrement) inférieurs à ceux du jeu de symboles (20). Cette baisse s'explique naturellement par l'ajout du bruit de type vectoriel sur ces tests. Ce dernier déforme alors les symboles créant des erreurs de reconnaissance supplémentaires. La figure (1.38) donne quelques exemple d'erreurs de reconnaissance relatives au bruit vectoriel.

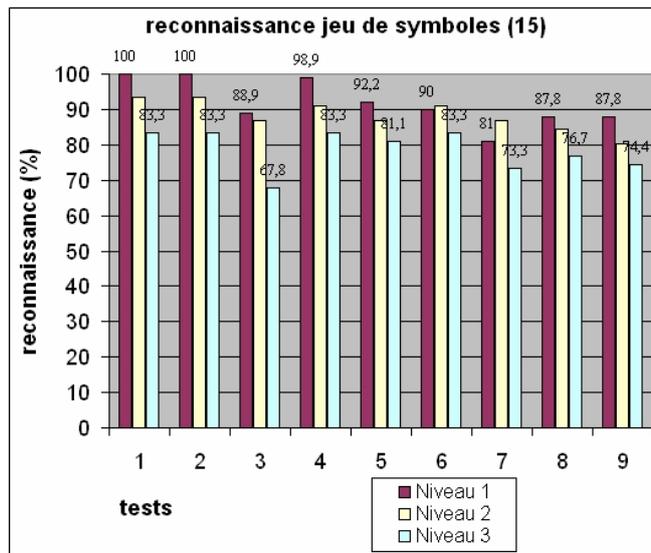


FIG. 1.37 – Résultats de reconnaissance des tests du jeu de symboles (15)

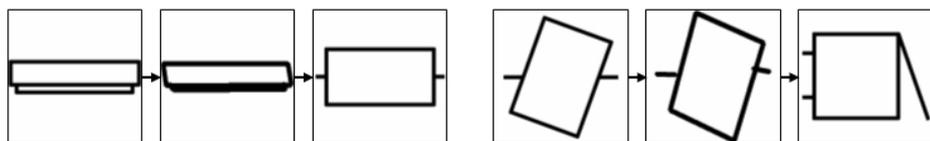


FIG. 1.38 – Exemples d'erreurs de reconnaissance dues au bruit vectoriel

⁴⁸Cet OPD est proche du précédent, nous ne le détaillerons pas ici.

⁴⁹À l'aide de notre interface d'apprentissage XMLgml présentée en Annexe D.

1.3.5 Conclusion

Dans cette section nous avons présenté un second cas d'usage de notre approche de reconstruction d'objets : le concours de reconnaissance de symboles de GREC2003. Celui-ci concerne la reconnaissance d'images de symboles architecturaux et électriques segmentés. Pour ce cas d'usage, nous avons mis en oeuvre et confronté deux stratégies de reconstruction d'objets : séquentielle et taxinomique. Celles-ci reposent sur l'utilisation de deux niveaux de représentation : les graphes hybrides et leur spécialisation en graphes de labels. Différents opérateurs sont alors utilisés afin d'extraire des primitives graphiques pour la représentation des symboles. La première stratégie enchaîne alors séquentiellement ces opérateurs, tandis que la seconde emploie une taxinomie de représentations afin de les contrôler. Au travers de cette taxinomie, les connaissances graphiques ont été fragmentées au sein du système simplifiant (fortement) le problème de reconnaissance. Nous illustrons cette considération au travers de nos résultats de reconnaissance et des bases d'apprentissage exploitées.

1.4 Conclusion

Dans ce chapitre nous avons présenté deux cas d'usage de notre approche de reconstruction d'objets. Ceux-ci concernent la reconnaissance de symboles sur des images de documents graphiques. Nous avons présenté un premier cas d'usage de reconnaissance sur des plans d'itinéraire FT. Nous avons montré alors comment une stratégie de reconstruction d'objets, combinant des approches région/contour, était particulièrement adaptée pour la reconnaissance des symboles sur ce type de plan. Le second cas d'usage concerne des images de symboles architecturaux et électriques segmentés du concours GREC2003. Nous avons illustré de quelle façon une stratégie de reconstruction d'objets, basée sur une taxinomie de représentations, permettait de fragmenter les connaissances graphiques au sein du système simplifiant (fortement) le problème de reconnaissance.

Ainsi, au travers de ces deux cas d'usage, nous avons illustré l'intérêt de notre approche par reconstruction d'objets pour des problèmes de reconnaissance. Via cette approche, les systèmes tirent partie des variations de représentations des connaissances, ainsi que la complémentarité des opérateurs, afin de simplifier le processus de reconnaissance et d'accroître leurs performances de traitement. De même, nous avons illustré les propriétés de généricité et d'adaptabilité de notre approche. Nous avons en effet confronté, et fait évoluer, différentes stratégies de reconstruction à partir de la combinaison d'un même jeu d'opérateurs⁵⁰.

⁵⁰Certains opérateurs spécifiques ont été toutefois développés pour certaines stratégies.

Bibliographie

- [Ablameyko 00] S. Ablameyko & T.P. Pridmore. Machine interpretation of line drawing images. Springer Verlag Publisher, ISBN : 3-540-76207-8, 2000.
- [Adam 98] S. Adam. Interprétation de documents du réseau téléphonique, approche multi-spécialistes. Rapport de Master, Laboratoire PSI, Université de Rouen, France, 1998.
- [Adam 01] S. Adam. *Interprétation de Documents Techniques, des Outils à leur Intégration dans un Système à Base de Connaissances*. Thèse de Doctorat, Université de Rouen, France, 2001.
- [Baja 94] G.S. Di Baja. *Well-Shaped, Stable, and Reversible Skeletons from the (3,4)-Distance Transform*. Journal of Visual Communication and Image Representation, vol. 5, no. 1, pages 107–115, 1994.
- [Grenier 01] V. Grenier. *Contribution à l'Interprétation de Documents Techniques : une Approche Système*. Thèse de Doctorat, Université de Rouen, France, 2001.
- [Hameau 02] A. Hameau & G. Renault. Réalisation d'un outil de recherche d'arcs de cercle. Rapport de Master, Laboratoire PSI, Université de Rouen, France, 2002.
- [Lassaulzais 97] A. Lassaulzais. Contribution au développement d'un système de vectorisation et d'interprétation de plans de réseaux. Rapport de Master, Laboratoire PSI, Université de Rouen, France, 1997.
- [Nicolas 02] S. Nicolas. Vers une bibliothèque pour la manipulation et la factorisation de sous-graphes. Rapport de Master, Laboratoire PSI, Université de Rouen, France, 2002.
- [Nilsson 80] J. Nilsson. Principles of artificial intelligence. Morgan Kaufmann Editor, ISBN : 0-935382-01-1, 1980.
- [Otsu 79] N. Otsu. *A Threshold Selection Method from Gray-Level Histograms*. Transactions on Systems, Man and Cybernetics (TSMC), vol. 9, no. 1, pages 62–66, 1979.
- [Pavlidis 74] T. Pavlidis & S. L. Horowitz. *Segmentation of Plane Curves*. Transactions on Computers (TC), vol. 23, pages 860–870, 1974.
- [Perin 04] S. Perin. Formalisation et opérationnalisation de connaissances graphiques pour l'interopérabilité en analyse d'image de document. Rapport de Master, Laboratoire PSI, Université de Rouen, France, 2004.
- [Ramer 72] V. Ramer. *An Iterative Procedure for the Polygonal Approximation of Plane Curves*. Computer Vision Graphics and Image Processing (CVGIP), vol. 1, pages 244–246, 1972.

- [Samet 94] H. Samet & A. Soffer. *A Legend-Driven Geographic Symbol Recognition System*. In International Conference on Pattern Recognition (ICPR), pages 350–355, 1994.
- [Valveny 04] E. Valveny & P. Dosch. *Symbol Recognition Contest : A Synthesis*. In Workshop on Graphics Recognition (GREC), volume 3088 of *Lecture Notes in Computer Science (LNCS)*, pages 368–386, 2004.
- [Yan 04] L. Yan & L. Wenyin. *Interactive Recognizing Graphic Objects in Engineering Drawings*. In Workshop on Graphics Recognition (GREC), volume 3088 of *Lecture Notes in Computer Science (LNCS)*, pages 126–137, 2004.
- [Yang 05] S. Yang. *Symbol Recognition via Statistical Integration of Pixel-Level Constraint Histograms : A New Descriptor*. *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 27, no. 2, pages 278–281, 2005.

Table des figures

1.1	Extrait de plan d'itinéraire FT sur retombe	1
1.2	Prétraitement des plans FT	2
1.3	Symboles FT	2
1.4	Exemple de graphe de voisinage	3
1.5	Cas de fragmentation d'un symbole	4
1.6	Cas d'occlusions parasites	4
1.7	Topologie des occlusions des symboles	5
1.8	Extraction et filtrage d'occlusions	5
1.9	Cas de proximité des symboles	6
1.10	Cas de regroupement des symboles	7
1.11	Reconnaissance statistico-structurale des symboles	7
1.12	Classes d'occlusions des symboles	8
1.13	Base de graphes modèles	10
1.14	OPD des stratégies de groupement & segmentation	11
1.15	Exemples de résultats des stratégies de groupement & segmentation	12
1.16	Exemple de relations de voisinage distantes	12
1.17	OPD par stratégie de simplification	14
1.18	Exemple de vectorisation des contours et de simplification	15
1.19	Illustration de la méthode d'effacement circulaire	15
1.20	Exemple de soustraction d'images d'occlusions	16
1.21	Exemple de caractère connecté à un symbole	17
1.22	Jeu de symboles (20)	18
1.23	Types et niveaux de bruit	18
1.24	Bruits impulsionnel & morphologique	19
1.25	Exemple de pré-traitement contextuel	20
1.26	Exemple de graphe hybride	21
1.27	Résultats de la reconnaissance structurale	22
1.28	Résultats de la reconnaissance structurale/classe de symbole	23
1.29	Types de bruit structurel	24
1.30	Exemples de graphes hybrides non univoques	25
1.31	Spécialisation d'un graphe hybride en graphe de labels	25
1.32	OPD des stratégies séquentielle et taxinomique	26
1.33	Mise en oeuvre de la taxinomie de représentations (1)	27
1.34	Mise en oeuvre de la taxinomie de représentations (2)	28
1.35	Résultats de reconnaissance par stratégie séquentielle vs taxinomique	28
1.36	Comparaison avec les approches du concours	29
1.37	Résultats de reconnaissance des tests du jeu de symboles (15)	30
1.38	Exemples d'erreurs de reconnaissance dues au bruit vectoriel	30

Liste des tableaux

1.1	Base d'images tests	3
1.2	Distribution des cas de proximité selon les classes de symbole	6
1.3	Résultats de la reconnaissance statistique des occlusions	8
1.4	Carte des distances des classes d'occlusion	8
1.5	Résultats de la reconnaissance de symboles des différentes stratégies	11

Liste des pseudo-algorithmes

1.2.1 Effacement circulaire	16
-----------------------------------	----

Table des matières

1	Cas d'usage	0
1.1	Introduction	0
1.2	Reconnaissance de symboles sur plans d'itinéraire	1
1.2.1	Introduction	1
1.2.2	Base d'images tests	3
1.2.3	Extraction et filtrage des occlusions	4
1.2.4	Construction et exploitation des graphes de voisinage	6
1.2.5	Reconnaissance structurelle de symboles	9
1.2.6	Stratégies de reconstruction d'objets	10
1.2.7	Conclusion	17
1.3	Concours de reconnaissance de symboles GREC2003	18
1.3.1	Introduction	18
1.3.2	Pré-traitement contextuel	19
1.3.3	Une représentation des symboles et son évaluation	21
1.3.4	Stratégies de reconstruction d'objets	25
1.3.5	Conclusion	31
1.4	Conclusion	31
	Bibliographie	32
	Table des figures	34
	Liste des tableaux	35
	Liste des pseudo-algorithmes	36