De l'appariement de graphes symboliques à l'appariement de graphes numériques : Application à la reconnaissance de symboles

Rashid Jalal Qureshi, Jean-Yves Ramel, Hubert Cardot

Université François-Rabelais de Tours,
Laboratoire d'Informatique (EA 2101), 64, Avenue Jean Portalis, 37200 Tours – France

{rashid.qureshi,jean-yves.ramel,hubert.cardot}@univ-tours.fr

Résumé *: Les représentations sous forme de graphes structurels ont été appliquées dans un grand nombre de problèmes en vision par ordinateur et en reconnaissance de formes. Néanmoins, lors de l'étape d'appariement de graphes, les algorithmes classiques d'isomorphisme de graphes sont peu performants quand l'image est dégradée par du bruit ou des distorsions vectorielles. Cet article traite de la reconnaissance de symboles graphiques grâce à la formulation d'une nouvelle mesure de similarité entre leur représentation sous forme de graphes étiquetés. Dans l'approche proposée, les symboles sont d'abord décomposés en primitives structurelles et un graphe attribué est alors généré pour décrire chaque symbole. Les nœuds du graphe représentent les primitives structurelles tandis que les arcs décrivent les relations topologiques entre les primitives. L'utilisation d'attributs numériques pour caractériser les primitives et leurs relations permet d'allier précision et, invariance à la rotation et au changement d'échelle. Nous proposons également une nouvelle technique d'appariement de graphes basée sur notre fonction de similarité qui utilise les valeurs numériques des attributs pour produire un score de similarité. Cette mesure de similarité a de nombreuses propriétés intéressantes comme un fort pouvoir de discrimination, une invariance aux transformations affines et une faible sensibilité au bruit.*

Mots-clés *:* Reconnaissance de symboles graphiques, representation structurelle, comparaison de graphes.

1 Introduction

De nombreuses méthodes de reconnaissance de symboles peuvent être trouvées dans la littérature [LUO 04] [TER 04] [MES 96] [DOS 04] [LLA 01] et un état de l'art est proposé en [COR 00]. La plupart des techniques présentées ont été mises au point et utilisées pour un domaine d'application particulier mais très peu de travaux ont été effectués dans le but de produire un système générique. Concernant les méthodes utilisant des graphes structurels, il ressort de notre étude que, en pratique, seuls des algorithmes de mise en correspondance inexacte peuvent être efficaces en raison du bruit et de la variabilité des formes présentes dans les document graphiques.

Ainsi, cet article présente une nouvelle méthode de reconnaissance de symboles graphiques visant à plus de généricité. Cette méthode utilise un algorithme d'appariement imparfait de graphes comportant des étiquettes symboliques et numériques.

L'article est organisé de la manière suivante : la section 2 donne une vue générale des étapes de mise en place de la représentation structurelle des symboles. La section 3 présente notre algorithme d'appariement de graphes par mesure de similarité. Ensuite, dans la section 4, nous décrivons les résultats obtenus avant de conclure sur l'intérêt de notre méthode.

2 Représentation structurelle et numérique de symboles

Dans la plupart des méthodes structurelles, nous observons l'usage d'une représentation symbolique des formes à reconnaître. Par exemple, une méthode basée sur la construction d'une signature vectorielle pour discriminer les symboles dans les documents techniques est présentée en [DOS 04]. Les signatures vectorielles sont construites à partir d'une description symbolique des relations existantes entre les segments comme le parallélisme, les angles droits, les rapports de recouvrement, le nombre de jonctions en V ou en L, etc. Ce type de description peut fonctionner pour reconnaitre quelques symboles dans un domaine particulier mais quand le nombre de symboles à reconnaitre augmente, de nombreuses fausses alertes sont alors observées. C'est pourquoi nous avons décidé de construire une représentation qui fusionne des informations numériques et structurelles provenant des traitements de bas niveau. Ceci nous a conduit, par la suite, à trouver une méthode d'appariement de graphes qui soit capable de traiter ces deux types d'information simultanément. Pour cela, nous utilisons des graphes attribués. Le graphe permet de gérer les relations structurelles entre les primitives et les attributs fournissent des informations numériques précises associées à ces primitives et à leurs relations.

2.1 Vectorisation des symboles

Deux approches principales peuvent être utilisées pour la vectorisation de symboles : celles qui sont basées sur le calcul d'un squelette ou le calcul de l'axe médian

des formes, et celles qui font correspondre les contours opposés des lignes.

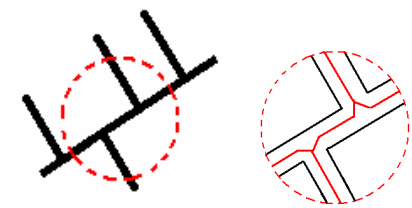


FIG. 1 - Vectorisation par approche squelette

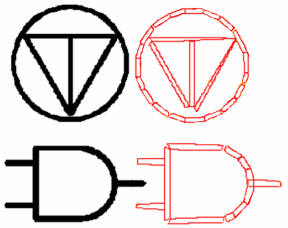


FIG. 2 - Appariement de contours

Une comparaison de ces deux approches a été présentée récemment par Tombre [TOM 00]. Le principal défaut des méthodes de vectorisation basées sur les squelettes est qu'elles sont sensibles au bruit et tendent à introduire des distorsions aux points d'extrémités et aux jonctions (figure 1). Les approches d'appariement de contours comme [HAN 94] et [RAM 98] sont souvent plus robustes au bruit que les méthodes basées sur le squelette. La technique proposée dans [RAM 00] pour représenter les images de documents graphiques à l'aide d'un ensemble de quadrilatères fonctionne bien dans le cas des symboles graphiques linéaires. Nous avons donc choisi d'utiliser cette méthode pour vectoriser nos symboles graphiques et obtenir nos primitives structurelles de base : les quadrilatères (figure 2).

2.2 Construction du graphe structurel

Chaque quadrilatère obtenu par vectorisation représente un trait dans l'image et possède des attributs comme la longueur (ℓ) de son axe médian, les angles des deux vecteurs le constituant (v_1 , v_2), la largeur de chaque côté (w_1 , w_2) et une zone d'influence représentée par le rectangle en pointillés dans la figure 3. Les dimensions de cette zone dépendent de la longueur (ℓ) et des largeurs des deux côtés de chaque quadrilatère.

La zone d'influence joue un rôle crucial dans la phase de construction du graphe puisque seuls les quadrilatères se trouvant dans la zone d'influence de chaque quadrilatère seront considérés lors de la mise en place des arcs décrivant les relations de voisinage entre primitives.

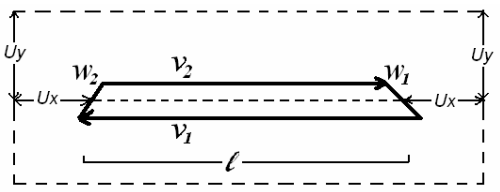


FIG. 3 - Un quadrilatère et sa zone d'influence

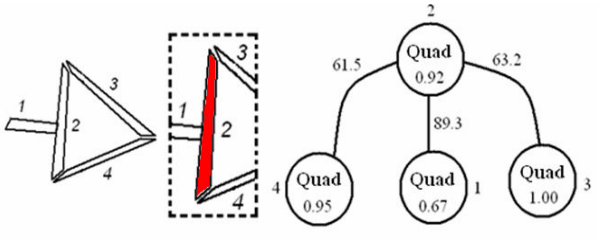


FIG. 4 – Partie du graphe associé au quadrilatère 2

La figure 4 montre comment les quadrilatères proches spatialement sont associés lors de la construction du graphe. Chaque sommet du graphe représente un quadrilatère et ses caractéristiques sont enregistrées comme attributs (longueur, angle, épaisseur…). De même, les attributs de chaque arc fournissent des informations précises sur la relation existante entre les 2 primitives considérées. L'analyse successive de la zone d'influence de chaque quadrilatère extrait de l'image initiale permet d'obtenir, au final, un graphe complet représentant l'ensemble du contenu de l'image.

2.3 Caractéristiques actuelles des nœuds et des arcs

Lors de la phase de reconnaissance, nous utilisons actuellement qu'un seul attribut parmi tous ceux disponibles pour caractériser les quadrilatères. Il s'agit de leur longueur relative. La longueur relative λ_i du $i^{ème}$ quadrilatère d'un symbole peut être calculée comme le rapport entre sa longueur ℓ_i et la longueur ℓ_{max} du plus long quadrilatère extrait de l'image, c'est-à-dire :

$$\lambda_i = \ell_i / \ell_{max}$$

L'usage de la longueur relative à la place de la longueur exprimée en pixels permet l'invariance par rapport au facteur d'échelle.

Actuellement, chaque arc ne comporte également qu'un seul attribut correspondant à l'angle relatif (φ_{ij} [0, π]) entre les deux quadrilatères (Q_i, Q_j) mis en relation. Cette valeur est calculée à partir des angles de leur axe médian, c'est à dire si θ_i et θ_j sont les angles de Q_i et Q_j par rapport à l'axe horizontal (figure 5), l'angle relatif entre Q_i et Q_j est exprimé par

$$\varphi_{ij} = \left| \theta_i - \theta_j \right|$$

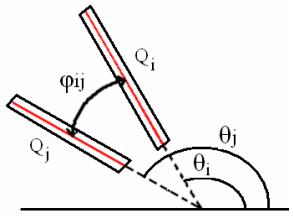


FIG. 5 - Angle entre quadrilatères

L'usage de l'angle relatif permet au graphe d'être invariant par rotation. Cette méthode permet donc, au final, d'obtenir une représentation du contenu d'une

image à l'aide d'un graphe non orienté muni d'attributs numériques et symboliques sur les sommets et les arcs (Fig. 4).

Soit A_v et A_E respectivement les ensembles des sommets et des arcs. On peut alors définir un graphe attribué G par un 4-tuple $G = (V , E , \alpha , \beta)$ où : V est l'ensemble fini des sommets, $E \subseteq V \times V$ est l'ensemble des arcs, $\alpha : V \rightarrow A_V^i$ est une fonction qui assigne les attributs aux sommets, $\beta : E \rightarrow A_E^j$ est une fonction qui assigne les attributs aux arcs. Ici, i varie de *1* à δ et j varie de *1* à Ω, où δ et Ω représentent respectivement le nombre d'attributs associés aux sommets et aux arcs.

3 De l'appariement de graphes symboliques à l'appariement de graphes numériques

3.1 Les algorithmes classiques d'appariement

Le problème de l'appariement de graphes est de trouver une correspondance entre les sommets d'un graphe et les sommets d'un autre graphe qui satisfasse à certaines contraintes ou critères d'optimalité [BUN 99].

Les méthodes utilisées classiquement sont l'isomorphisme de graphe/sous-graphe ou la recherche d'appariement par utilisation d'une distance d'édition. Un algorithme d'isomorphisme de graphe/sous-graphe tolérant aux erreurs formulé en terme de graphe d'adjacence de régions (RAG) a été proposé en [LLA 01] pour reconnaître des symboles dans des diagrammes dessinés à la main. Une distance d'édition classique est utilisée pour calculer la similarité entre les chaînes décrivant les frontières des régions contenues dans les graphes. Cependant, en cas de présence de bruit, des points caractéristiques supplémentaires peuvent apparaitre, et complexifier fortement l'analyse des graphes. Le concept de graphe « probing » (sondage de graphes) présenté en [LOP 01] est une combinaison intéressante des approches structurelles et statistiques. Différents types d'information, appelés sondes, sont extraits de la représentation structurelle d'une forme décrite par un graphe pour constituer un vecteur de caractéristiques statistiques. Il devient alors possible d'utiliser une méthode de reconnaissance statistique classique pour reconnaitre les formes. Les sondes utilisées peuvent concerner, par exemple, les occurrences d'un certain type de sommets, leur étiquette, le nombre d'arcs sur les sommets, etc. Bien que le sondage de graphe permette de calculer une mesure de similarité entre deux graphes ou entre un graphe et un sous-graphe, il ne donne pas d'information sur la meilleure mise en correspondance entre les deux graphes.

3.2 Appariement de graphes numériques

Nous proposons une nouvelle méthode pour déterminer le meilleur appariement possible entre deux représentations structurelles de symboles graphiques grâce à la formulation d'une nouvelle mesure de similarité entre graphes. Notre méthode peut être vue comme un double processus : la sélection d'appariements entre sommets, et ensuite le calcul d'une mesure de similarité pour ces appariements spécifiques entre les deux graphes.

3.2.1 Mesure de similarité entre graphes

Pour comparer deux graphes avec des attributs numériques et symboliques sur les sommets et sur les arcs, nous proposons une mesure basée sur un calcul de distance entre les attributs associés aux sommets et un calcul de distance entre les attributs associés aux arcs. Ainsi, nous calculons la similarité entre deux graphes attribués pour une mise en correspondance Mp donnée à l'aide de la formule suivante :

$$Sc_{Mp} = \left[\sum_{i=1}^{m} (1 - \Delta V_i) + \sum_{j=1}^{n} (1 - \Delta E_j) - \left(\sum_{i=1}^{k} \omega_i + \sum_{j=1}^{\ell} \omega'_j \right) \right] \quad (1)$$

où m est le nombre total de sommets appariés dans Mp et n le nombre total d'arcs entre ces nœuds. ω_i et ω'_j sont les pénalités que nous avons choisies d'appliquer lorsque un nœud i ou un arc j est associé à plusieurs autres dans l'appariement courant.

Grâce à ce mécanisme de pénalité, nous autorisons l'association d'un sommet (ou d'un arc) dans un graphe à un ou plusieurs sommets (ou arcs) dans l'autre graphe. Ces pénalités dépendent du nombre d'attributs utilisés lors des calculs de distance entre sommets (et entre arcs).

$$\Delta V_i = \frac{\sum_{k=1}^{\delta} f_k \left(A_V^k , A_{V'}^k \right)}{\delta} \quad (2) \text{ et } \Delta E_j = \frac{\sum_{k=1}^{\Omega} g_k \left(A_E^k , A_{E'}^k \right)}{\Omega} \quad (3)$$

Dans la formule (1), ΔV_i correspond à la distance entre deux sommets appariés, normalisée entre 0 et 1 (formule(2)). La fonction f_k compare la valeur du $k^{ième}$ attribut des deux sommets appariés et retourne une valeur de similarité entre 0 et 1. Comme nous l'avons déjà vu, pour l'instant, un seul attribut est considéré lors de la comparaisons de 2 sommets (à savoir la longueur relative du quadrilatère), on utilise donc simplement la formule (4). De même, ΔE_j correspond à la distance entre deux arcs normalisée entre 0 et 1(formule(3)).

La fonction g_k est utilisée pour comparer les deux valeurs du $k^{ième}$ attribut des deux arcs. Actuellement, nous avons utilisé un seul attribut lors de la comparaison des arcs, à savoir l'angle relatif entre les deux quadrilatères associés. Nous avons donc défini g_1 comme indiqué formule (5).

$$f_1 \left(A_V^1 , A_{V'}^1 \right) = \left| \lambda_i - \lambda_{i'} \right| \quad (4)$$

$$g_1 \left(A_E^1 , A_{E'}^1 \right) = \frac{\left| \varphi_{ij} - \varphi_{i'j'} \right|}{180} \quad (5)$$

Il est simple d'introduire d'autres attributs dans le calcul de similarité entre 2 sommets ou 2 arcs. Il suffit pour cela de définir les fonction f_k ou g_k à utiliser lors de la comparaison. Enfin, pour normaliser la mesure de similarité des graphes entre 0 et 1, on utilise :

$$Sim(G,G') = \frac{Sc_{Mp}}{\left[\delta \times \left(C(V) + C(V')\right) + \Omega \times \left(C(E) + C(E')\right)\right]}$$

où C est une fonction de cardinalité qui retourne le nombre de sommets ou d'arcs dans un graphe.

3.2.2 Recherche du meilleur appariement

Pour déterminer le meilleur appariement, la manière la plus simple de procéder est de tester toutes les mises en correspondance possibles pour les deux graphes et de garder celle qui produit le meilleur score de similarité. Comme l'espace de recherche s'accroit de manière exponentielle quand la taille des graphes augmente, la demande en ressources de calcul devient rapidement excessive. C'est pourquoi des algorithmes d'une complexité moindre comme les procédures de séparation et évaluation ou les algorithmes gloutons sont souvent utilisés pour éviter les recherches exhaustives, mais avec le risque de trouver une solution sous-optimale. Dans notre système, pour trouver le meilleur appariement entre les deux graphes, nous utilisons un algorithme qui s'inspire de l'algorithme glouton présenté en [CHA 03]. SimGraph est le nom de la routine que nous avons mise en place pour trouver le meilleur appariement entre 2 graphes. Elle prend deux graphes en entrée et retourne la meilleure mise en correspondance (M_{best}) des sommets entre les 2 graphes.

La recherche débute avec une carte des mises en correspondance vide et, à chaque itération, un couple de sommets candidat (v, v') est autorisé à entrer dans la carte des mises en correspondance courante (M_c) à condition qu'il y ait augmentation maximale du score de similarité. S'il y a plus d'un couple candidat qui fait augmenter la fonction de score de manière égale, un score de second niveau est calculé sur la base de l'analyse des arcs associés aux sommets candidats. Cette condition aide au choix des sommets à faire rentrer dans l'appariement à l'aide d'une fonction « look_ahead » qui teste l'efficacité d'un couple candidat avant de l'autoriser à entrer dans M_c . Ce processus d'insertion itératif de couples de nœuds associés continue jusqu'à ce que le score ait cessé d'augmenter et qu'il n'y ait plus, ni de nœuds ni d'arcs à ajouter dans la carte des mises en correspondance courante.

3 Résultats expérimentaux

Pour tester notre système, nous avons effectué des essais en utilisant la base GREC2003 [GRE 03]. En travaillant avec 50 symboles différents, nous avons généré un ensemble de 1100 exemples de symboles avec différents niveaux de distorsion, des transformations et des ajouts de bruits classiques. En raison de l'usage d'attributs invariants à la rotation et au changement d'échelle, un taux de reconnaissance de 100% a été obtenu dans le cas des images de test ayant subi uniquement des rotations et des changements d'échelle. La table 1 montre les scores de similarité obtenus en utilisant notre algorithme d'appariement de graphes entre des modèles de symboles et des symboles identiques globalement transformés respectivement par rotation et changement d'échelle. Pour des raisons de concision, seuls les scores de quelques prototypes sont montrés dans les tableaux.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	…
RS_1	1.00	0.87	0.86	0.80	0.79	0.82	0.76	0.82	…
RS_2		1.00	0.88	0.83	0.80	0.87	0.83	0.78	…
RS_3			1.00	0.85	0.77	0.85	0.82	0.83	…
RS_4				1.00	0.86	0.82	0.78	0.76	…
RS_5					1.00	0.82	0.81	0.89	…
RS_6						1.00	0.74	0.75	…
RS_7							1.00	0.81	…
RS_8								1.00	…

TAB.1 - Scores de similarité obtenus entre symboles modèles Mi et symboles après rotation et changements d'échelle (RSi)

Les tableaux 2 et 3 montrent les scores de similarité obtenus à l'aide de notre algorithme d'appariement de graphes entre des modèles de symboles et les symboles déformés localement, respectivement par du bruit et par une distorsion vectorielle (figure 6). Nous pouvons voir que le score de similarité n'est pas toujours égal à 1, mais que la plupart du temps le score de similarité maximal est présent sur la diagonale du tableau ce qui montre l'aptitude au rappel de notre système. Les taux de reconnaissance qui figurent dans le tableau 4 incluent tous les types de dégradation appliquées sur les 50 modèles de symboles présentés figure 7.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	…
N_1	0.92	0.51	0.55	0.80	0.54	0.49	0.51	0.64	…
N_2		0.91	0.53	0.80	0.61	0.61	0.54	0.55	…
N_3			0.92	0.62	0.72	0.37	0.63	0.61	…
N_4				0.90	0.73	0.56	0.56	0.58	…
N_5					0.90	0.44	0.67	0.63	…
N_6						0.93	0.59	0.61	…
N_7							0.91	0.59	…
N_8								0.91	…

TAB.2 - Scores de similarité entre symboles modèles et symboles bruités

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	…
D_1	0.90	0.68	0.46	0.70	0.64	0.67	0.46	0.65	…
D_2		0.93	0.83	0.78	0.68	0.70	0.68	0.45	…
D_3			0.91	0.72	0.66	0.64	0.64	0.56	…
D_4				0.91	0.63	0.76	0.66	0.75	…
D_5					0.92	0.63	0.69	0.56	…
D_6						0.91	0.71	0.49	…
D_7							0.93	0.60	…
D_8								0.92	…

TAB. 3 - Scores de similarité entre symboles modèles et symboles déformés

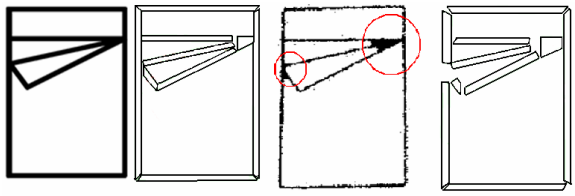


FIG. 6 - Influence des distorsions sur la représentation vectorielle obtenue

		Nombre de Modèles	Nombre de Test	Bien reconnus	Taux de reconnaissance
Rotation		50	150	150	100%
Changement d'échelle		50	100	100	100%
Bruit	Niveau-1	50	250	242	96.8%
	Niveau-2	50	250	238	95.2%
	Niveau-3	50	250	230	92.0%
Distorsion		15	100	94	94.0%

TAB. 4 - Résumé des performances de SimGraph

Il existe de nombreux facteurs qui peuvent influencer la performance d'une méthode de reconnaissance de symboles.

Un problème commun à l'ensemble des méthodes présentées à la compétition GREC 2003 est le passage à l'échelle. Les méthodes présentées fonctionnent bien pour un nombre limité de classes de symboles (de 5 à 20) mais leurs performances sont significativement dégradées pour une base de plus de 50 symboles différents. Notre méthode montre un très grand pouvoir de discrimination dans le cas d'images non bruitées avec un taux de reconnaissance de 100% pour un ensemble de 50 modèles de symboles.

Sur une telle base, les méthodes des autres participants affichaient un taux de reconnaissance de 84% et 92%. Ainsi, dans le cas de transformations affines, les résultats de la méthode proposée sont meilleurs que ceux de la majorité des participants à GREC'03.

Les taux de reconnaissance obtenus par notre méthode sur les images bruitées restent plus faibles. Ce résultat s'explique probablement par le fait que nous n'avons pas utilisé de prétraitement comme les filtrages ou la suppression des petits quadrilatères qui introduisent des sommets supplémentaires dans le graphe.

Le nombre de classes de symboles pour les tests de distorsion vectorielle est plus faible (seulement 15 symboles) parce que la méthode utilisée dans la compétition GREC03 pour produire des distorsions vectorielles fonctionne uniquement avec des symboles composés exclusivement de lignes droites (pas d'arcs de cercle, voir figure 8).

Notre taux de reconnaissance pour ces symboles déformés (table 4), même s'il n'est pas supérieur, est comparable à celui des meilleures méthodes utilisées pas les autres participants.

4 Conclusion

Dans cet article, nous avons présenté une méthode de reconnaissance de symboles graphiques linéaires représentés par des graphes structurels comportant à la fois des attributs numériques et symboliques. Lors de la construction des graphes, l'utilisation de quadrilatères préserve les parties significatives des symboles. L'utilisation de leur longueur relative comme attributs sur les sommets et de l'angle relatif entre quadrilatères voisins comme attribut sur les arcs rend nos graphes invariants à la rotation et au changement d'échelle. Une nouvelle mesure de similarité est également proposée pour comparer de tels graphes étiquetés et ainsi trouver le meilleur appariement possible entre 2 graphes. Les résultats obtenus sur des images ayant subi des transformations globales sont très encourageants. Pour tenir compte de la présence de bruit et de distorsions, notre approche est basée sur un appariement de graphes tolérants aux erreurs qui utilise des valeurs numériques comme attributs plutôt que des valeurs symboliques (comme les jonctions en T ou en L ou le parallélisme). De plus, ce système peut facilement apprendre de nouveaux symboles simplement en fournissant au système des images du nouveau modèle. Non seulement notre méthode calcule un degré de similarité pour la classification mais elle fournit aussi le meilleur appariement sommet à sommet entre graphes. Bien sûr, il reste encore beaucoup de travail à faire pour améliorer et enrichir notre méthode. Dans un avenir proche, nous avons prévu d'ajouter davantage d'informations numériques et symboliques dans les graphes par ajout d'attributs sur les sommets et les arcs pour reconnaître des symboles graphiques plus complexes notamment ceux comportant des formes pleines.

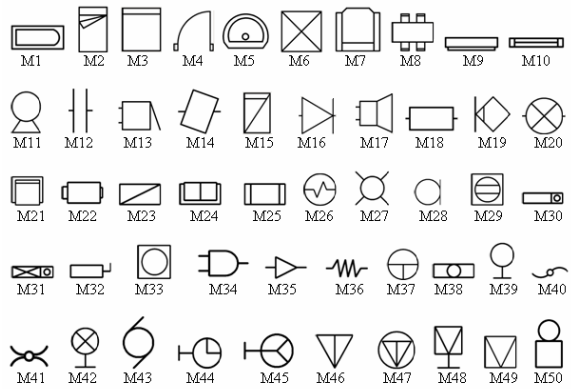


FIG. 7 - Première base d'images test

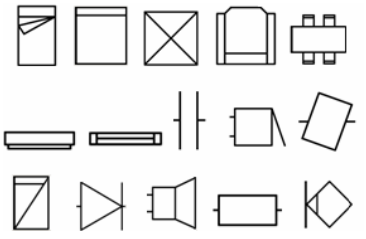


FIG. 8 - Deuxième base d'images test

Références

[BUN 99] BUNKE H., Error correcting graph matching. On the influence of the underlying cost function, *IEEE Transactions on pattern analysis and machine intelligence,* Vol. 21(9), 1999, pp. 917-922.

[CHA 03] CHAMPIN P. A., SOLNON C., Measuring the Similarity of Labelled Graphs, *Proceedings of the 5th International Conference on Case-Based Reasoning, Springer-Verlag*, 2003, pp. 80-95.

[COR 00] CORDELA L. P., VENTO M., Symbol recognition in documents : a collection of techniques ?, *International Journal on Document Analysis and Recognition,* Vol. 3(2), 2000, pp. 73 – 88.

[DOS 04] DOSCH P., LLADÓS J., Vectorial Signatures for Symbol Discrimination, *Selected papers from GREC'03, LNCS 3088, Springer Verlag,* 2004, pp. 154-165.

[GREC03] www.cvc.uab.es/grec2003/SymRecContest/, GREC'03 contest sample images

[HAN 94] HAN C., FAN K., Skeleton generation of engineering drawings via contour matching, *Pattern Recognition,* Vol. 27 (2), 1994, pp. 261- 275.

[LLA 01] LLADÓS J., MARTÍ E., VILLANUEVA J. J., Symbol Recognition by Error-Tolerant Subgraph Matching between Region Adjacency Graphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 23(10), 2001, pp. 1137 – 1143.

[LOP 01] LOPRESTI D., WILFONG G., Applications of Graph Probing to Web Document Analysis, *Proceedings of the First International Workshop on Web Document Analysis*, Seattle, WA, 2001, pp. 51-54.

[LUO 04] LUO Y., LIU W., Interactive Recognition of Graphic Objects in Engineering Drawings, *Selected papers from GREC'03, LNCS 3088, Springer Verlag,* 2004, pp. 128 -141.

[MES 96] MESSMER B., BUNKE H., Automatic Learning and Recognition of Graphic Symbols in engineering drawings, *In Kasturi, R., Tombre, K., eds.: Graphic Recognition: Methods and Applications. Springer LNCS 1072.* Berlin, 1996, pp. 123-134.

[RAM 98] RAMEL J.Y., VINCENT N., EMPTOZ H., A coarse vectorization as an initial representation for the understanding of line drawing images, *Springer LNCS 1389,* Berlin, 1998, pp. 48-57.

[RAM 00] RAMEL J.Y., VINCENT N., EMPTOZ H., A structural representation for understanding line drawing images. *International Journal on Document Analysis and Recognition,* Vol. 3(2), 2000, pp. 58 – 66.

[TER 04] TERRADES O. R., VALVENY E., Indexing Technical Symbols using Ridgelets Transform, *Selected papers from GREC'03, LNCS 3088, Springer Verlag,* 2004, pp. 176 -187.

[TOM 00] TOMBRE K., TABONNE S., Vectorization in graphics recognition : To thin or not to thin, *In Proceedings of 15th International Conference on Pattern Recognition,* Vol. 2. Barcelona, Spain, 2000, pp. 91-96.