



HAL
open science

Une méthode de localisation et de reconnaissance de symboles sans connaissance a priori

Daniel Zuwala, Salvatore Tabbone

► **To cite this version:**

Daniel Zuwala, Salvatore Tabbone. Une méthode de localisation et de reconnaissance de symboles sans connaissance a priori. Sep 2006, Fribourg, pp.127-131. inria-00119397v2

HAL Id: inria-00119397

<https://hal.science/inria-00119397v2>

Submitted on 5 Mar 2007

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

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

Une méthode de localisation et de reconnaissance de symboles sans connaissance a priori

Daniel Zuwala – Salvatore Tabbone

LORIA-INPL-Université de Nancy 2
Campus scientifique BP 23
54506 Vandoeuvre-lès-Nancy Cedex, France

daniel.zuwala@loria.fr, tabbone@loria.fr

Résumé : Nous présentons ici une méthode pour segmenter des symboles dans des documents graphiques, sans aucune connaissance a priori sur les symboles. Ce système se base sur une méthode de description structurelle qui va permettre de mettre en avant un certain nombre de régions pouvant contenir un symbole. A partir d'un découpage du document en chaînes de points clés, nous proposons de fusionner successivement les régions entre elles, en fonction d'un critère de densité. Cela nous permet de «reconstruire» les symboles et d'effectuer une reconnaissance à partir d'une requête présentée par l'utilisateur.

Mots-clés : segmentation, localisation de symboles, reconnaissance de symboles

1 Introduction

La reconnaissance de symboles a suscité beaucoup de travaux depuis de nombreuses années. Différents types de méthodes ont vu le jour, se basant soit sur des méthodes de template matching [VAL 99], soit sur des grammaires [FLA 95], soit sur des techniques issues de la programmation dynamique [MER 93] ou encore sur l'appariement de graphe [LEE 92]. L'une des raisons pour lesquelles la problématique de la reconnaissance de symboles est toujours ouverte est dû au fait qu'il existe un grand nombre et une grande variété de symboles, et qu'à part dans des cas spécifiques, il est souvent difficile de disposer d'un représentant explicite de tous les symboles.

En général, le cas d'utilisation classique est de pouvoir reconnaître un ensemble prédéfini de symboles dans un ensemble de documents. La segmentation des symboles est faite en utilisant des connaissances a priori sur le document [ARM 93], et la méthode de reconnaissance intègre en générale une connaissance sur les possibles déformations pouvant être rencontrées, ou alors met en oeuvre un apprentissage sur un ensemble conséquent d'exemples. De nombreuses techniques ont été développées, utilisant soit une description structurelle de l'objet, soit des techniques de reconnaissances statistiques [CHH 98, COR 00, LLA 02].

Dans cet article, nous allons considérer le problème sous un autre angle. Nous proposons une méthode capable de localiser et de reconnaître des symboles dans un ensemble de documents graphiques sans avoir de connaissance a priori sur les symboles.

Notre système est composé en deux parties. La première est basée sur une méthode structurelle qui va s'aider d'une décomposition du document en chaînes de points pour générer un certain nombre de symboles candidats. La deuxième partie est composée d'un système de requête qui permet à l'utilisateur de retrouver les symboles qui l'intéresse parmi les symboles candidats qui ont été segmentés lors de la phase de décomposition et ceci en se basant sur une description pixel des symboles.

L'article est organisé comme indiqué ci-après. Dans les sections 2 et 3, nous présentons la méthode qui permet de détecter des symboles dans des documents graphiques. Ensuite nous décrivons ce que nous avons utilisé pour reconnaître les symboles par rapport à la requête de l'utilisateur (section 4) et proposons une évaluation expérimentale de notre méthode (section 5). Finalement les conclusions et les perspectives de notre approche sont donnés en section 6.

2 Présentation de la méthode

2.1 Présentation générale

Notre méthode pour localiser les symboles est basée sur des chaînes de points extraites du document. Le but est d'isoler des sous-ensembles de chaînes de points qui pourraient représenter un symbole (§2.2).

Une approche simple pourrait être de considérer toutes les combinaisons possibles de chaînes. Cette approche est bien entendu inapplicable dans le cadre de grands documents. En effet, ceci est dû non seulement à l'explosion combinatoire qui en résulterait, mais aussi au fait que le grand nombre d'hypothèses qui en découlerait viendrait détériorer les performances en terme de reconnaissance.

Notre idée est d'alors de fusionner de manière itérative ces chaînes de points dans le but de reconstruire les symboles du document. Ce processus de fusion peut être représenté par un dendrogramme (§2.3).

2.2 Création d'un graphe de jonction

Pour commencer, nous décrivons le document graphique sous la forme d'un graphe de jonction. Au préalable, le document doit subir des pré-traitements préliminaires. Il doit être binarisé (Fig 1(a)), puis squelettisé (Fig 1(b)).

Une fois qu'il est squelettisé, nous allons repérer dans le squelette les points de jonctions (c'est-à-dire les points ayant

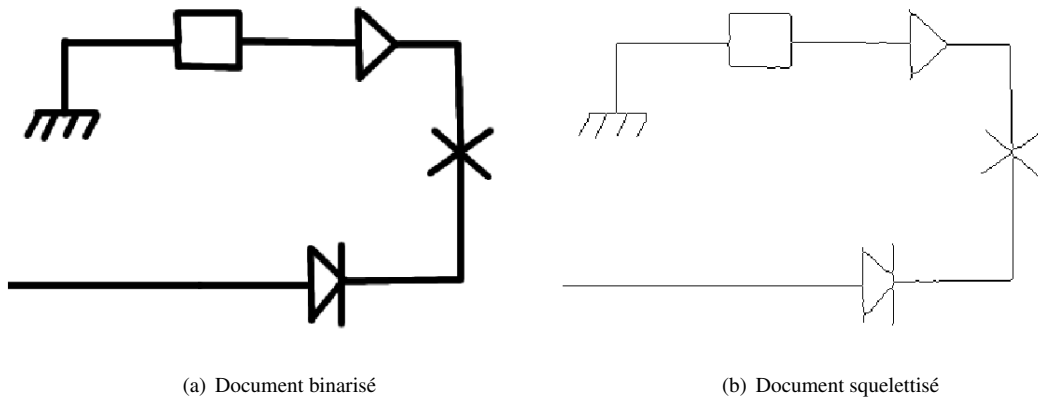


FIG. 1 – Pré-traitements du document (1ère et 2ème étape).

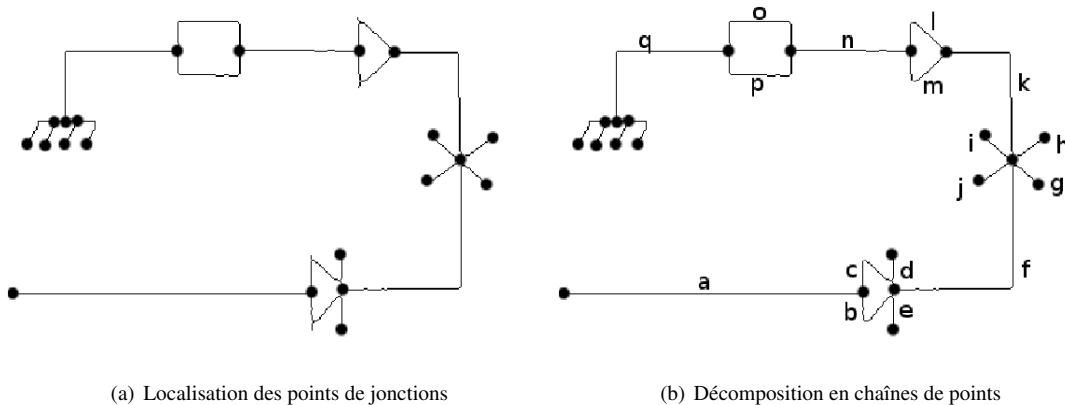


FIG. 2 – Pré-traitements du document (3ème et 4ème étape).

plus de 2 voisins), et les points terminaux (les points n'ayant qu'un seul voisin), ce qui est représenté en figure 2(a). La squelettisation peut malheureusement introduire de nombreux artefacts. Ainsi on trouve en pratique plus de points de jonctions que l'on voudrait, mais ce n'est heureusement pas préjudiciable pour la méthode que nous présentons ici. En effet les noeuds indésirables ne gêneront pas la reconstruction du symbole, mais augmenteront la complexité de la méthode.

Une fois que les points de jonctions et que les points terminaux ont été trouvés, nous pouvons alors retrouver les chaînes de points qui composent le document 2(b). Ces chaînes de points sont définies par un ensemble de points connectés ne possédant que 2 voisins, et dont les extrémités sont soit des points terminaux, soit des points de jonctions.

Nous pouvons alors construire un graphe de jonction. Les noeuds représentent les chaînes de points, tandis que deux chaînes de points connectées sont reliées par un arc. La figure 3 nous montre une partie du graphe de jonction, correspondant au document présenté à la figure 2(b).

2.3 Construction du dendrogramme

Une fois le graphe construit, il s'agit de fusionner successivement les noeuds de ce graphe de jonction pour construire le dendrogramme. La figure 4 montre une partie du dendrogramme pour un simple document composé de deux symboles simples (un carré et un triangle) reliés par une ligne.

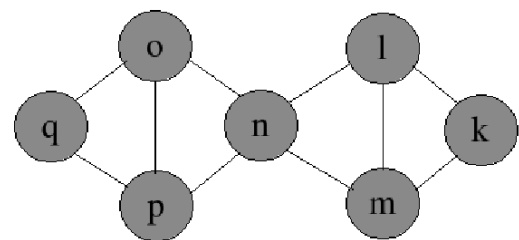


FIG. 3 – Une partie du graphe de jonction définie sur la Fig 1(b).

Les noeuds entourés par un rectangle (en pointillé sur la figure) sont ceux qui correspondent aux symboles correctement reconstruits.

2.4 Le critère d'agrégation

Pour guider la fusion, nous avons besoin d'un critère d'agrégation qui va nous permettre de décider quelles chaînes de points doivent être fusionnées à chaque étape.

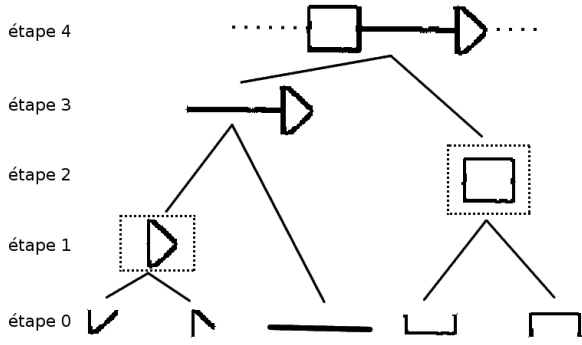


FIG. 4 – Un dendrogramme.

Il faut alors comprendre que chaque étape représentera un symbole potentiel. Bien définir ce critère est primordiale, car il faut être capable de fusionner en premier les chaînes de points appartenant aux symboles.

Pour définir un critère d'agrégation efficace, nous allons émettre les deux hypothèses suivantes quant à la nature des symboles :

- un symbole est un ensemble compact de chaînes de points connectées.
- les chaînes de points d'un symbole ont tendance à être convexes.

A partir de ces deux hypothèses, nous allons introduire deux mesures pour des chaînes de points. Une mesure sur la compacité qui sera notée m_c , et une mesure sur le degré de convexité qui sera notée m_f .

2.4.1 La compacité

Soit un ensemble de chaîne de points C , x_C^G le barycentre de tous les points composant C , n_C le nombre de points dans C , x_i le i -ème point C , et $d(x, y)$ la distance euclidienne entre les points x et y . La compacité $m_c(C)$ de C est donné par :

$$m_c(C) = \frac{\sum_{i=1}^{n_C} d(x_i, x_C^G)}{n_C} \quad (1)$$

2.4.2 Le degré de convexité

Pour mesurer le degré de convexité $m_f(C)$ d'un ensemble de chaîne de point C , nous calculons l'aire comprise par ces chaînes de points en suivant une direction verticale et horizontale. Un exemple de l'aire calculée est donné par la figure 5. $m_f(C)$ de C est alors définie par le rapport entre l'aire trouvée et l'aire du rectangle englobant C .

$$m_f(C) = \frac{\text{aire} - \text{trouvée}(C)}{\text{aire} - \text{rectangle} - \text{englobant}(C)} \quad (2)$$

$m_f(C)$ est normalisé dans $[0, 1]$.

2.4.3 Le critère d'agrégation

Nous pouvons alors combiner ces deux critères en une seule mesure :

$$m(C) = m_c(C)(1 - m_f(C)) \quad (3)$$



(a) la chaîne de point (b) l'aire considérée

FIG. 5 – Calcul du degré de convexité.

Ainsi pour un ensemble de chaînes de points C , plus ces chaînes de points seront compactes et plus $m_c(C)$ sera faible. De même, plus ces chaînes de points auront tendance à être convexes, plus $m_f(C)$ sera élevé, et donc plus $m(C)$ sera faible. Pour résumer, quand $m(C)$ est faible cela correspond aux hypothèses que nous avons introduites sur la caractérisation des symboles. On va donc utiliser cette mesure dans la construction du dendrogramme.

2.5 Algorithme

Nous présentons ci après, l'algorithme général pour la construction du dendrogramme.

Construire le graphe de jonctions à partir du document squelettisé.

Pour chaque chaînes de points C_i dans le graphe **faire**

 calculer $m(C_i)$

Fin Pour

Pour chaque arcs $e(C_i, C_j)$ dans le graphe **faire**

 calculer le $m(C_i \cup C_j)$ correspondant
 placer $e(C_i, C_j)$ dans une liste Q triée par $m(C_i \cup C_j)$ croissant

Fin Pour

Tant que (Q n'est pas vide) **faire**

 retirer le premier élément $e(C_i, C_j)$ de la liste Q
 créer un nouveau noeud qui sera associé à $C_i \cup C_j$
 mettre à jour les différents arcs du graphe
 retrier la liste Q par $m(C_i \cup C_j)$ croissant

Fait

3 Améliorations de la segmentation

La mesure introduite peut néanmoins être mise en défaut dans des documents contenant des symboles fortement connectés. Ceci est dû à une vue trop locale, qui nous empêche d'agréger des chaînes de points qui auraient pourtant donné une fusion plus intéressante plus loin dans le dendrogramme.

Pour améliorer ceci nous avons introduit une méthode de prévision, qui au lieu de se baser simplement sur les chaînes de points qui vont effectivement fusionner, se base également sur les conséquences que cette fusion peut avoir dans les étapes suivantes. En d'autres termes, nous allons considérer non seulement les chaînes de points courantes mais aussi les chaînes de points qui y sont connectées avec une profondeur maximum n_{max} donnée.

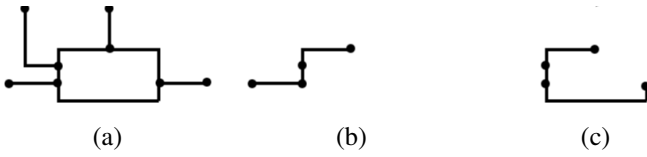


FIG. 6 – Symbole trouvé avec un vue locale (b) ou globale (c).

Ainsi pour chaque $n \in [0..n_{max}]$, nous calculons le critère m pour chaque ensemble de chaînes de points. Soit m_{min}^i la mesure minimum à la profondeur i . Nous combinons alors ces minima en une seule mesure :

$$m^{n_{max}}(C_1, C_2) = \sum_{k=0}^{n_{max}} \alpha(k) m_{min}^k(C_1, C_2) \quad (4)$$

où α est un facteur d'atténuation.

La figure 6 montre l'impact de cette mesure. La figure 6(a) représente le symbole original dans le document, et la figure 6(b) est la fusion résultante que l'on trouve en gardant une vue locale. La fusion résultante que l'on trouve avec une vue plus globale de l'approche est présentée en Fig 6(c).

Bien entendu, cette amélioration a un coût en terme de calcul, qu'il est possible de réduire en adoptant une stratégie de type "diviser pour conquérir". Nous donnons à titre indicatif en figure 10, le temps de traitements d'un documents avec ou sans cette stratégie. Nous voyons alors que le gain est significatif à partir d'une profondeur maximum de 2. Néanmoins les temps de traitements de cette approche sont 20 fois supérieurs à la vue locale (correspond à $n=0$ dans la figure 10).

4 Etape de reconnaissance

Une fois tous les symboles candidats générés, le système passe à l'étape de reconnaissance. L'aspect structurel des symboles candidats n'est plus considéré ici mais uniquement sur la répartition pixel. L'utilisateur peut alors soumettre une requête directement issu d'un des documents analysés.

Les symboles candidats sont alors filtrés en utilisant des critères géométriques simples par rapport à la requête. En effet notons w et h , la largeur et la hauteur du symbole requête. Pour tous les symboles candidats ayant une largeur w_i et une hauteur h_i , nous ne gardons que ceux qui répondent au critère suivant :

$$0.5 \frac{w}{h} < \frac{w_i}{h_i} < 1.5 \frac{w}{h}$$

Ensuite, un descripteur pixel est calculé pour tous les symboles restants. Nous avons choisit l'ART (Angular Radial Transform) [KIM 99]. Ce descripteur est relativement rapide à calculer et est robuste dans une certaine mesure aux déformations et à la distorsion. On peut bien entendu utiliser n'importe quel autre descripteur.

5 Expérimentations

5.1 La méthode d'expérimentation

La méthode a été testée sur une base d'images contenant 30 documents de schémas électriques d'avion. La taille

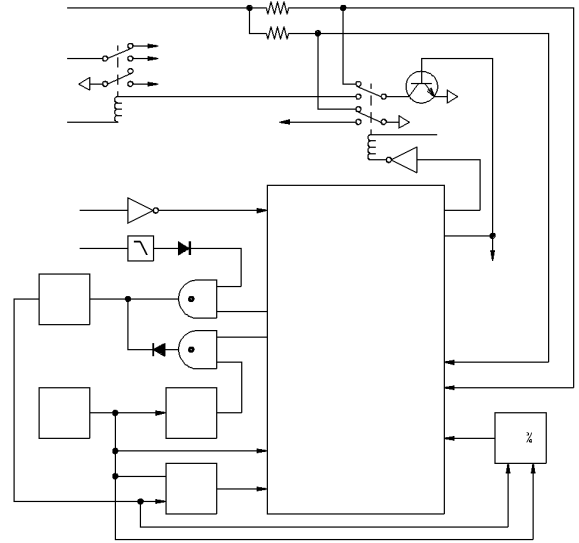


FIG. 7 – Exemple de document traité.

moyenne de ces documents est de 3000x3000 pixels. La figure 7 montre une partie de ces documents, tandis que la figure 8 montre les symboles extraits à différentes étapes données. Il faut noter qu'une étape de séparation texte/graphique a été appliquée afin de réduire la complexité de la méthode. A partir de ces documents une vérité terrain a été construite, rassemblant 108 symboles en 7 classes différentes.

étape		étape		étape	
81		175		627	
88		195		628	
107		199		629	
138		623		719	

FIG. 8 – Exemples de reconstruction du document figure 7 à plusieurs étapes données.

Nous avons utilisé les taux de précisions et de rappels pour évaluer la performance de la méthode. La figure 9 montre les résultats obtenus en utilisant le descripteur ART. Ceux-ci semblent prometteurs puisque nous atteignons une précision de 80% pour un taux de rappel de 50%, et qu'au

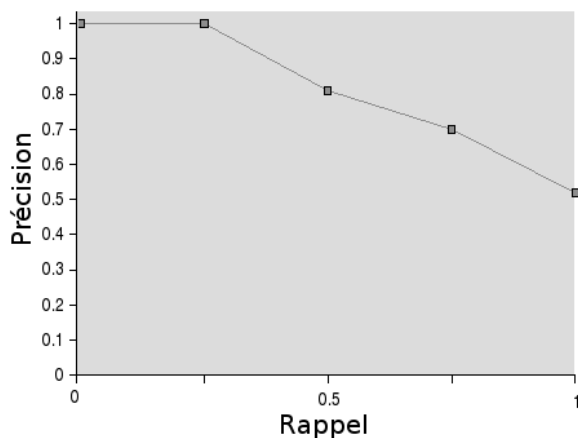


FIG. 9 – Précision et rappel avec l'ART.

final nous obtenons une précision de 50% pour un taux de rappel de 100%.

N	0	1	2
sans "diviser pour conquérir"	2	7	51
avec "diviser pour conquérir"	2	6	40

FIG. 10 – Temps de traitements d'un document en secondes, pour une profondeur maximum donnée, avec ou sans la stratégie "diviser pour conquérir".

5.2 Résultats et observations

La décomposition de ces 30 documents a pris environ 10 min en n'utilisant que la méthode locale, sur un Pentium IV 2.6GHz. Environ 85,000 symboles potentiels ont été isolés, et seulement un tiers furent gardés après un filtrage sur des critères géométriques simples.

Ces expérimentations révèlent que :

- Après filtrage des symboles candidats, il en restent un nombre tout de même très important. Pour 108 symboles à reconnaître, nous avons plus 28000 symboles candidats.
- Le critère d'agrégation que nous avons défini semble efficace, puisque sur les 108 symboles à reconnaître, seulement 4 n'ont pas été reconstruit correctement, et donc ne peuvent être reconnus par la suite.
- La vue globale nous permet de retrouver les 4 symboles qui avaient été manqués avec la vue locale, mais au détriment d'une complexité qui demeure importante, même avec des heuristiques de parcours (comme le montre la figure 10).

6 Conclusion

Dans cet article, nous avons introduit une nouvelle méthode pour localiser des symboles dans un document graphique sans connaissance a priori sur les symboles. Les seules hypothèses que nous avons posées considèrent les symboles comme une ensemble de chaînes de points com-

pactes et plutôt convexes. En utilisant ensuite un descripteur photométrique, il est alors possible d'avoir un système complet pour reconnaître tout type de symboles dans les documents. Les résultats sont prometteurs, et nous envisageons de tester la méthode sur une base de documents volumineuse et plus diverses. Il est aussi envisagé de trouver des méthodes de filtrages plus évoluées afin de réduire le nombre de symboles candidats retourné par la méthode.

Références

- [ARM 93] ARMAND J.-P., Musical Score Recognition : A Hierarchical and Recursive Approach, *Proceedings of 2nd International Conference on Document Analysis and Recognition, Tsukuba (Japan)*, 1993, pp. 906–909.
- [CHH 98] CHHABRA A. K., Graphic Symbol Recognition : An Overview, TOMBRE K., CHHABRA A. K., Eds., *Graphics Recognition—Algorithms and Systems*, vol. 1389 de *Lecture Notes in Computer Science*, pp. 68–79, Springer-Verlag, April 1998.
- [COR 00] CORDELLA L. P., VENTO M., Symbol recognition in documents : a collection of techniques ?, *International Journal on Document Analysis and Recognition*, vol. 3, n° 2, 2000, pp. 73–88.
- [FLA 95] FLASIŃSKI M., Use of graph grammars for the description of mechanical parts, *Computer-Aided Design*, vol. 27, n° 6, 1995, pp. 403–433.
- [KIM 99] KIM W.-Y., KIM Y.-S., A new region-based shape descriptor, *TR 15-01, Pisa (Italy)*, December 1999.
- [LEE 92] LEE S.-W., Recognizing Hand-Drawn Electrical Circuit Symbols with Attributed Graph Matching, BAIRD H. S., BUNKE H., YAMAMOTO K., Eds., *Structured Document Image Analysis*, pp. 340–358, Springer-Verlag, Heidelberg, 1992.
- [LLA 02] LLADÓS J., VALVENY E., SÁNCHEZ G., MARTÍ E., Symbol Recognition : Current Advances and Perspectives, BLOSTEIN D., KWON Y.-B., Eds., *Graphics Recognition – Algorithms and Applications*, vol. 2390 de *Lecture Notes in Computer Science*, pp. 104–127, Springer-Verlag, 2002.
- [MER 93] MERLET N., ZERUBIA J., Classical mechanics and reads detection in SPOT images, *Research rapport*, April 1993, INRIA.
- [VAL 99] VALVENY E., MARTÍ E., Application of Deformable Template Matching to Symbol Recognition in Hand-written Architectural Drawings, *Proceedings of 5th International Conference on Document Analysis and Recognition, Bangalore (India)*, September 1999, pp. 483–486.