

APPARIEMENT DE DEUX GRAPHS
STRUCTURELS QUELCONQUES
POUR LA RECONNAISSANCE DE
LETTRES MANUSCRITES

PAIRING ANY TWO STRUCTURAL
GRAPHS
FOR HANDPRINTED LETTER
RECOGNITION

**Patrice DARGENTON, Nicole VINCENT, Hubert
EMPTOZ**

LISPI-RF&D, INSA LYON

*Laboratoire d'Informatique des Systèmes de
Production Industrielle*

*Equipe de Reconnaissance des Formes et Diagnostic
Institut National des Sciences Appliquées de LYON*

Bat. 403, 20 Av. A. EINSTEIN, 69621

VILLEURBANNE (FRANCE)

Tel (33) 72 43 80 96 - Fax (33) 72 43 85 29

Résumé

Nous proposons dans cet article une méthode d'appariement de deux graphes structurels quelconques en vue de la reconnaissance de lettres manuscrites. Ces graphes sont construits pour représenter les lettres manuscrites omniscriteurs dans le cadre de la reconnaissance hors ligne de l'écrit, en réduisant l'information contenue dans l'image au minimum nécessaire pour modéliser précisément la structure des caractères. Nous montrons que la reconnaissance structurelle permet de mettre en évidence les ambiguïtés naturelles entre les lettres, en offrant un indice de reconnaissance utile dans un contexte lexical. La robustesse de la méthode est testée dans le cas d'une application utilisant des caractères bâtons.

Mots clés :

Écriture manuscrite omniscriteur
Reconnaissance hors ligne de lettres
Appariement de graphes structurels

Abstract

We propose in this paper a pairing method for any two structural graphs for handprinted letter recognition. These graphs are built in order to reduce information at the minimum necessary to describe accurately the structure of script letters, in the case of off-line omniscaptor handwriting recognition. We show that the structural recognition is able to exhibit the natural ambiguity between letters, and provide a recognition degree which is useful in a lexical context. The robustness of the proposed method is tested in the case of a handprinted character recognition system.

Keywords :

Omniscaptor handwriting
Off-line recognition of letters
Structural graph pairing

INTRODUCTION

Dans le domaine de la reconnaissance omniscriteur des mots manuscrits cursifs (hors ligne), on distingue deux principales approches :

La première approche concerne les méthodes qui consistent, à partir d'une hypothèse sur le mot, obtenue par une reconnaissance globale, à vérifier la segmentation et la reconnaissance des lettres (méthodes descendantes) ;

La seconde approche concerne les méthodes analytiques qui se basent sur des hypothèses de lettres pour aboutir à la reconnaissance du mot (méthodes ascendantes).

La principale critique que l'on peut formuler à l'égard de la première approche est que la génération d'hypothèses sur le mot n'exploite pas suffisamment la notion de lettre (l'extraction de primitives est globale dans le mot) alors que dans le cas des documents imprimés, la seule reconnaissance des lettres suffit pour aboutir à la reconnaissance complète du mot.

Quant à la deuxième approche, elle conduit souvent à une indécision car, pour tenir compte des fortes variations propres aux lettres manuscrites, naturellement ambiguës, on doit utiliser une description trop approximative des lettres. En outre, il est

évidemment difficile de segmenter les lettres avant de les avoir reconnues.

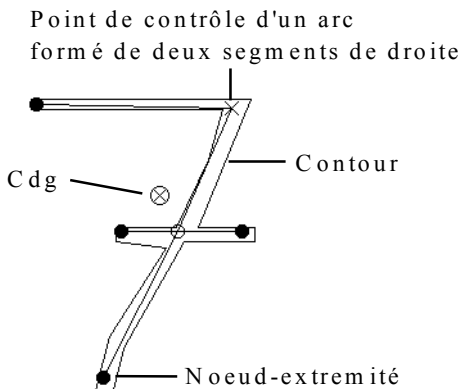
Partant de ce constat, nous nous efforçons de réduire le plus possible l'incertitude au niveau de la lettre, grâce à une reconnaissance de type structurel : pour cela, nous avons conçu un algorithme de construction d'un graphe structurel représentatif d'une forme [DARGENTON 93]. Ce graphe, qui est construit sur toutes les lettres (ou entités proches telles que n-gram ou tronçon de lettre) que l'on peut extraire du mot, a pour but la réduction de l'information contenue dans l'image au minimum nécessaire pour décrire précisément la lettre manuscrite.

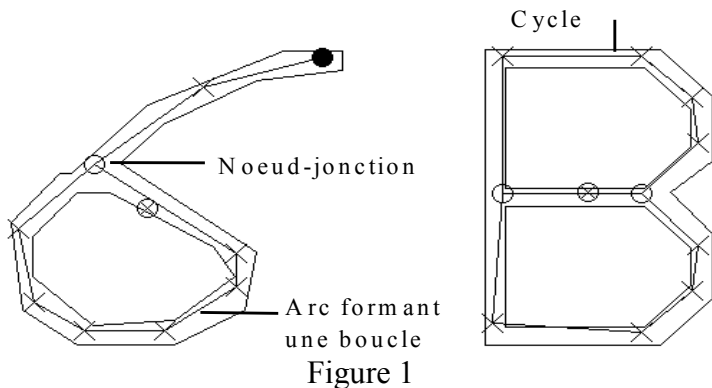
A partir de ce graphe, notre objectif est de réaliser un appariement structurel avec tous les graphes des lettres du dictionnaire de formes afin de reconnaître la plus proche. Dans cet article, nous nous écartons de la reconnaissance des mots cursifs pour présenter une méthode d'appariement de deux graphes structurels quelconques appliquée à la reconnaissance de lettres manuscrites isolées.

1. DESCRIPTION DES GRAPHES OBTENUS

Chaque graphe est constitué d'un ensemble de noeuds reliés par un ou plusieurs arcs. On distingue seulement deux types de noeuds : les noeuds-extrémités, qui ne sont reliés qu'à un seul arc, et les noeuds-jonctions qui sont les points de conjonction de trois arcs ou plus (voir figure 1).

Un arc est constitué par une chaîne de segments de droites délimités par des points de contrôle obtenus par approximation polygonale. Un arc forme une boucle lorsque ses extrémités sont confondues en un même noeud (voir le graphe du '6'). Plusieurs arcs peuvent former un cycle (voir le graphe du 'B').





Dans une approche utilisant une grammaire de graphes [WONG 87], le graphe traduit des relations de proximité ("à côté", "au-dessus"...) entre les segments de l'image.

Le modèle ARG (Attributed Relational Graph) peut par exemple être construit à partir des arcs de cercle approchant le tracé des caractères [d'ACIERNO 91]. Le niveau de description obtenu permet l'exploitation de la méthode dans une zone de continuité entre le domaine de l'imprimé et des caractères manuscrits bien écrits.

Le modèle RAG (Region Adjacency Graph), qui traduit des relations entre trois types de région (région régulière, extrémité et jonction), aboutit à une description comparable [NISHIDA 92].

D'une façon générale, les noeuds d'un graphe marquent une primitive structurelle, et les arcs expriment les relations entre ces noeuds. Dans notre cas, la signification du graphe est une expression directe de la connectivité d'un ensemble de points remarquables de l'image.

On doit en premier lieu définir un point de référence par rapport auquel va être calculée une distance entre les noeuds des deux graphes. Le barycentre des noeuds du graphe est la méthode la plus simple pour obtenir un centre de gravité (cdg) de la forme.

La méthode que nous avons retenue est la suivante : le cdg est le barycentre de tous les milieux des segments pondérés par leur longueur. Le cdg ainsi défini est plus stable dans le cas général où les deux graphes sont quelconques. Les cdg ont été représentés par une croix encerclée sur la figure 1.

D'autre part, tous les graphes sont recadrés à l'échelle de telle sorte que la diagonale du rectangle englobant le caractère soit toujours de même taille. Sur les figures, la taille des noeuds rend compte de ce recadrage.

2. RECHERCHE D'UN ISOMORPHISME OPTIMAL DE GRAPHE

L'exploitation du graphe structurel obtenu commence par la recherche d'un isomorphisme optimal entre deux graphes, celui de la lettre à reconnaître et celui d'une des lettres de l'alphabet de référence, afin d'établir des points de comparaison. Ce problème est proche de celui rencontré en stéréovision [BACKER 87] dans lequel il s'agit de trouver la meilleure mise en correspondance entre les graphes des deux images de la vue stéréoscopique. Une autre formulation du problème de l'appariement est rencontrée dans la vision avec une seule caméra mais avec la prise en compte d'une séquence d'images dans le temps [BOUKARRI 91]. En vision robotique, les contraintes d'appariement se trouvent renforcées en tenant compte simultanément de la stéréoscopie et du suivi du mouvement dans le temps [CHEBARO 91].

Les méthodes d'appariement diffèrent par deux points principaux qui sont l'élément de base (la primitive) à apparier et la procédure d'appariement.

Cet élément de base est généralement le segment de droite, mais des primitives plus élaborées telle que l'arc de cercle, ou moins élaborées telle que le niveau de gris d'un pixel et de ses proches voisins, sont

également employées. Le degré de consistance définissant la contrainte de l'appariement est lié à cet élément : lorsque celui-ci est un polygone, la vérification de la consistance des appariements de ses cotés entraîne une contrainte supplémentaire qui assure une cohérence supérieure à la prise en compte des segments isolément, mais au prix d'une plus grande complexité.

L'utilisation de l'arc du graphe défini au paragraphe 1, qui est une séquence de segments (pouvant être remise en cause et segmentée), constitue une approche intermédiaire comme nous le montrerons en détail dans cette communication.

En ce qui concerne la procédure d'appariement pour la recherche d'un isomorphisme optimal de graphe, cinq techniques principales peuvent être dégagées :

- La relaxation (relaxation discrète ou probabiliste, propagation de contraintes, technique de recuit simulé) ;

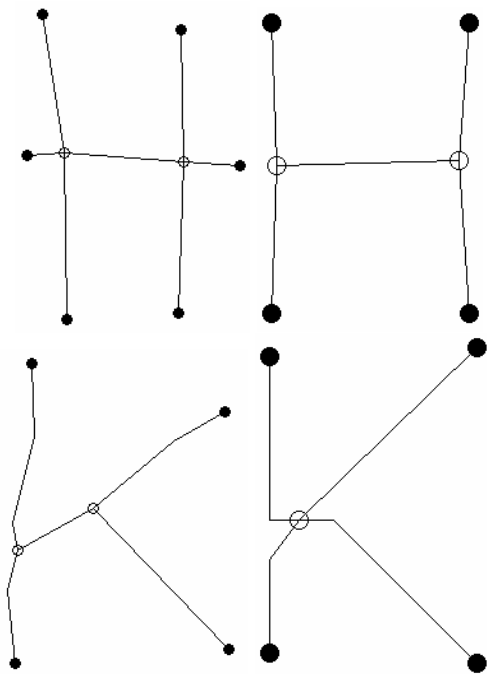
- La corrélation (calcul de corrélation des primitives) ;

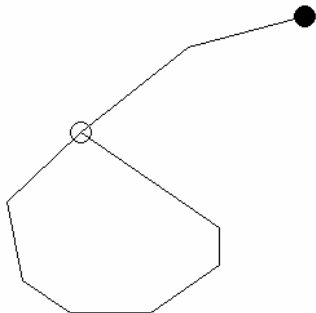
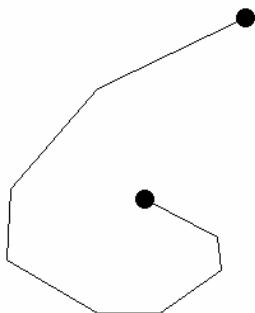
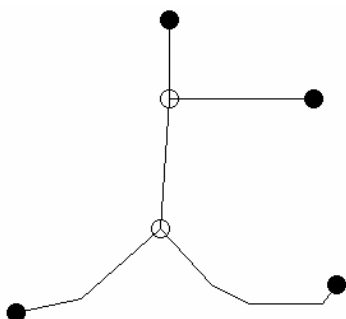
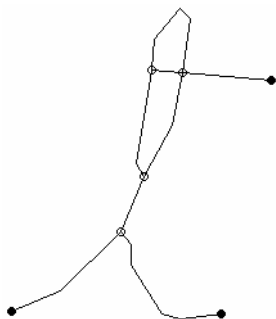
- La recherche heuristique (prédiction et vérification d'hypothèses) ;

- La recherche combinatoire (recherche dans un arbre, c'est-à-dire une structure arborescente de données) ;
- La programmation dynamique (calcul d'une distance d'édition).

Dans les trois premières techniques, la mise en correspondance est effectuée en deux dimensions (2D), la recherche dans un arbre est intermédiaire entre 2D et 1D, tandis que la programmation dynamique est une technique d'appariement spécifique aux séquences 1D.

On ne peut formuler le problème de la recherche de l'isomorphisme optimal de graphe comme une recherche dans un arbre [YOU 84], car cette procédure n'est pas utilisable dans le cas où le nombre de noeuds des deux graphes à apparier est différent. En effet, cette recherche implique des fusions de noeuds et d'arcs, ainsi que des fragmentations, éliminations et concaténations d'arcs du graphe comme l'attestent les exemples de couples suivants que l'on se propose d'apparier :





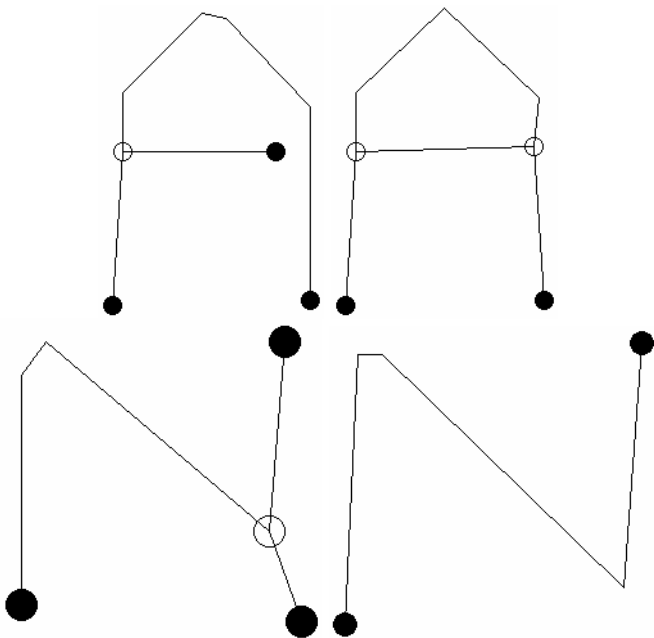


Figure 2

On serait tenté de prévoir l'ensemble des graphes potentiels (variantes de lettres) pour un caractère donné dans un dictionnaire de lettres. Si cela semble judicieux pour le '6' étant donné que les deux exemples de '6' sont fondamentalement distincts sur le plan structurel, ce choix ne serait pas raisonnable pour le 'A' et le 'N' dans lesquels la divergence de structure est seulement due à un faible bruit ('A') ou à une légère

transformation ('N'). On perdrait alors l'intérêt de la reconnaissance de la structure, et on ne se limiterait plus, alors, qu'à une procédure vérifiant l'identité des graphes avec une décision binaire (OUI / NON).

L'appariement est un problème simple lorsque les structures sont identiques ou proches ; il suffit dans ce cas de procéder à l'appariement des noeuds des deux graphes à l'aide d'un calcul de corrélation :

On calcule, dans une première phase, la distance spatiale entre les noeuds des deux formes, en superposant par translation leurs centres de gravité (origine du repère). On normalise cette distance en utilisant la diagonale du rectangle englobant la lettre. Cette distance spatiale est ensuite pondérée par la différence des degrés de chaque noeud, c'est-à-dire le nombre d'arcs qui y sont connectés. Enfin, le couple de noeuds correspondant à la distance minimale est mémorisé.

Dans un deuxième temps, on procède à l'appariement des arcs des deux graphes en fonction des noeuds correspondants :

Pour chaque arc de la première forme, on recherche les correspondants de ses noeuds-extrémités dans l'autre forme.

Puis, on recherche tous les arcs de l'autre forme reliant ces correspondants.

Ensuite, pour chacun de ces arcs, on calcule un coût d'appariement qui tient compte, d'une part, de la somme des coûts d'appariement des points de contrôle de chaque arc relativement au centre de gravité de leur arc, et d'autre part, de la distance entre les centres de gravité des arcs par rapport aux centres de gravité correspondants des deux formes.

Enfin, on mémorise le couple d'arcs de moindre coût. Si l'arc n'a aucun correspondant dans l'autre forme, ou bien si son coût d'élimination est inférieur au moindre coût, alors il est éliminé.

En revanche, lorsque les structures sont différentes (voir les graphes des caractères '6', 'A' et 'N' sur la figure 2), cette technique est mise en échec, bien que les formes soient très proches d'aspect. Cette constatation nous amène à conclure que l'appariement doit être accompli par une recherche à deux niveaux distincts : une première recherche de correspondance évidente sur les chaînes entières de segments, puis une seconde recherche de similarité de sous-chaînes de segments.

La première assure la robustesse de l'appariement pour des structures proches, tandis que la seconde permet de poursuivre en profondeur la recherche seulement si nécessaire. Ceci évite

l'explosion combinatoire due au grand nombre de segments de droites qui composent l'image.

3. CALCUL DE LA DISTANCE D'ÉDITION ENTRE LES ARCS

Nous allons définir dans cette étape une distance d'édition entre tous les couples d'arcs possibles des deux formes, sans tenir compte des noeuds les délimitant. Pour le calcul de cette distance, on utilise une méthode classique (programmation dynamique à l'aide de la distance de LEVENSHTEIN) [YOU 84] de mise en correspondance de caractères d'une chaîne linéaire en respectant l'ordre séquentiel. Dans notre cas, les caractères représentent les segments de droites qui forment les arcs reliant les noeuds.

Cette méthode suppose définis un coût de substitution de caractères, un coût d'insertion et un coût de destruction pour chaque caractère dans la chaîne.

Le coût de substitution que nous avons calculé dépend d'une part, de la différence d'orientation des segments de droites - différence comprenant la direction et le sens des segments et calculée à l'aide d'un simple produit scalaire - et d'autre part, de la valeur absolue de la différence de leurs longueurs.

Coût d'Orientation : (v_1 et v_2 vecteurs unitaires)

$$CO = 1 - \vec{v}_1 \cdot \vec{v}_2$$

Coût de différence de Longueur :

$$CL = \text{abs} (l_1 - l_2)$$

Le coût global de substitution de deux segments est une combinaison linéaire de CL et CO comprenant une saturation afin de pénaliser les trop grandes différences de longueur quelle que soit l'orientation ou bien les trop grandes différences d'orientation quelle que soit la longueur.

Le coût de destruction d'un segment prend en compte seulement sa longueur. Le coût d'insertion étant réciproque du coût de destruction, il n'a pas été utilisé dans notre étude.

Pour que cette technique puisse mettre en évidence des similarités entre les chaînes proches d'aspect, mais dont les nombres de segments diffèrent, nous y avons ajouté une procédure de concaténation de segments adjacents en calculant un coût correspondant (figure 3).

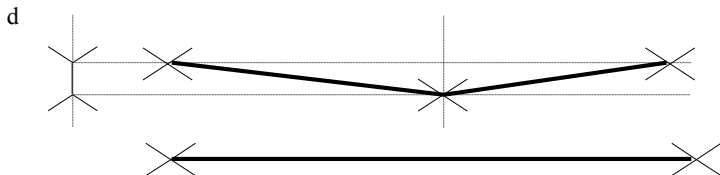


Figure 3

Ce coût est simplement la distance d du segment résultant, au point de connexion des deux petits segments à partir desquels il a été formé. Ce coût est nul si la distance d est nulle, mais ce cas ne peut pas être rencontré si l'approximation polygonale des arcs a été accomplie avec succès lors de la construction du graphe structurel [DARGENTON 93].

Le nombre maximum de concaténations successives a été fixé à deux (MAX_CONCAT), c'est-à-dire trois segments au plus peuvent être remplacés par un seul.

L'algorithme d'édition de chaînes avec concaténations multiples est récapitulé ci-dessous :

```

/* Initialisation */
d[0][0]=0 ;
for( i=1 ; i<=long_i ; i++ )
d[i][0]=d[i-1][0] +
cout_des_arc(chaine[0][i-1]) ;

```

```

for( j=1 ; j<=long_j ; j++ )
d[0][j]=d[0][j-1] +
cout_des_arc(chaine[1][j-1]) ;
/* Calcul de la distance d'éditoin */
for( i=1 ; i<=long_i ; i++ )
for( j=1 ; j<=long_j ; j++ )
{
    d1=d[i-1][j-1] +
cout_sub_arc(chaine[0][i-
1],chaine[1][j-1]) ;
    d2=d[i-1][j] +
cout_des_arc(chaine[0][i-1]) ;
    d3=d[i][j-1] +
cout_des_arc(chaine[1][j-1]) ;

    for( k=0 ; k<MAX_CONCAT-1 ; k++ )
    if (i-2-k>=0)
    di[k]=d[i-2-k][j-1] +
cout_concat[0][i-1][k] +
    cout_sub_arc(liste_concat[0][i-
1][k],chaine[1][j-1]) ;

    for( k=0 ; k<MAX_CONCAT-1 ; k++ )
    if (j-2-k>=0)
    dj[k]=d[i-1][j-2-k] +
cout_concat[1][j-1][k] +
    cout_sub_arc(chaine[0][i-
1],liste_concat[1][j-1][k]) ;

    for( k=0 ; k<MAX_CONCAT-1 ; k++ )
    d[i][j]=min(d1,d2,d3,di[k],dj[k]) ;

```

Chaîne[f][i] pointe sur l'indice du segment i de la chaîne de la forme f, dans la liste générale de tous les segments de cette forme (long_i est sa longueur) ;

Liste_concat[f][i][k] pointe sur l'indice du segment concaténé i, construit à partir du segment i jusqu'au segment i+k+1, dans la liste de tous les segments concaténés de cette chaîne de la forme f ;

Coût_concat[f][i][k] est le coût de concaténation correspondant.

Lors de la mise au point de cette procédure, on a tenté d'inclure le coût de translation de chaque extrémité de segment dans la routine de calcul du coût de substitution ainsi que dans celle de calcul du coût de destruction. L'expérience a montré que ce coût de translation devant être combiné avec le coût de longueur et d'orientation, les similarités entre les segments étaient alors occultées par ce coût. En affaiblissant la contribution de ce coût par rapport aux deux autres (CL et CO), le problème inverse se manifestait par la détection de ressemblances entre des chaînes opposées dans les deux formes par rapport au cdg, de sorte que la perception des lettres manuscrites en était altérée.

La décision a donc été prise de ne prendre en compte le coût de translation que seulement après le calcul de la distance totale d'édition des deux chaînes, en cumulant la somme des coûts de translation de chaque segment pondérés par la longueur du segment.

En résumé, pour chaque couple de chaînes, on retient deux coûts distincts qui sont le coût final d'appariement des chaînes (CA) issu du calcul de la distance d'édition, ainsi que le coût final de translation des chaînes (CT). Ces deux coûts seront combinés par la suite en utilisant une pondération et une saturation, de la même manière que dans le calcul du coût final de substitution de segments.

4. PROCÉDURE D'APPARIEMENT DES ARCS

Pour accomplir cette étape, on utilise le résultat du calcul de la distance d'édition entre les arcs, à l'endroit puis à l'envers, en ne conservant que le coût le plus faible. Le coût minimum d'appariement de chaque arc des deux formes vers tous les arcs de l'autre forme est mémorisé dans un tableau, avec le sens d'appariement correspondant.

La procédure finale d'appariement des arcs s'effectue en deux étapes. Au cours de la première étape est validé l'appariement de tous les couples de chaînes

dont l'une est à la distance minimum de l'autre et réciproquement. Dans la seconde étape, on est donc obligé d'apparier les arcs restants, ceux dont seulement une des deux distances est minimum.

Pour cela, les coûts sont classés dans l'ordre croissant (les coûts des deux formes sont en concurrence) et les couples sont alors validés suivant ce classement afin de commettre le moins possible d'erreurs.

Exceptionnellement, cette procédure est mise en défaut lorsque deux arcs sont réciproquement à la distance minimale l'un de l'autre, mais que l'appariement n'est pas optimal sur l'ensemble du graphe. Cependant, ce problème est résolu en recherchant l'ensemble des permutations de couples d'arcs avec deux autres arcs non appariés qui offrent un coût inférieur à l'appariement réciproque minimal.

Quoiqu'il en soit, ce cas est marginal et la solution adoptée ne modifie pas sensiblement les résultats satisfaisants obtenus.

5. RECHERCHE DES SIMILARITÉS DE SOUS-CHAÎNES

Cette étape constitue la partie sensible de l'ensemble de la méthode d'appariement des graphes

structurels. Elle consiste à isoler une partie d'une chaîne, c'est-à-dire une sous-chaîne, qui soit fortement similaire à une (sous-) chaîne appariée au cours de l'étape précédente, alors que le restant des deux chaînes est dissemblable. La recherche des similarités ne s'effectue donc qu'entre des couples de chaînes appariées et non sur l'ensemble de leurs combinaisons. La restriction de cette recherche en profondeur dans un nombre minimum de cas offre un compromis intéressant entre l'efficacité et la robustesse de la méthode.

Le succès de l'appariement des graphes dépend beaucoup de la pertinence de la coupure envisagée. Dans ce sens, on va chercher à maximiser le critère suivant pour chacune des deux chaînes :

Isoler le maximum de segments consécutifs de coût d'appariement faible dans la première partie de la chaîne, l'autre partie contenant le maximum de segments consécutifs de coût fort.

Le coût d'appariement détaillé pour chaque segment est issu du calcul de la distance d'édition des deux chaînes ; le critère est obtenu à partir des deux indices C_1 et C_2 suivants :

$$C_1[s] = (1 - \sum_{i=1}^{segment} |i| / sca) \cdot (\sum_{i=1}^{segment} |i|) / sl$$

$$sl = \sum_{i = \text{segment}}^{\text{segment}} \frac{|N|}{|i|}$$

$$C_2[s] = (1 - \sum_{i = \text{segment}}^{\text{segment}} \frac{|i|}{sca}) \cdot (\sum_{i = \text{segment}}^{\text{segment}} \frac{|i|}{sl})$$

$$sca = \sum_{i = \text{segment}}^{\text{segment}} \frac{|N|}{|i|}$$

Avec :

ca[i] = coût d'appariement des segments[i] correspondant à l'opération numéro i (substitution ou destruction) de la distance d'édition

l[i] = longueur moyenne de ces deux segments (substitution) ou bien d'un seul (destruction).

Le critère final est calculé par le maximum de la valeur absolue de (C₁[s] - C₂[s]) pour s variant entre le premier et le dernier segment. Soit c_max l'indice du segment s pour lequel le critère est maximum, on retient C₁ final = C₁[c_max] et C₂ final = C₂[c_max].

L'indice le plus fort de C₁ et C₂ correspond à la partie qui est similaire entre les deux chaînes et doit être supérieur à un seuil minimum pour que la coupure

soit validée. Cette partie doit être la plus longue possible par rapport à la chaîne entière.

L'indice le plus faible de C_1 et C_2 correspond à la partie qui est dissemblable entre les deux chaînes et doit être inférieur à un seuil maximum pour que la coupure soit validée.

Le respect de ces deux conditions assure la pertinence de la coupure qui est alors provisoirement acceptée. Dans le cas contraire, la coupure est ignorée et on passe à la recherche de similarité sur deux autres chaînes appariées.

La possibilité d'une coupure logique de la chaîne en trois parties au lieu de deux a été envisagée en premier lieu, mais cette voie a été abandonnée du fait de la difficulté de l'ajustement du critère de coupure. D'ailleurs, la préférence donnée à une coupure en deux fragments a été justifiée par la suite grâce à l'application récursive de la procédure en vérifiant le principe que chaque coupure pertinente en deux parties fait toujours baisser le coût d'appariement. Ainsi, la chaîne peut être fragmentée en trois parties en deux étapes successives.

La suite de la procédure consiste à examiner l'influence de la coupure sur l'appariement des graphes. Pour cela, on recherche, à partir de la distance d'édition, la position de la coupure sur chacune des chaînes originales, et on crée deux nouvelles chaînes, sur les

deux formes si nécessaire, tandis que les chaînes originales sont suspendues.

La procédure d'appariement de chaînes décrite dans le paragraphe 3 est ensuite appliquée sur ces nouvelles chaînes de la même façon.

La décision finale de coupure est prise en faisant le bilan du coût global d'appariement avant et après la coupure. Si le coût a diminué, la coupure est pertinente et est donc validée. On procède alors à la mise à jour des appariements et les chaînes originales sont abandonnées. Sinon, si le coût a augmenté, la coupure est rejetée et les nouvelles chaînes sont abandonnées.

Dans chacun des graphes des caractères '6', 'A' et 'N' de la figure 2, une diminution du coût d'appariement a été obtenue grâce à une coupure appropriée.

6. DÉTERMINATION DE LA DISTANCE ENTRE LES DEUX GRAPHES

6.1. Distance entre les arcs appariés

Au cours des précédentes étapes, on a fait correspondre chaque arc avec un et un seul arc de l'autre forme. Si le nombre d'arcs finalement obtenus après l'isolation des parties similaires, est différent dans

les deux formes, il reste alors des arcs non appariés. Chacun de ces arcs doit être éliminé, à moins que sa fusion avec un arc déjà apparié n'offre un coût moindre que son élimination. A l'issue de cette procédure, l'appariement de tous les arcs est complètement déterminé, et on peut alors définir le coût final d'appariement, c'est-à-dire la distance entre les arcs, comme étant la moyenne des coûts des arcs, pondérés par la longueur des arcs.

Le coût d'élimination d'un arc étant proportionnel à sa longueur, celle-ci intervient donc de façon quadratique pour les arcs éliminés.

Cette distance est bien significative dans de nombreux cas, mais dans d'autres cas, la distance entre les noeuds est plus discriminante.

6.2. Distance entre les noeuds appariés

Afin d'utiliser des points de référence comparables dans les deux formes, y compris dans le cas de fusion de noeuds, on procède d'abord à la construction des noeuds finaux. Ceux-ci sont obtenus par l'examen des arcs appariés réciproquement, en vérifiant si plusieurs arcs possèdent un noeud commun dans une forme, mais que leurs correspondants sont distincts dans l'autre forme. Dans ce cas, ces correspondants sont regroupés pour former un seul noeud final situé sur le graphe à la position de leur

barycentre. C'est précisément le cas pour le graphe test de la lettre 'K' de la figure 5a qui est comparé avec le graphe de référence de la figure 5b.

Les graphes possèdent alors le même nombre de noeuds finaux et la distance peut donc être calculée simplement par leur déplacement relatif aux arcs qui y sont connectés. Le cdg des arcs fournit un point de repère pour les noeuds qui a l'avantage d'être plus souple que le cdg global de la forme.

Dans certains cas le coût de chaque noeud ne suffit pas à rendre compte de la divergence des graphes. On fait donc intervenir de plus la différence de proportion des arcs connectés au noeud entre les deux formes.

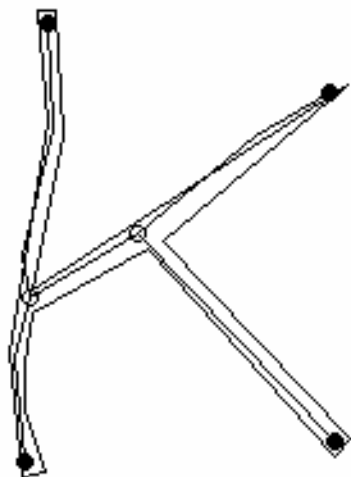
La distance finale entre les noeuds est la somme des coûts de chaque noeud pondérés par l'importance hiérarchique du noeud dans le graphe. Celle-ci est représentée par la proportion des arcs connectés à chaque noeud par rapport à l'ensemble des arcs appariés, non compris les arcs éliminés. En effet, ceux-ci ne sont pas comparables par définition. Il en résulte que leur contribution à la distance finale est empirique. On considère le déplacement du noeud à l'extrémité de l'arc éliminé lors de la rétraction de l'arc. Ce coût est additionné avec un coefficient de pondération au coût global des noeuds. Un bon réglage est obtenu lorsque le coût d'appariement de deux graphes représentant la

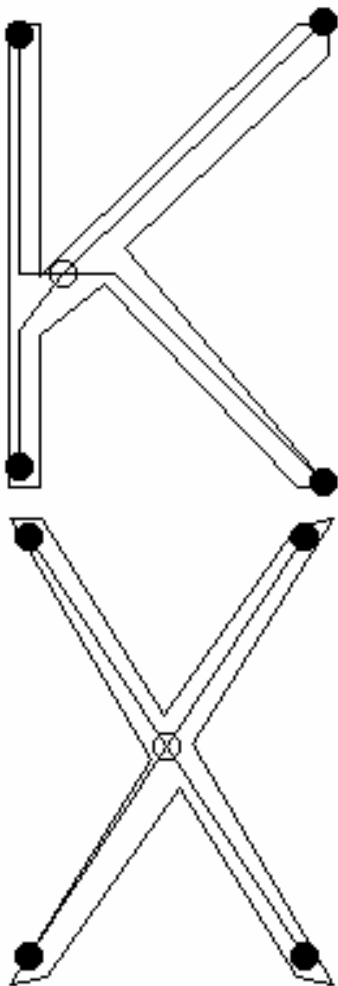
même lettre est faible tandis que le coût est fort pour deux lettres différentes.

6.3. Exemples de distances entre deux graphes

Les deux coûts que nous avons définis précédemment présentent une certaine complémentarité en fonction de la nature des graphes à apparier.

Le coût des chaînes est plus significatif pour les lettres de faible niveau structurel telles que "GNSZ25", tandis que le coût des noeuds est plus approprié pour distinguer les structures des lettres "KXH" sur la figure 5.





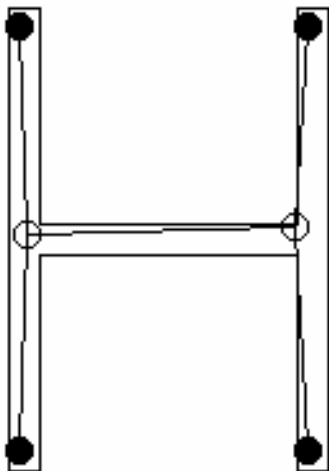


Figure 5

Les deux lignes de coûts suivantes montrent le résultat de l'appariement du graphe de la lettre manuscrite 'K' réalisé avec l'ensemble des lettres de l'alphabet (les trois autres graphes représentés sur la figure 5 sont les trois premiers caractères obtenus dans le classement des coûts) :

Coût relatif aux noeuds :

CN = K| 9.6 X|12.0 H|17.1 F|32.7 Y|36.2 A|48.4
R|49.1 E|50.5 T|55.4...

Coût relatif aux arcs :

CA = K|16.0 X|19.7 H|21.4 E|30.1 F|33.7 Y|38.7
A|45.5 T|70.6 B|100...

6.4. Exemples de similarités détectées

Les trois exemples suivants illustrent des cas typiques de variations observées sur l'écriture manuscrite. Dans l'immédiat, on ne se préoccupe pas de décider si les graphes de chaque couple sont censés représenter la même lettre ; on veut seulement mettre en évidence, grâce à l'appariement de leurs graphes, qu'ils sont proches d'aspect et contiennent des similarités incontestables.

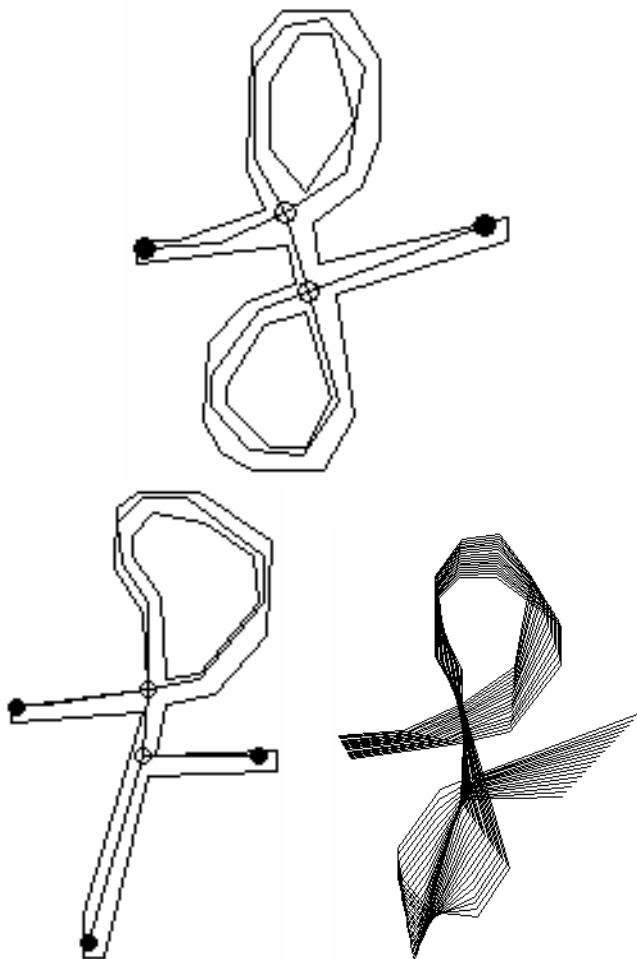
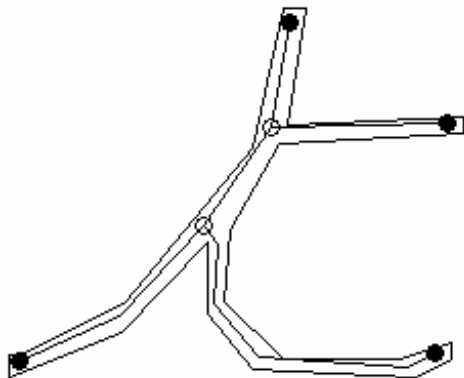
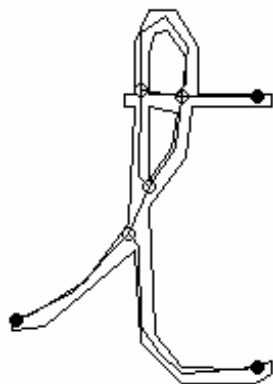


Figure 6a
35/47



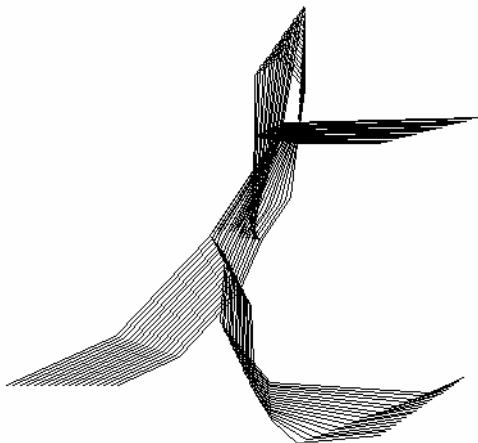


Figure 6b

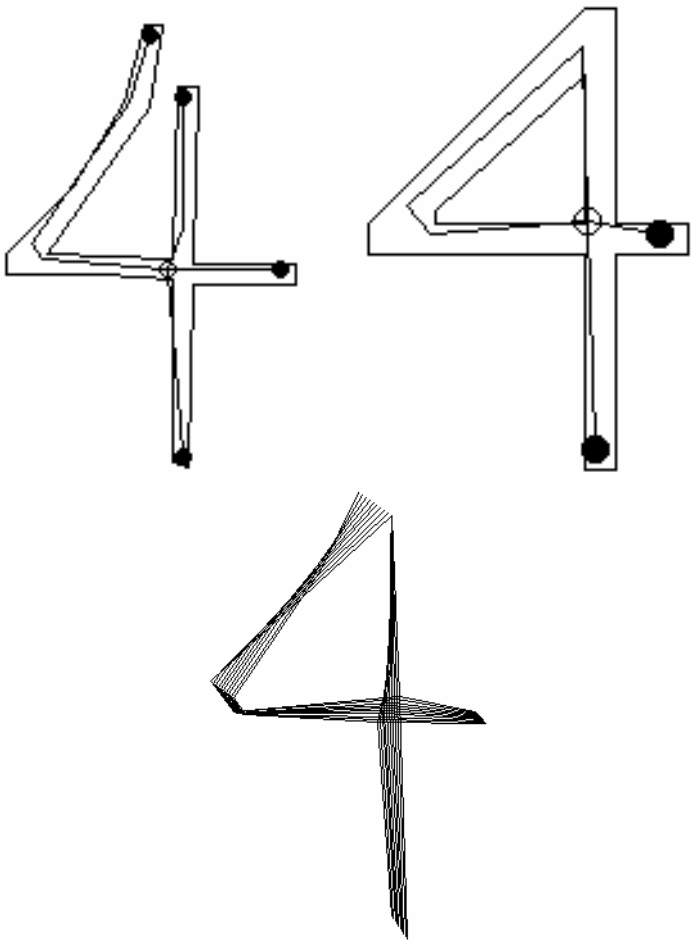


Figure 6c

Nous avons montré la transformation continue des deux formes en traçant les arcs reliant les noeuds lorsque ceux-ci sont progressivement interchangés avec leurs correspondants dans l'autre graphe. La fusion de la forme à reconnaître avec la forme apprise modélise en quelque sorte la fonction d'interprétation qui nous semble essentielle au processus de la lecture. La notion de "Warping" (transformation continue d'une forme en une autre) à surtout été considérée dans le cas de l'écriture en-ligne [KONDO 92].

Dans la figure 6a, la présence d'une boucle supérieure commune aux deux graphes a été mise en évidence. Les coûts obtenus sont $CN=22,4$ et $CA=20,3$.

La figure 6b illustre le cas de la fusion d'arcs qui se retrouvent dans les hampes dont le tracé est doublé. Les coûts obtenus sont $CN=33,0$ et $CA=20,6$.

La figure 6c montre le cas du chiffre '4' qui est remarquable par le fait que la structure des deux caractères présentés est bien distincte alors que leur aspect global est identique, comme l'indique le coût $CA=6,1$. La fusion des noeuds des deux branches supérieures a cependant permis un coût faible aussi pour les noeuds, soit $CN=4,9$.

L'objectif de l'appariement, dans ces trois exemples est d'obtenir une distance entre le graphe du caractère modifié et son caractère modèle qui soit inférieure à celles obtenues avec tous les autres

modèles. Etant donnée la multiplicité des contributions au calcul des coûts (translation, orientation, longueurs des segments...), il est délicat d'interpréter précisément le sens de leur valeur numérique. Cela exige un contrôle méthodique de l'ensemble des variables dans les tests effectués. Nous travaillons actuellement sur plusieurs alphabets manuscrits pour évaluer la discrimination obtenue pour des lettres minuscules. Dans l'immédiat, nos résultats en reconnaissance ne concernent que des lettres manuscrites majuscules.

7. EXEMPLE DE RECONNAISSANCE

La méthode a été testée sur un ensemble de caractères bâtons, ceux utilisés par l'INSA pour la saisie automatique des dossiers d'inscription des étudiants candidats à l'entrée à l'école d'ingénieurs [VINCENT 93]. Pour répondre à une nécessité de rigueur en reconnaissance de caractères manuscrits omniscriteur, une forte contrainte sur l'écriture a été imposée aux étudiants (figure 7). Les chiffres '1', '6' et '9' ont été dupliqués du fait du non respect de l'alphabet original par les scripteurs. Deux signes de ponctuation '-' et '/' ont été prévus. La taille des caractères après saisie est d'environ 50 pixels de haut et de large. Un dossier d'inscription numérisé en 200 dpi est présenté sur la figure 8.

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9 9 6 - / 1

Figure 7

DARGENTON, PATRICE
CHATILLON/THOUET

79200 PARTHENAY

49951899

M

080168

F

X

X

4

3

2

Figure 8

Les résultats de reconnaissance, effectuée sur une dizaine de dossiers de candidature, sont les suivants :

Sur les 466 caractères lus, 449 ont été reconnus, soit un taux de reconnaissance de 96,3% à comparer avec les 98% pour le système actuel (dont la moitié des erreurs sont dues aux candidats qui ne respectent pas les contraintes de l'alphabet).

Les principales erreurs de reconnaissance rencontrées montrent l'existence d'ambiguïtés sur l'alphabet utilisé, notamment entre les caractères 'A' et 'R' ainsi qu'entre 'l' et 'I'. L'auto-reconnaissance des lettres de l'alphabet permet d'évaluer l'ensemble des ambiguïtés afin de prévenir les risques d'erreur de reconnaissance, et si nécessaire d'améliorer l'alphabet.

Le taux de reconnaissance détaillé entre les deux distances CN et CA définies précédemment montre un résultat de 90,7% pour les noeuds et de 95,9% pour les arcs. Sur les 43 erreurs relevées pour les noeuds, 28 ont été corrigées grâce au coût des arcs ayant été préféré pour sa meilleure discrimination calculée par rapport au coût arrivant en deuxième position dans le classement. Réciproquement, 4 erreurs ont été corrigées pour les arcs.

La correction automatique de ces erreurs montre que la probabilité de reconnaissance peut être évaluée grâce à la discrimination offerte par les coûts : une

discrimination trop faible indique un résultat peu fiable, tandis qu'une forte discrimination sur chacun des deux coûts CN et CA indique une reconnaissance certaine. D'autres tests sont en cours pour exploiter un troisième coût indépendant basé sur l'appariement des contours des caractères, en les considérant comme des arcs ordinaires (boucles).

CONCLUSION

Malgré la complexité du problème, nous avons montré que l'appariement de deux graphes structurels quelconques constitue une approche intéressante dans la reconnaissance des caractères manuscrits. Il permet de déterminer la meilleure "ressemblance" ou correspondance entre deux formes, tandis que la reconnaissance consiste à sélectionner ensuite le meilleur appariement réalisé parmi un alphabet de référence.

Les résultats obtenus sur un ensemble réduit de lettres (caractères bâtons majuscules) offrent des perspectives intéressantes en ce qui concerne la reconnaissance des mots manuscrits. La mise en évidence des ambiguïtés naturelles des lettres manuscrites sous forme d'un classement de proximité

est une information qui peut être utilisée dans un contexte lexical.

Nous envisageons deux façons d'entreprendre la reconnaissance structurelle des mots manuscrits :

- la première, en construisant le graphe sur toutes les lettres (ou entités proches telles que n-gram ou tronçon de lettre) que l'on pourra extraire sur le mot ;

- la seconde, en recherchant dans le graphe complet du mot non segmenté des isomorphismes optimaux de sous-graphes de lettres apprises.

Notre objectif est de retenir un alphabet de référence contenant un nombre minimum nécessaire de lettres de référence (en y incluant quelques variantes) pour une application particulière.

Il nous reste à tester notre méthode sur une plus grande échelle pour observer comment évolue la discrimination entre les lettres de référence lorsque leur nombre augmente. Nous estimons que l'approche est validée dans la mesure où la distance entre les graphes est égale à la distance entre les lettres originales.

BIBLIOGRAPHIE

[d'ACIERNO 91] d'ACIERNO, A., De STEFANO, C. and VENTO, M. A Structural character recognition method using neural networks. ICDAR-91, First International Conference on Document Analysis and Recognition, 30 Sep.-2 Oct. 1991, Saint Malo, France, Vol. 2, p. 803-811

[BACKER 87] BACKER E., GERBRANDS J.J. Inexact graph matching used in machine vision. NATO ASI Series. Vol. F30, Pattern Recognition Theory and Applications, Edited by P.A. Devijver and J. Kittler, © Springer-Verlag Berlin Heidelberg 1987, p. 347-356

[BOUKARRI 91] BOUKARRI, B., JUVIN, D. et VIALIA, M. Reconstruction tridimensionnelle récursive de scènes au moyen d'une caméra mobile. - Application robotique en univers structure -. Actes du 8^e congrès AFCET-RFIA, Lyon-Villeurbanne, 27-29 Nov. 1991, Vol. 1, p. 417-428

[CHEBARO 91] CHEBARO, B., MASSIP-PAILHES, L. et CASTAN, S. Conception d'un système d'analyse du mouvement 3D de régions utilisant une séquence stéréoscopique. Actes du 8^e congrès AFCET-RFIA, Lyon-Villeurbanne, 27-29 Nov. 1991, Vol. 1, p. 45-54

[DARGENTON 93] DARGENTON P., VINCENT N., EMPTOZ H. Construction d'un graphe structurel représentatif d'une forme. ICOHD'93, Sixth International Conference on Handwriting and Drawing, Paris, July 4-5-6-7, 1993, p. 231-233

[KONDO 92] KONDO, S., MAITREE, K., ITOH, D. and ATSUTA, K. Structure analysis of handwriting using opposing relations. Proceedings of the 11th International Conference on Pattern Recognition, The Hague, The Netherlands, Sep. 1992, Vol. B, p. 529-532

[NISHIDA 92] NISHIDA, H., SUZUKI, T. and MORI, S. Thin line representation from contour representation of handprinted characters. From Pixel To Features III : Frontiers in Handwriting Recognition, S. Impevodo and J.C. Simon Editors, 1992, p. 29-40

[VINCENT 93] VINCENT N., EMPTOZ H. Automatic recognition of handwritten forms for registration in INSA. ICOHD'93, Sixth International Conference on Handwriting and Drawing, Paris, July 4-5-6-7, 1993, p. 83-85

[WONG 87] WONG A.K.C. Structural Pattern Recognition : A Random Graph Approach. NATO ASI Series. Vol. F30, Pattern Recognition Theory and

Applications, Edited by P.A. Devijver and J. Kittler,
© Springer-Verlag Berlin Heidelberg 1987, p. 323-345

[YOU 84] YOU M., WONG A.K.C. An Algorithm for
Graph Optimal Isomorphism. Proceedings of the 7th
International Conference on Pattern Recognition,
Canada, 1984, p. 316-319