UNIVERSITÉ DE MONTRÉAL

PROPOSITION D'UNE MÉTHODE D'INFÉRENCE DE LA SÉQUENCE DANS L'IMAGE D'UNE SIGNATURE MANUSCRITE

LOUIS BEAUMIER
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ES SCIENCES APPLIQUÉES (M.Sc.A.)
(GÉNIE ÉLECTRIQUE)
OCTOBRE 1996

© Louis Beaumier, 1996

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services
Acquisitions et services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-26453-X

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

PROPOSITION D'UNE MÉTHODE D'INFÉRENCE
DE LA SÉQUENCE DANS L'IMAGE D'UNE
SIGNATURE MANUSCRITE

présenté par: BEAUMIER Louis
en vue de l'obtention du diplôme de: Maîtrise es Sciences Appliquées
a été dûment accepté par le jury d'examen constitué de:

M. MOORE Marc, Ph.D.	président-rapporteur
M. PLAMONDON Réjean, Ph.D.	membre et directeur de recherche
M. SABOURIN Robert, Ph.D.	membre et codirecteur de recherche
M. BERTRAND Michel, Ph.D.	membre

À Caroline

REMERCIEMENTS

Je tiens premièrement à remercier mon directeur de recherche, M. Réjean Plamondon, pour m'avoir permis de réaliser ce projet, ainsi que pour la confiance témoignée tout au long de ce projet.

Je voudrais également remercier mon codirecteur, M. Robert Sabourin, pour son appui, son support et la détermination qu'il a manifestée pendant ces longues années.

J'aimerais également remercier tout spécialement certaines personnes impliquées plus ou moins volontairement dans ce projet: Tony Pilote et Vincent del Castillo mes collègues et amis de longue date, Julie Potvin pour avoir partagé mes états d'âme et de nombreux cafés, Jean-Philippe Valois et Champlain Landry pour avoir maintenu l'équilibre entre mon corps et mon esprit, Wacef Guerfali et Franck Leclerc pour leur accueil et leur bonne humeur.

Finalement je dois dire merci à tous les scripteurs qui se sont gentiment portés volontaires pour la constitution de la banque de données utilisée dans ce projet. Leur contribution fut essentielle pour la bonne conduite de ce projet.

RÉSUMÉ

Dans le domaine de la vérification de signatures manuscrites, les systèmes *en-ligne* sont actuellement plus performants que les systèmes *hors-ligne*. Une des raisons expliquant cette supériorité réside dans la connaissance des caractéristiques spatio-temporelles du signal traité. La connaissance du mouvement à l'origine d'une image de signature manuscrite permettrait l'application (partielle) des techniques d'analyse utilisées par les systèmes *en-ligne* pour des applications *hors-ligne*.

Le présent mémoire traite d'une technique d'inférence de la séquence dans l'image d'une signature manuscrite. Cette technique a été développée dans le but de servir d'étape de prétraitement dans un système de vérification de signatures manuscrites *hors-ligne*.

La solution proposée procède en comparant l'image de la signature (le signal de *test*) avec un des signaux de la banque de références (le signal de *référence*) et tente d'inférer la séquence dans l'image à partir de l'information fournie par le signal de *référence*.

Une des particularités de ce système est l'utilisation de signaux de nature différente. Les signaux de *référence* sont en effet des signaux de nature dynamique, c'est-à-dire

qu'ils sont constitués d'échantillons de position captés lors de la production de la signature.

Puisque les signaux ne sont pas de la même nature, ils doivent nécessairement faire l'objet d'une transformation particulière afin de les rendre comparables. L'espace de représentation choisi est celui du *graphe*. Pour chacun des signaux, un graphe doit donc être construit. Pour l'image, après l'avoir binarisée, des points caractéristiques, représentant soit des extremums locaux ou des points de discontinuité (*extrémités*, *croisements de traits*), sont identifiés sur les contours de la signature. Ces différents points sont ensuite regroupés en *zones caractéristiques*. Ce sont les zones ainsi formées qui constituent les noeuds du graphe associé à l'image. Un des critères guidant l'agrégation des points en zone est une mesure nommée *indice de visibilité* dont la valeur dépend de la distance euclidienne entre les points et du positionnement des points sur les contours.

La formation du graphe à partir du signal de référence se fait sensiblement de la même façon, mais plus simplement. Puisqu'il s'agit d'une séquence de point, chaque noeud est formé à partir d'un point unique identifié sur la séquence de points. Les points à identifier sont les *extremums locaux* et les *extrémités* de composantes.

Les différents seuils de décision nécessaires au traitement de l'information sont, lorsque possible, déterminés en fonction d'un paramètre dépendant de l'image: la *largeur moyenne du tracé*. Cette valeur, estimée à partir du contenu de l'image, rend la technique proposée indépendante de la résolution des images et de la largeur de la pointe de l'outil d'écriture.

Après la formation des deux graphes, une procédure de mise en correspondance permet d'identifier, dans le graphe associé à l'image, les noeuds correspondants dans le graphe associé à la séquence de référence. Il s'agit ici d'un problème d'étiquetage consistant qui trouve sa solution en utilisant un algorithme de mise en correspondance par *relaxation*. Cette technique itérative, ici adaptée pour tenir compte de la spécificité du problème, permet de faire un choix parmi un ensemble de candidats en bonifiant des poids de mise en correspondance sur la base des poids accordés aux différents voisins du noeud traité.

La mise en correspondance a comme résultat de faire correspondre, si possible, chaque noeud du graphe associé à l'image avec un noeud du graphe associé à la séquence de référence. Puisque ce dernier est, de par sa nature même, orienté (ou séquencé), il est possible de reproduire la séquence correspondante dans le graphe associé à l'image, sur la base des résultats de la mise en correspondance.

Il ne reste plus qu'à générer une liste de points (un squelette) en suivant la séquence ainsi établie. Cette connaissance sur le cheminement du tracé permet, entre autres, de procéder à une squelettisation des zones de discontinuité de façon informée. Il n'est donc pas nécessaire de faire une interprétation locale de la discontinuité concernée puisque le comportement du tracé est dicté par une information de plus haut niveau.

Les performances du système proposé ont été évaluées à l'aide d'une banque de 400 signatures. Trois expériences différentes permettent d'illustrer la capacité d'inférence de la séquence ainsi que la possibilité d'utiliser ce système comme premier étage de détection des faux aléatoires.

Les résultats en inférence de la séquence montrent que, dans le cas d'une comparaison avec une signature authentique, les taux de reconstruction varient entre 55 % et 88 %. Pour ce qui est de la capacité de détection des faux aléatoires, l'expérience fut menée en tentant de minimiser l'erreur de type 1 (le rejet d'une signature authentique). Pour cette expérience, l'erreur de type 1 fut nulle pour 19 des 20 scripteurs (ce qui représente 1.25 % d'erreur en moyenne) et l'erreur de type 2 (l'acceptation d'un faux) fut de 18.74 %. Ce qui veut dire l'élimination de plus de 80 % des faux aléatoires.

La technique proposée semble donc prometteuse et le travail accompli constitue une base intéressante pour l'élaboration future d'une nouvelle technique *hors-ligne* de vérification de signatures manuscrites.

ABSTRACT

In the field of signature verification, *on-line* systems perform better than *off-line* ones. The main reason for this superiority seems to be the use of temporal information. Recovering the temporal information from the image of a signature would allow the (partial) application of *on-line* system techniques to *off-line* systems.

The present work is about a sequence inference technique in signature images. The purpose of this was to create a pre-treatment stage for a new *off-line* signature verification system.

The proposed solution consists of comparing the signature image (the *test* signal) with a signal from the reference database (the *reference* signal), and tries to identify a sequence in the image based on the information provided by the *reference* signal.

One particularity of this approach is the use of two different types of signals. The reference signals are of a dynamic nature, which means that they are composed of positions sampled during the production of the signature.

Because the signals are of a different nature, they must be transformed in a way that allows comparison. The chosen representation space is the graph. For each type of signal, a specific graph must be constructed. After a binarization stage, specific pixels, like *extremum* or *break point*, are identified along the contour of the signature. To form the node of the graph, pixels are grouped together on the basis of a *visibility index*. The value of this index depends on the Euclidean distance between pixels and on their contour position.

The graph representing the dynamic signal is obtained in a similar way; because this signal is formed from a sequence of pixels, the nodes of the graph are formed from only one pixel. The pixels can be local extremum or end point of a component.

Whenever it is possible, each threshold used during the process is fixed as a multiple of an image dependent parameter: the *signature mean width*. This value, which is estimated from the binarised image, allows the technique to be self-adaptive, regardless of the resolution of the images, and the width of the writing tool used.

After the formation of the two graphs, a matching process identifies the corresponding nodes between the two graphs. This problem can be viewed as a consistent labeling process, and is solved using the relaxation matching technique. This iterative technique, after being adapted for this special application, is designed

to make a choice among a number of candidates through an initial weight assignment. During the updating process, each neighboring node is taken into consideration when determining the new weight.

The optimal outcome of the matching process is to associate each node of the image graph with a node of the reference sequence graph. Since this graph is, by its nature, oriented (or sequenced), it is easy to reproduce this sequence in the image graph on the basis of the matching results.

The final task to perform is the generation of a list of pixels in accordance with the identified sequence in the image. The knowledge concerning the sequence simplifies the skeletonization process, especially in the stroke-crossing regions. In the stroke-crossing regions, the skeleton can be generated on a basis of knowledge of higher level than the one provided by local analysis of the region.

System performances have been evaluated using a database of 400 signatures. Three different experiments illustrate the sequence inference capacity of the system as well as the possibility of using the system as the first screening stage for random forgery detection.

The results for sequence inference show that for a comparison with a genuine signature, the reconstruction rate varies between 55 % and 88 %. For the false signature detection capacity experiment was conducted in order to minimize the type 1 error (the rejection of a genuine signature). In this experiment, type 1 error was null for 19 out of the 20 writers (which represents a mean error of 1.25 %), and type 2 error (the acceptance of a random forgery) had a man error of 18.74 %. The result is an elimination of more than 80 % of the random forgeries.

The proposed technique is promising, and combined with the work accomplished, provides an interesting framework for future elaboration on an *off-line* signature verification technique.

TABLE DES MATIÈRES

DÉDICACE iv

REMERCIEMENTS v

RÉSUMÉ vi

ABSTRACT xi

TABLE DES MATIÈRES xv

LISTE DES TABLEAUX xxi

LISTE DES FIGURES xxiii

LISTE DES SYMBOLES xxxvi

LISTE DES ANNEXES xxxix

1. INTRODUCTION 1

1.1 La vérification de signatures 1

1.1.1 L'utilité de ces systèmes 1

1.1.2 Les étapes essentielles 2

1.1.3 Les solutions proposées: deux approches 5

1.1.4 Les caractéristiques intéressantes 6

1.2 L'inférence de la séquence 7

1.2.1 L'utilité de la séquence 7

1.2.2 Les techniques existantes 9

1.3 Proposition de recherche 18

1.3.1 Les objectifs visés 18

1.3.2 Description de la méthode proposée 20

1.3.3 Les caractéristiques spécifiques à la méthode proposée 25

1.3.4 Plan du mémoire 26

2. TRAITEMENT DE L'INFORMATION STATIQUE 28

2.1 Description générale 28

2.2 Prétraitement de l'image 33

2.2.1 Nature des images traitées 33

2.2.2 Binarisation de l'image 34

2.2.2.1 Détermination du seuil 34

2.2.2.2 Élimination du bruit 38

2.2.2.3 Comblement des petits trous 39

2.2.3 Extraction des contours du tracé 40

2.3 Formation des zones extremums 43

2.3.1 Localisation et identification des points extremums 43

2.3.2 Validation des points extremums 47

2.3.2.1 Identification des régions bruitées 47

2.3.2.2 Filtrage des régions bruitées 49

2.3.3 Formation des zones extremums 52

2.4 Formation des zones de discontinuité 55

2.4.1 Localisation et identification des points de discontinuité 55

2.4.1.1 Nature des points identifiés 55

2.4.1.2 Estimation de la courbure des contours 56

2.4.1.3 Filtrage des profils de courbure 59

2.4.2 Validation des régions de discontinuité 63

2.4.3 Formation des zones de discontinuité 66

2.4.3.1 Critères de formation 66

2.4.3.2 Méthode de sélection des points 67

2.4.3.3 Identification des zones de discontinuité et des boucles dégénérées ... 70

2.5 Formation du graphe de relations 72

2.5.1 Composition du graphe de relations 72

2.5.2 Identification des arcs 73

2.5.3 Édition du graphe de relations 75

3. TRAITEMENT DE L'INFORMATION DYNAMIQUE 80

3.1 Description générale 80

3.2 Interpolation de la trajectoire 84

3.2.1 Nature des données 84

3.2.2 Filtrage des données 87

3.2.3 Choix de la méthode d'interpolation 89

3.3 Formation des zones caractéristiques 90

3.3.1 Localisation et identification des points caractéristiques 91

3.3.2 Validation des points caractéristiques 92

3.3.3 Formation des zones caractéristiques 93

3.4 Graphe de représentation 94

4. MISE EN CORRESPONDANCE DES GRAPHES 96

4.1 Description générale 96

4.2 Développement du graphe de relations 100

4.2.1 Caractéristiques du développement 100

4.2.2 Développement des croisements 101

4.2.3 Développement des boucles dégénérées 104

4.2.4 Développement des extrémités 109

4.3 Mise en correspondance par relaxation 111

4.3.1 Description de la procédure de mise en correspondance 111

4.3.2 Attribution des poids initiaux 113

4.3.3 Mise à jour des poids par relaxation 119

4.3.4 Validation des mises en correspondance 122

4.3.5 Formation du graphe de mise en correspondance 124

4.4 Mise en correspondance par croissance consistante 124

4.4.1 Identification des sous-séquences 124

4.4.2 Critères d'ajout 126

4.4.3 Procédure de mise en correspondance 127

5. SQUELETTISATION DE L'IMAGE 132

5.1 Définition du squelette 132

5.2 Squelettisation de signaux manuscrits 133

5.2.1 Composition des signaux manuscrits 133

5.2.2 Squelettisation adaptée à l'écrit 134

5.3 Squelettisation de la signature manuscrite 135

5.3.1 Définition de la méthode 135

5.3.2 Squelettisation du tracé isolé 137

5.3.3 Squelettisation des intersections de traits 145

6. ÉVALUATION DES PERFORMANCES 152

6.1 Composition et acquisition de la banque de données 152

6.2 Métriques définies pour l'évaluation des performances 154

6.3 Étapes d'évaluation des performances 159

6.4 Évaluation de l'efficacité 161

6.4.1 Protocole expérimental 161

6.4.2 Critères de validation 163

6.5 Évaluation de la capacité de détection des faux aléatoires 164

6.5.1 Protocole expérimental 164

6.5.2 Critères de validation 169

6.6 Analyse des résultats 179

7. CONCLUSION 200

7.1 Rappel des objectifs et travail accompli 200

7.2 Contribution originale 202

7.3 Limites et contraintes 205

7.4 Nouvelles voies de recherche 207

BIBLIOGRAPHIE 210

ANNEXES 219

LISTE DES TABLEAUX

Tableau 2.1 Algorithme utilisé pour combler les petits trous 40

Tableau 4.1 Algorithme synthèse de la procédure de mise en correspondance par relaxation 112

Tableau 4.2 Algorithme synthèse de la procédure de mise en correspondance par croissance consistante..... 130

Tableau 5.1 Algorithme utilisé pour la squelettisation par suivi de contours. 143

Tableau 6.1 Protocole d'évaluation de l'efficacité: comparaison de signaux originaux. 162

Tableau 6.2 Protocole d'évaluation de l'efficacité: comparaison de signaux authentiques. 162

Tableau 6.3 Tableau synthèse résumant la composition des différents ensembles de signatures utilisés pour l'évaluation des performances en identification des faux aléatoires. 166

Tableau 6.4 Algorithme pour l'évaluation des performances du système pour la discrimination des faux aléatoires: phase d'*apprentissage*. 167

Tableau 6.5 Algorithme pour l'évaluation des performances du système pour la discrimination des faux aléatoires: phase de *généralisation*. 168

Tableau 6.6 Synthèse des différentes expériences menées pour l'évaluation des performances du système proposé..... 179

Tableau 6.7 Résultats de l'expérience en classification. Les taux d'erreur sont exprimés en pourcentage. 194

Tableau F.1 Procédure d'interpolation récursive par la méthode des splines cubiques. 246

LISTE DES FIGURES

Figure 1.1 *DFD* illustrant les étapes de traitement nécessaires à la vérification de signatures. 3

Figure 1.2 Illustration des erreurs de squelettisation devant être corrigées avant l'inférence de la séquence. (a) Correction d'une boucle ou d'une hampe. (b) Correction d'un croisement de traits. 11

Figure 1.3 Synthèse du traitement proposé dans [BOCC93]. La méthode est ici appliquée à l'allographe *A*. (a) Transformation en axe médian (*MAT*). (b) Approximation polygonale (*PMAT*). (c) Correction des distorsions (*CPMAT*). 14

Figure 1.4 *DFD* synthèse de la technique proposée pour l'inférence de la séquence. 23

Figure 2.1 *DFD* schématisant le traitement de l'information statique. 29

Figure 2.2 (a) Image de luminance I_e ; (b) le graphe de relations G_{rel} extrait de cette image. 30

Figure 2.3 Résultat du prétraitement de l'image de luminance Ie: l'image binaire I_{bin} et les listes de points de contours P_{cont}. 31

Figure 2.4 Image binaire I_{bin} dans laquelle sont identifiées les zones d'intérêt Z_{rel}, représentées par des cercles. 32

Figure 2.5 Histogramme de luminance associé à l'image en niveaux de gris présentée à la Figure 2.2 (a). 35

Figure 2.6 Histogramme bidimensionnel utilisé pour l'estimation du seuil de binarisation par une mesure d'entropie 2D. 37

Figure 2.7 Exemple de contours extraits. Les contours externes sont extraits dans le sens anti-horaire et les contours internes, dans le sens horaire. 42

Figure 2.8 Profils des coordonnées (x,y) associées au contour externe du tracé représenté. 45

Figure 2.9 Image illustrant les 8 types d'extremum possibles. Un extremum *relatif* est identifié par le symbole *prime*. 46

Figure 2.10 Illustration de la variation minimum considérée comme significative. 48

Figure 2.11 Les *régions bruitées* identifiées (A et B) pouvant être interprétées comme un extremum, un seul point par région est retenu. 50

Figure 2.12 Exemples de régions bruitées ne pouvant être interprétées comme un extremum. Pour la région A deux points sont retenus. Pour toutes les autres, aucun point n'est retenu. 52

Figure 2.13 Illustration du résultat de la formation des Z_{ext} pour une image donnée. 54

Figure 2.14 Exemple de profil de variation d'orientation calculé avec la méthode de O'Gorman. Puisque le pas d'échantillonnage est constant, ces valeurs reflètent directement la courbure.....58

Figure 2.15 Illustration de différentes opérations morphologiques appliquées à une fonction f en utilisant l'*élément structurant* B : a) érosion $f \ominus B$, b) dilatation $f \oplus B$, c) ouverture f_B et chapeau-haut-de-forme $f - f_B$, d) fermeture f^B et chapeau-bas-de-forme $f - f^B$. 61

Figure 2.16 a) Exemple de *points de discontinuité* identifiés et des *régions de discontinuité* correspondantes. b) Profil de courbure correspondant au contour extérieur de l'objet présent en (a). Les portions de profil identifiées correspondent à des *régions de discontinuité* 65

Figure 2.17 Illustration du concept de clique maximale. Dans cet ensemble de 5 points, 2 cliques maximales sont identifiées. 68

Figure 2.18 Les zones identifiées par des flèches représentent des boucles dégénérées 71

Figure 2.19 Exemple montrant les différents types d'arc. Tous les arcs qui ne sont pas *complets* (traits pleins), peuvent être soit *incomplets* (pointillés) ou *multiples* (tirets). 74

Figure 2.20 Exemple de G_{rel} montrant certaines erreurs d'interprétation (b) et l'image binaire correspondante incluant les différents points caractéristiques (a). 75

Figure 2.21 Représentation d'une portion de tracé devant faire l'objet d'une édition. La portion présentée est modifiée par l'application des deux règles d'édition. 76

Figure 2.22 Résultat de l'application des règles d'édition sur la portion de tracé présenté à la Figure 2.21. Dans cet exemple, il y a eu fusionnement de zones et ajout d'un point. 78

Figure 2.23 G_{rel} de la Figure 2.20 avant (a) et après (b) édition. 79

Figure 3.1 *DFD* montrant les différentes étapes de traitement de l'information dynamique. 81

Figure 3.2 Exemple de points échantillonnés. 82

Figure 3.3 Résultat de l'interpolation entre les échantillons présentés à la figure précédente. 82

Figure 3.4 Points caractéristiques P_{car} identifiés sur la trajectoire interpolée. Il s'agit de points *extremums* et d'*extrémités de trait*. 83

Figure 3.5 Graphe de représentation G_{rep} formé à partir des points caractéristiques P_{car} identifiés. 84

Figure 3.6 Illustration de la procédure de changement de référentiel pour les données des signaux dynamiques. 85

Figure 3.7 Échantillons captés par la tablette à digitaliser lors de la production de l'allographe *t*. 86

Figure 3.8 Échantillons avant (a) et après (b) filtrage. 88

Figure 4.1 *DFD* résumant les étapes de traitement pour la mise en correspondance. 97

Figure 4.2 Exemples théoriques illustrant l'opération de développement de *croisements*. 103

Figure 4.3 Exemples théoriques illustrant l'opération de développement de *boucles dégénérées*. 105

Figure 4.4 Exemples théoriques illustrant le positionnement des points sur les contours de *boucles dégénérées*..... 107

Figure 4.5 Exemples théoriques illustrant le positionnement des centres des zones résultant du développement de *boucles dégénérées*..... 108

Figure 4.6 Exemple de *boucle dégénérée* identifiée comme une *extrémité* E voisine d'un *croisement* C. Détails de l'image de luminance (a), de l'image binarisée (b) et du graphe de relations (c)..... 109

Figure 4.7 Exemple théorique illustrant l'opération de développement d'une *extrémité*. Pour qu'une telle zone soit développée, il faut obligatoirement qu'elle soit voisine d'un *croisement*. 110

Figure 4.8 Exemple d'un *graphe de relations développé* G_{rel}'. Ce graphe est le résultat du développement du graphe présenté à la Figure 2.2 b. 111

Figure 4.9 Représentation graphique des différentes valeurs tirées de l'image, utilisées pour l'estimation et la normalisation des positions spatiales. Les

paramètres correspondants sont extraits de la séquence de la même façon. 115

Figure 4.10 Illustration de la signification des angles θ_r et θ_d utilisés pour l'évaluation du facteur angulaire $da_i(\lambda)$. L'angle θ est associé au noeud Z qui a comme voisins les noeuds V1 et V2. La méthode de mesure est la même dans les deux graphes. 117

Figure 5.1 *DFD* montrant les étapes associées à la procédure de squelettisation de la signature manuscrite. 137

Figure 5.2 Illustration de la procédure de déplacement des pointeurs le long des contours. (a) La position courante est représentée par les pointeurs PG_1 et PD_1. (b) La distance minimum est associée au couple PG_1 et PD_2, qui deviennent la nouvelle position courante, PG_1' et PD_1'. 140

Figure 5.3 Exemples de squelettisation du *tracé isolé* par suivi de contours. 144

Figure 5.4 Illustration de la procédure d'interpolation pour les intersections de traits. (a) Positionnement des points de contrôle. (b) Interpolation d'un polynôme de degré 3 dans l'intervalle. 147

Figure 5.5 Exemples de squelettisation des *intersections de traits* par interpolation. 150

Figure 6.1 Représentation schématique de la banque de données. 153

Figure 6.2 Illustration du concept de *séparabilité* des classes. 170

Figure 6.3 Graphiques illustrant, pour chacun des 20 scripteurs, la valeur de d en fonction des facteurs α et β. Les zones foncées, représentant des valeurs de d positives, indiquent les valeurs de α et β pour lesquelles les deux classes sont complètement *séparées*. 172

Figure 6.4 Graphiques illustrant, pour chacun des scripteurs, les taux d'erreur ε_1 et ε_2 en fonction de la position du seuil. Ces résultats de la phase d'apprentissage permettent d'identifier des seuils de classification personnalisés pour chaque scripteur. 175

Figure 6.5 Positionnement des différents seuils de classification lorsqu'il y a chevauchement des classes (a) et lorsqu'elles sont séparées (b). Positionnement proposé pour le seuil de classification dans les deux même situations (c et d)..... 178

Figure 6.6 Graphe montrant la *proportion de séquence non considérée* ΔL pour les expériences 1, 2 et 3a. 181

Figure 6.7 Graphique des valeurs de Δl_i et Δp_i pour une comparaison donnée. Chaque point de coordonnée $(\Delta l_i, \Delta p_i)$ corrrespond à un appariement i entre deux segments. 182

Figure 6.8 (a) Séquence de référence; (b) séquence inférée. Les lettres identifiant les couples de segments réfèrent aux points correspondants dans le graphe de la Figure 6.7. 183

Figure 6.9 Exemples de résultats de traitement tirés de l'expérience 2. 185

Figure 6.10 Trois représentations d'une signature *authentique* rejetée lors de l'expérience en classification: (a) Image de luminance I_e, (b) séquence

inférée S_{inf} superposée à l'image binarisée I_{bin} , (c) graphe de relations G_{rel} 195

Figure 6.11 Quatre représentations tirées de la même information originale illustrant le problème de la différence d'interprétation: (a) image binarisé, (b) graphe de relations, (c) séquence de référence et (d) graphe de représentation..... 197

Figure A.1 Illustration de la mesure de distance *d* entre les points pour l'estimation de *LMT*. La distance est évaluée selon l'axe correspondant au type d'extremums formant le couple. 220

Figure B.1 Portion de contour et positionnement des paramètres nécessaires au calcul de la variation angulaire..... 222

Figure B.2 Illustration d'une portion de contours problématique. La différence de trajectoire entre les droites P_1P_2 et S_1S_2 donne un angle de courbure θ_c pour le point *p* plus grand que 180°..... 223

Figure B.3 Extrémité théorique de composante..... 225

Figure C.1 Région de discontinuité dans un croisement idéal. Le trait foncé représente la région de discontinuité identifiée après le filtrage morphologique. Dans un tel cas, θ_2 et θ_3 sont égaux et le maximum de courbure est atteint au point P_1. 227

Figure D.1 (a) Exemple des vecteurs utilisés comme analogie à la notion de « direction du regard ». (b) Technique utilisée pour déterminer l'orientation du vecteur perpendiculaire..... 231

Figure D.2 Illustration des angles formés par les perpendiculaires et la droite reliant les points p_i et p_j. Ces angles servent à l'évaluation des fonctions fa_i et fa_j. 233

Figure D.3 Comportement de l'indice de visibilité en fonction de la position relative des points. (Tous ces graphiques sont exprimés en fonction de d_{max})..... 235

Figure E.1 Représentation idéale d'un croisement de traits avec un angle $\alpha < 90°$..... 237

Figure E.2 Illustration de la variation entre la position du contour avant et après la procédure de *comblement des petits trous*. Le contour théorique est montré en (a). Les pixels ombrés en (b) sont ceux ajoutés au tracé par la

procédure de *comblement des petits trous*. Le nouveau contour est représenté par les pixels tracés en noir. 239

Figure F.1 Décomposition du signal dynamique en deux fonctions $x=f(n)$ et $y=f(n)$, afin de rendre possible l'interpolation par splines cubiques. 245

LISTE DES SYMBOLES

D_e	liste des points échantillonnés
D_e'	liste filtrée des points échantillonnés
D_{int}	liste des points interpolés (incluant D_e')
$d_{cont}(p_i,p_j)$	distance sur le contour entre deux points p_i et p_j
d_{Xd} , d_{Yd}	taille en x et y du rectangle circonscrit des points de la séquence interpolée
d_{Xs} , d_{Ys}	taille en x et y du rectangle circonscrit des pixels de l'image binaire
$E(m,n)$	étiquette indiquant si le pixel (m,n) appartient au fond (0) ou au tracé (1)
G_{MEC}	graphe de mise en correspondance
G_{rel}	graphe de relations
G_{rel}'	graphe de relations développé (après interprétation des Z_{disc})
G_{rep}	graphe de représentation
$g(m,n)$	un pixel de luminance localisé à la position (m,n), avec $0 \leq g(m,n) \leq 255$
I_{bin}	$= \{E(m,n)\}^{Nm,Nn}$, une image binaire
I_e	$= \{g(m,n)\}^{Nm,Nn}$, une image de luminance
N_m	nombre max. de pixels sur l'abscisse
N_n	nombre max. de pixels sur l'ordonnée

P_{car}	liste des points caractéristiques
P_{cont}	liste de pixels formant les contours des objets
P_{disc}	liste de pixels identifiés comme discontinuités
P_{ext}	liste de pixels identifiés comme extremums
p_i	un point sur le contour d'un objet
S_{disc}	liste des points de squelettisation des discontinuités
S_{inf}	liste des points du squelette interpolé ($S_{int} + S_{disc}$)
S_{int}	liste des points de squelettisation du tracé isolé
Z_{car}	liste des zones caractéristiques
Z_{disc}	liste des zones de discontinuité
Z_{ext}	liste des zones extremums
Z_{rel}	$= Z_{disc} \cup Z_{ext}$
ζ_{MEC}	indice de similarité
$\theta(p_i)$	valeur de la courbure au point p_i
μ_{Xd} , μ_{Yd}	moyenne en x et y de la distribution des points de la séquence interpolée
μ_{Xs} , μ_{Ys}	moyenne en x et y de la distribution des pixels de l'image binaire
$v(p_i,p_j)$	indice de visibilité entre les points p_i et p_j
σ_{Xd} , σ_{Yd}	variance en x et y de la distribution des points de la séquence interpolée

σ_{X_S} , σ_{Y_S} variance en x et y de la distribution des pixels de l'image binaire

τ_{cont} seuil distance inter-extremum sur le contour

τ_{int} seuil de distance pour filtrage de $\mathbf{P}_{car}$

τ_{lum} seuil de binarisation

τ_{pix} seuil bruit

τ_{MEC} seuil de détection des faux aléatoires

$\tau_{\Delta\theta}$ seuil différence d'amplitude de courbure pour validation des discontinuités

τ_{θ} seuil de courbure pour validation des extrémités

$\psi(g(m,n))$ entropie au point $g(m,n)$

LISTE DES ANNEXES

ANNEXE A Méthode d'estimation de la largeur moyenne du tracé LMT 219

ANNEXE B Choix des paramètres M et W pour le calcul de la courbure 222

ANNEXE C Choix du seuil amplitude $\tau_{\Delta\theta}$ pour la validation des points de discontinuités 227

ANNEXE D Méthode d'évaluation de l'indice de visibilité $v(p_i,p_j)$ 230

ANNEXE E Choix du paramètre d_{max} pour l'évaluation de l'indice de visibilité $v(p_i,p_j)$ 236

ANNEXE F Implantation de la méthode d'interpolation par splines cubiques 240

ANNEXE G Composition de la banque de données 247

1. INTRODUCTION

1.1 La vérification de signatures

1.1.1 L'utilité de ces systèmes

Le rôle d'un système de vérification de signatures est de statuer sur l'authenticité d'une signature manuscrite. Une telle procédure a comme but l'utilisation de la signature manuscrite comme *clé* d'identification des individus. La vérification procède donc en comparant une signature manuscrite dite de *test* avec une signature dite de *référence* dont l'origine est connue. Si la signature de *test* est identifiée comme étant authentique, c'est-à-dire comme ayant été produite par la même personne que la signature de *référence*, alors l'identité de l'individu ayant produit la signature de *test* est supposée celle de celui ayant produit la signature de *référence*.

Afin de rendre un tel verdict il est donc primordial de posséder une information a priori sur les personnes à identifier: cela constitue la *banque de référence* du système. Les signaux composant cette banque doivent être reconnus comme authentiques puisqu'ils serviront de référence dans les comparaisons visant à identifier les individus. Un *numéro d'identification personnel* (*NIP*) permet d'associer chaque signature de *référence* avec un individu donné. Lors d'une vérification, la personne dont l'identité doit être validée doit produire un spécimen de sa signature et fournir son *NIP*. Le *NIP* peut aussi être un numéro de compte bancaire lors de la vérification de la signature sur un chèque. Peu

importe sa provenance, le *NIP* permet de choisir la signature de *référence* avec laquelle comparer la signature de *test* que vient de produire le scripteur.

1.1.2 Les étapes essentielles

Vu de l'extérieur, un système de vérification de signatures peut être perçu comme une *boîte noire* dans laquelle entre une signature et un *NIP* et de laquelle sort une décision sur l'authenticité de la signature. Vue de l'intérieure, cette *boîte noire* se divise en un certain nombre d'étapes communes à toutes les techniques.

Le *diagramme de flux de données* (*DFD*) de la Figure 1.1 résume ces étapes essentielles. Il faut dans un premier temps faire l'*acquisition* de la signature. Cette étape, constituant l'interface avec l'extérieur, permet de créer une première *représentation machine* de la signature. Le choix de cette représentation dépend directement des étapes de traitement successives. Toute l'information nécessaire à la suite du traitement doit y être représentée.

La signature-objet ainsi créée est ensuite soumise au *prétraitement*. Cette étape peut se diviser en plusieures opérations, allant du filtrage du signal jusqu'à la segmentation de celui-ci. Le nombre et la complexité des traitements sont propres à chacun des systèmes de vérification de signatures. Le résultat de cette étape doit servir à la comparaison avec le signal de *référence*.

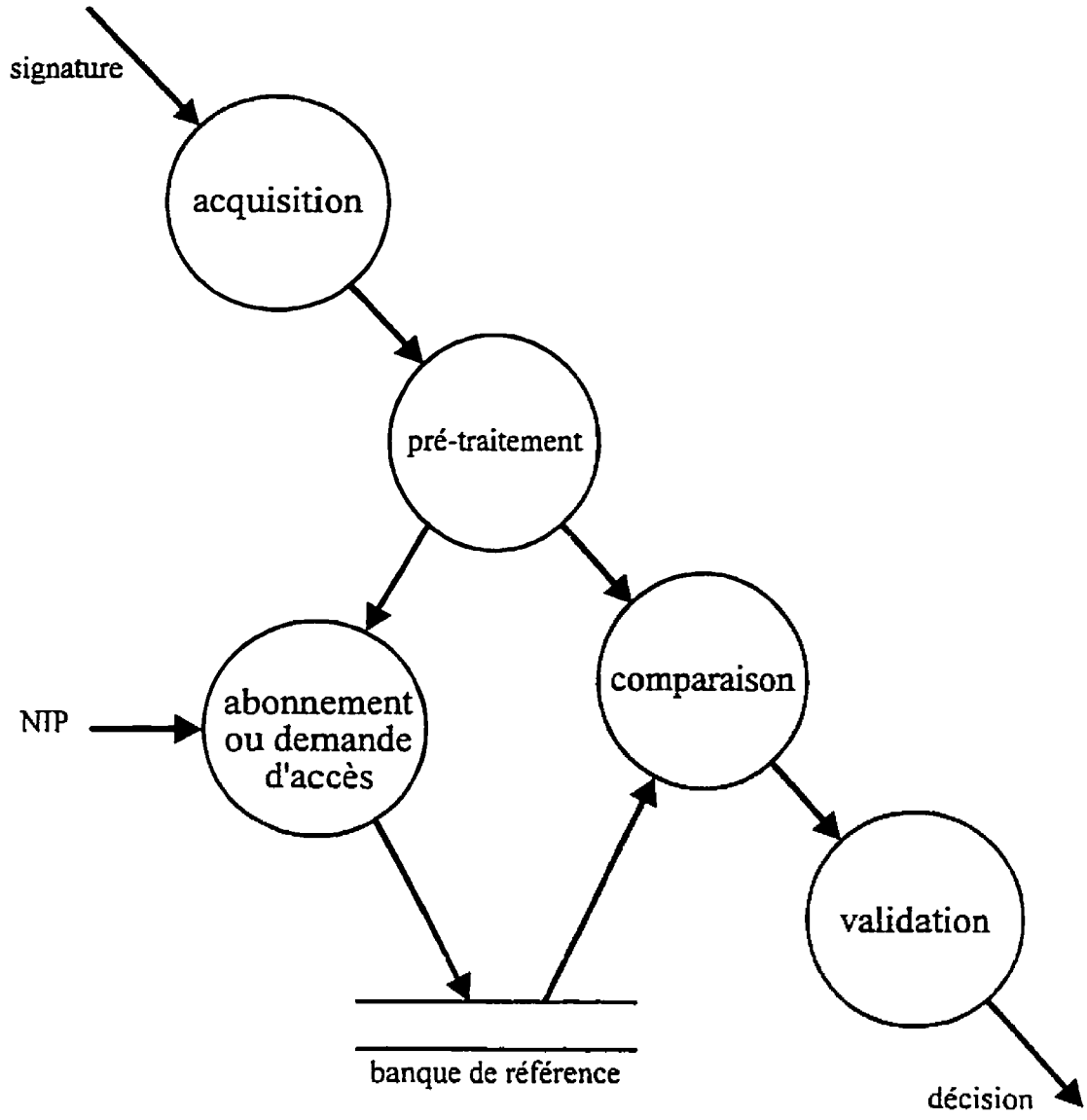


Figure 1.1 *DFD* illustrant les étapes de traitement nécessaires à la vérification de signatures.

Les deux premières étapes décrites jusqu'ici sont également impliquées dans la procédure d'*abonnement* au système. L'*abonnement* consiste à ajouter à la banque de données les informations relatives à un nouveau scripteur. L'information servant de *référence* doit donc être acquise et prétraitée avant d'être ajoutée à la banque de donnée.

L'information contenue dans la banque de références est utilisée à la prochaine étape de traitement: la *comparaison*. C'est à cette étape que sont comparées la signature de *test* et la signature de *référence*. Les représentations obtenues après le prétraitement sont comparées et différentes mesures sont prises de ces comparaisons. Encore une fois, la nature et le nombre de ces mesures varient d'un système à l'autre. Une constante existe cependant entre tous les systèmes: le caractère discriminant des mesures prises. Qu'elles soient globales ou locales, de nature paramétrique ou non, elles ont comme rôle de distinguer les signatures authentiques des fausses.

Les différentes mesures prises lors de la *comparaison* doivent ensuite permettre de rendre une décision sur l'identité du scripteur. C'est l'étape de la *validation*. La procédure à suivre pour rendre la décision diffère, là aussi, selon les systèmes. Les mesures peuvent être combinées selon un modèle précis afin de donner une valeur unique, ou bien constituer les paramètres d'un espace multidimensionnel pour former un vecteur de la même dimension. La nature de la décision peut également varier selon les systèmes. Dans certains cas, il peut s'agir d'une décision binaire (*authentique* ou *faux*), tandis que dans certains autres cas, cette décision peut s'exprimer sur un continuum allant du *complètement authentique* au *complètement faux*. Cela permet d'exprimer un taux de confiance sur l'authenticité de la signature traitée.

Chacune des étapes de la procédure de vérification de signatures peut donc varier beaucoup selon les différentes techniques développées à ce jour. L'importance et la complexité de chacune de ces étapes peuvent rendre plus ou moins difficiles la distinction et la séparation de chacune d'entre elles. Le modèle fourni à la Figure 1.1 représente, d'une certaine façon, une solution « générique » au problème de la vérification de signatures, pouvant ainsi servir de référence dans la suite de ce texte.

1.1.3 Les solutions proposées: deux approches

Si plusieurs caractéristiques permettent de distinguer les différentes solutions proposées, une se démarque par son caractère fondamental: il s'agit de la nature de l'information traitée. La variable qui est déterminante dans ce cas est le moment de production de la signature. Selon les applications, la signature peut être produite pendant le déroulement de l'étape d'acquisition, ou bien elle peut avoir été produite a priori.

Si l'acquisition est faite au moment même de la production de la signature, il est possible de capter de l'information de nature *dynamique*. Une tablette à digitaliser effectuant un échantillonnage régulier de la position de l'outil d'écriture permet, par exemple, une représentation traduisant la progression spatio-temporelle de la signature.

Si, au contraire, la procédure d'acquisition se déroule *après* la production des spécimens de signatures, la seule représentation disponible étant l'image résultant du geste d'écrire,

alors l'information est de nature *statique*. Après digitalisation, la signature est représentée sous la forme d'une matrice bidimensionnelle dont chacun des éléments représente une portion de l'image. La valeur de chacun des éléments représente le niveau de luminance associé à la portion d'image correspondante.

Une récente revue de la littérature [LECL94], constituant une mise à jour de [PLAM89], résume les différentes techniques développées pour les deux types d'approches. Sa lecture permet de constater que les techniques *en-ligne* (utilisant l'information *dynamique*) offrent de meilleures performances. Cela explique, en partie peut-être, leur prédominance en nombre par rapport aux systèmes *hors-ligne* (utilisant l'information *statique*).

1.1.4 Les caractéristiques intéressantes

Un des facteurs expliquant les performances supérieures des systèmes *en-ligne* est précisément la dimension temporelle des signaux. Puisque à chaque échantillon de position (x,y) correspond une valeur de temps t, il est possible de construire différents signaux - position, vitesse ou accélération. La littérature montre que l'information la plus pertinente est obtenue en utilisant la signature complète d'un individu [PARI89], de laquelle sont extraits des profils de vitesse [PLAM88]. Cela tend à confirmer l'avantage des méthodes *en-ligne* sur les systèmes *hors-ligne*.

Les systèmes appartenant à cette seconde catégorie ne doivent pas pour autant être rejetés du revers de la main, car certaines applications nécessitent ce genre de traitement. Par exemple, l'authentification d'un chèque ou de tout autre document signé ne peut se faire qu'à partir d'informations *statiques*. L'avantage des systèmes hors-ligne est de permettre une authentification *a posteriori*, donc ne nécessitant pas la présence du scripteur.

Par contre, pour des applications de type contrôle d'accès, les systèmes *en-ligne* présentent la meilleure solution. Pour de telles applications, la présence du scripteur est nécessairement assurée et ne pose pas de contraintes supplémentaires.

Une contrainte qui est commune aux deux approches s'impose à la constitution de la banque de références utilisée. Puisque cette banque contient les spécimens de signatures qui servent à valider l'identité des scripteurs, il est essentiel que sa création se fasse dans un environnement contrôlé. Que ce soit pour des signaux *dynamiques* ou *statiques*, ils doivent être produits de façon telle que leur authenticité ne pose aucun doute.

1.2 L'inférence de la séquence

1.2.1 L'utilité de la séquence

La situation est la suivante: les techniques *hors-ligne* sont indispensables, mais les meilleurs résultats sont obtenus avec les techniques *en-ligne*. La solution à ce problème

serait l'application, l'adaptation des techniques *en-ligne* pour les systèmes *hors-ligne*. Mais la donnée manquante dans les systèmes *hors-ligne*, et qui est responsable de la supériorité des systèmes *en-ligne*, c'est l'information temporelle.

Si le temps pouvait être associé à l'information présente dans l'image, il serait possible d'établir un séquencement temporel de l'information présente, d'établir un ordre d'apparition de cette information, bref une *séquence*. La séquence connue dans l'image, toutes les ambiguïtés créées par les croisements ou les chevauchements de traits seraient dissipées.

Mais même si la séquence à l'origine du tracé d'une signature est connue, la notion temporelle reste relative: il est impossible de savoir le temps réel pris pour produire un tel tracé. Cependant, il est permis d'envisager l'application de certaines transformations pouvant fournir des informations sur les variations de vitesse entre les différentes portions d'un tracé.

Quelles que soient les techniques utilisées, les vitesses ainsi obtenues ne sont évidemment que relatives et ne sont pertinentes que dans la mesure où elles servent à mettre en relation les éléments d'un même tracé. Les différents tracés peuvent alors être comparés sur la base de leurs variations intrinsèques de vitesse plutôt que sur des valeurs absolues de vitesse.

Une séquence identifiée dans l'image permet de déduire une liste de points la représentant (une forme de squelette), qui peut ensuite servir à l'identification de paramètres s'apparentant à des vitesses. Parmi les techniques envisageables pour une telle tâche, il y aurait, entre autres, la transformation en série de Fourrier ou une décomposition en portions d'ellipses.

Les résultats obtenus par ces techniques pourraient être raffinés à l'aide de mesures prises directement dans l'image en niveaux de gris. Par exemple, les variations locales de l'intensité ou de la largeur du tracé peuvent peut-être indiquer des variations dans la vitesse de traçage.

Peu importe les techniques retenues, la condition *sine qua non* pour leur application demeure l'inférence dans l'image de la séquence des traits de la signature manuscrite. Dans une perspective plus globale de développement d'un nouveau système *hors-ligne* pour la vérification de signatures manuscrites, il est primordial de définir une technique permettant d'atteindre cet objectif préliminaire que constitue l'inférence de la séquence.

1.2.2 Les techniques existantes

Certains auteurs ont déjà tenté d'inférer la séquence dans une image de signature manuscrite et ce, avec un certain succès. Les lignes qui suivent constituent un résumé de ces différentes techniques.

Lee et Pan: Lee et Pan, dans [LEE 92], proposent d'utiliser la séquence pour positionner des *points critiques* tout au long d'un tracé manuscrit. La séquence est inférée à l'aide de sept heuristiques appliqués aux segments obtenus après squelettisation de l'image. Avant l'application des heuristiques, le squelette doit cependant être corrigé afin d'éliminer certains artefacts qui se retrouvent généralement dans les jonctions et les hampes, ainsi que le bruit introduit lors de la production du tracé ou de sa binarisation, comme l'illustre la Figure 1.2.

Les heuristiques utilisés pour inférer la séquence sont regroupés en deux classes: les règles de bas niveau et de haut niveau. Les règles de bas niveau, au nombre de cinq, concernent les décisions à prendre localement, au niveau du squelette.

Les deux premières règles de bas niveau concernent l'ordre de traçage des pixels du squelette. La première règle permet de choisir le bon pixel lorsque le pixel courant a plus d'un voisin non tracé. La seconde règle impose le choix d'un pixel identifié comme point de jonction si un tel pixel est le voisin immédiat du pixel courant.

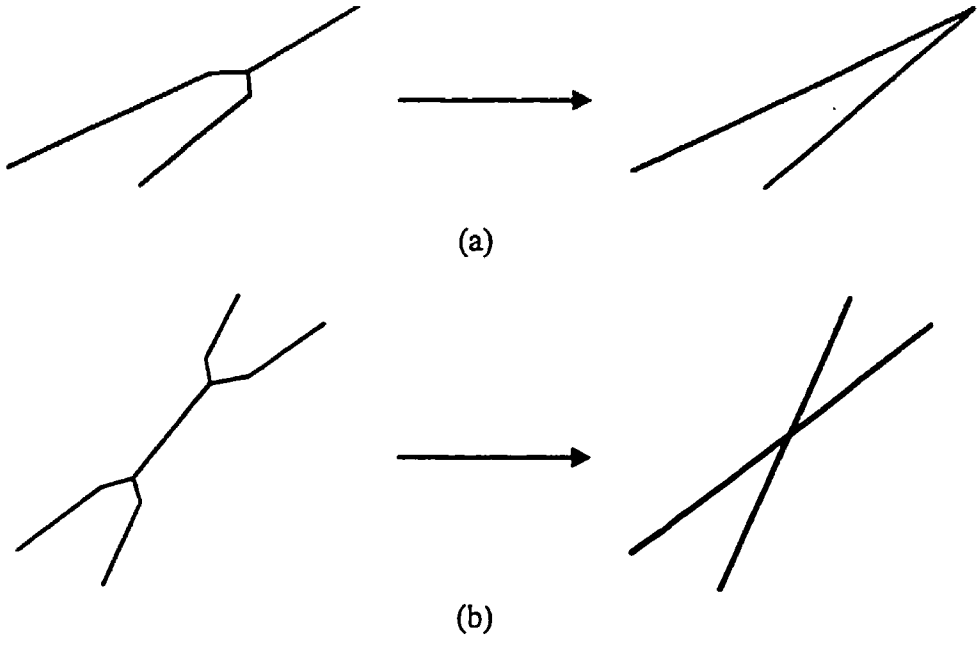


Figure 1.2 Illustration des erreurs de squelettisation devant être corrigées avant l'inférence de la séquence. (a) Correction d'une boucle ou d'une hampe. (b) Correction d'un croisement de traits.

Les troisième et quatrième règles de bas niveau concernent le traitement des points de jonction. Pour un point de jonction composé d'un nombre pair de segments, le prochain pixel identifié comme appartenant au segment est celui présentant la moins grande variation directionnelle. Lorsque le nombre de segments reliés à un même point de jonction est impair, un de ces segments doit être considéré comme un début ou une fin de trait. Le segment choisi sera celui présentant la plus grande variation directionnelle par rapport aux autres segments.

La cinquième et dernière règle de bas niveau concerne les rebroussements dans le tracé. Les auteurs supposent qu'il y a rebroussement dans le tracé lorsque, après avoir atteint la fin d'un trait, la dernière jonction impaire rencontrée possède un segment non traité. Ce segment est alors automatiquement choisi comme prochain point de départ.

Les règles de haut niveau traitent de l'ordonnancement des segments selon des critères globaux. Deux règles de cette nature sont utilisées ici. La première de ces règles permet de pondérer chacun des traits en fonction de son orientation. À chaque paire de pixel du trait est associée une mesure de performance qui tient compte de l'orientation des pixels dans le plan. La valeur associée à chacune des huit directions possibles a été établie à partir d'une affirmation concernant l'écriture humaine. Les auteurs affirment que les mouvements d'écriture vont normalement de la gauche vers la droite et du haut vers le bas. La mesure de performance permet d'inverser la séquence d'un trait selon la valeur de cette mesure. Si elle est négative, le sens est automatiquement inversé (une fois inversée la mesure prend automatiquement une valeur positive étant donnée la symétrie des valeurs associées aux huit directions).

La seconde règle de haut niveau traite des déplacements entre les traits. Pour ce faire, deux indices de performance sont calculés. Le premier indice s'inspire de la règle précédente en se basant sur la même affirmation concernant le sens de propagation de l'écriture. Cet indice est calculé en observant les positions des fins de traits. Le second

indice concerne les déplacements entre la fin d'un trait et le début du suivant. On tente ici d'ordonner les traits de façon à minimiser les déplacements entre les traits. Cette règle tente donc de maximiser la valeur du premier indice tout en minimisant la valeur du second.

La séquence ainsi identifiée sert à positionner le long du tracé un certain nombre de points critiques. C'est le seul usage réel de la séquence car les auteurs prétendent que la position et le nombre de ces points critiques sont plus significatifs que le comportement de la courbe les reliant. La méthode de positionnement de ces points permet de détecter les maxima et minima ainsi que les points d'inflexion de la courbe que représente la séquence. Ce sont ces points, plus les extrémités et les croisements préalablement identifiés, qui servent à la comparaison avec les autres spécimens dans la procédure d'authentification des scripteurs.

Étant donné le rôle de la séquence dans ce système, il n'est pas nécessaire d'identifier la vraie séquence, mais bien d'identifier de façon consistante une séquence, permettant ainsi que les points critiques soient, pour un scripteur donné, positionnés sensiblement de la même façon.

Boccignone, Chianese, Cordella et Marcelli : Une autre méthode d'extraction de la séquence est proposée dans [BOCC93]. Les étapes de traitement sont semblables à celles

proposées dans [LEE 92] mais les objectifs sont différents. Ici la signature est squelettisée à l'aide d'une *transformation en axe médian* (*MAT*) [ROSE77]. Du squelette ainsi obtenu, certains points sont isolés afin de servir à la correction du squelette par approximation polygonale. La figure qui suit est un exemple de la procédure proposée, appliquée à la lettre *A*.

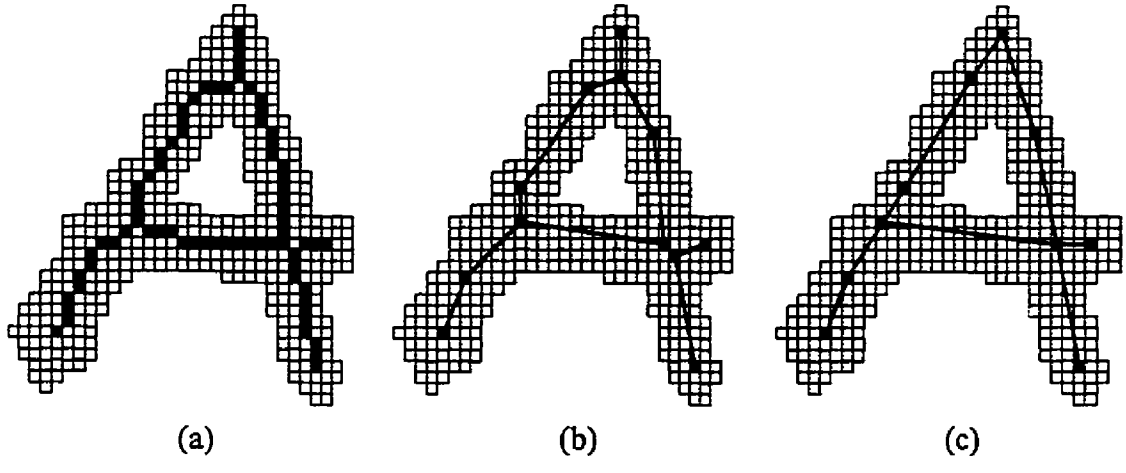


Figure 1.3 Synthèse du traitement proposé dans [BOCC93]. La méthode est ici appliquée à l'allographe *A*. (a) Transformation en axe médian (*MAT*). (b) Approximation polygonale (*PMAT*). (c) Correction des distorsions (*CPMAT*).

À chaque point constituant le squelette est associée de l'information sur la largeur du trait original. Les informations retenues permettent de fusionner les segments du squelette selon un critère de bonne continuité. Ce critère tient compte des trois facteurs suivants:

1. l'angle entre les segments;

2. le ratio entre les longueurs du plus petit et du plus long segment d'une paire;

3. la différence entre les largeurs des segments d'une paire.

Le squelette obtenu, après correction, est composé de segments de droite, ce qui ne reflète pas vraiment le mouvement sous-jacent au tracé. Pour une application en reconnaissance de caractères cela est acceptable, ce qui n'est pas le cas pour la vérification de l'identité. La même remarque s'applique concernant le traitement des rebroussements, qui semble ici inexistant.

Privitera et Plamondon: Plus récemment, une méthode fut proposée pour segmenter les images de mots en composantes. Dans [PRIV95], les auteurs s'inspirent du comportement humain pour le recouvrement de l'information temporelle dans une image d'un mot manuscrit. Cette technique procède en identifiant des topologies particulières sur les contours des mots. Ces topologies permettent d'identifier et de positionner les discontinuités générées lors de la production d'un mot.

Premièrement, des points caractéristiques sont identifiés sur les contours des mots. Ces points, correspondant à des maxima de courbures, permettent de situer les discontinuités possibles. Les discontinuités sont ensuites caractérisées en fonction du nombre et du type de points caractéristiques présents à l'intérieur d'un rectangle dont la taille peut varier. En

plus du nombre de points, le comportement du tracé à l'intérieur de ce rectangle permet l'identification de la discontinuité traité.

Un fois les discontinuités identifiées, la segmentation du mot est réalisée en interprétant les différentes discontinuités. Cette interprétation concerne principalement le couplage des différents segments formant une discontinuité. Les règles qui guident ce couplage sont inspirées des lois de Gestalt (bonne continuité, fermeture des contours, etc.).

Il est ensuite possible de reconstruire la trajectoire du tracé en appliquant les heuristiques concernant le geste d'écriture; par exemple, en débutant la reconstruction à partir de l'extrémité de la *composante* [PLAM92a] la plus près du coin supérieur-gauche de l'image.

Doerman et Rosenfeld: Une dernière méthode proposée dans la littérature tente justement de bonifier les trois précédents systèmes. Doerman et Rosenfeld [DOER91] prétendent en effet que la seule application de règles heuristiques imposées à un squelette de signature n'est pas suffisant pour extraire convenablement la séquence d'un tracé. Ils proposent donc une série d'indices détectables directement dans une image en niveaux de gris. Ces indices sont regroupés en trois catégories; globale, régionale et locale.

Les indices globaux sont ceux qui concernent l'aspect général de la signature, comme le sens de propagation du tracé qui est supposé de gauche à droite, l'énergie utilisée pour la construction de la signature (les retracements qui minimisent les levées de crayon, les déplacements entre la levée et la pose du crayon) et finalement le comportement d'un trait que l'on suppose se terminer généralement plus bas qu'il n'a débuté.

Les indices régionaux, eux, concernent la consistance de certains tracés et les relations entre les traits. Par exemple, on suppose que les boucles en haut de la ligne médiane de la signature sont généralement faites dans le sens anti-horaire et que les boucles sous la ligne médiane sont généralement faites dans le sens horaire, et si certains scripteurs font le contraire, ils le font de façon consistante.

Finalement les indices locaux sont des particularités que l'on peut observer aux extrémités des traits, comme des variations d'intensité, des *plumages* (« feathering ») ou des *crochetages* (« hooking »), ou dans des portions de boucles, comme des *striations* qui se forment de l'intérieur vers l'extérieur. Ces indices permettent tous de déterminer le sens du déplacement du crayon lorsqu'il est posé ainsi qu'entre deux traits.

Ces derniers indices peuvent varier en importance selon le scripteur ainsi que selon la nature de l'outil et de la surface d'écriture. Particulièrement la *striation* qui est exclusivement un phénomène associé aux stylos à bille.

1.3 Proposition de recherche

1.3.1 Les objectifs visés

La nécessité de développer une nouvelle méthode d'inférence de la séquence s'impose en partie par le fait que les techniques existantes ne répondent pas aux exigences requises pour le champ d'application envisagé. Le but étant ici le développement d'une nouvelle approche pour la vérification *hors-ligne* des signatures manuscrites qui serait basée sur certaines des techniques développées pour les systèmes *en-ligne*. La réalisation de cette nouvelle approche hybride repose sur la connaissance la plus exacte possible de la séquence à l'origine du tracé de la signature.

Or, aucune des méthodes présentées précédemment ne semble conçue expressément à cette fin. En effet, la technique proposée dans [LEE 92] utilise la séquence inférée uniquement pour positionner des points critiques, points qui servent ensuite à la vérification de l'identité.

Pour ce qui est de la seconde technique proposée [BOCC93], elle semble destinée à la reconnaissance de caractères; la séquence inférée sert plutôt à l'interprétation sémantique du tracé et ne requiert donc pas nécessairement une grande précision d'un point de vue spatio-temporel.

La technique proposée dans [PRIV95] est destinée à la segmentation des mots manuscrits. Elle utilise des heuristiques relativement bien adaptés au contexte de l'écrit. Ces mêmes heuristiques ne peuvent s'appliquer directement au traitement de signatures manuscrites.

Quant à la dernière technique proposée, [DOER91], il s'agit plus d'une proposition d'indices pouvant permettre d'ajouter certaines informations sur la progression du tracé. Mais puisque la présence de ces indices n'est pas garantie, il est impossible de les utiliser seuls.

À la lumière de l'analyse de ces différentes méthodes et en considérant le champ d'application concerné, il est possible de définir certains objectifs spécifiques que la méthode proposée doit rencontrer afin d'espérer des résultats intéressants:

1) L'algorithme de squelettisation à utiliser doit être adapté au contexte de l'écriture manuscrite de façon à éviter une étape de traitement secondaire pour la correction du squelette.

2) L'inférence de la séquence ne doit pas se faire simplement en identifiant un squelette dans l'image et en y appliquant des règles heuristiques spécifiques.

3) L'information que constitue la *banque de références* nécessaire à la vérification de signatures doit être utilisée à son plein potentiel.

Ces objectifs permettent d'orienter le développement de la nouvelle technique d'inférence de la séquence dans l'image d'une signature manuscrite. Les résultats de cette technique doivent ensuite servir à la vérification de l'identité par des techniques dérivées de celles utilisées par les systèmes *en-ligne*. Il faut donc que la séquence inférée soit la plus semblable possible de la vraie séquence à l'origine du tracé.

1.3.2 Description de la méthode proposée

La technique d'inférence de la séquence qui est proposée ici n'est valable que dans un contexte de vérification de signatures. Car le mécanisme d'inférence dans l'image est guidé par l'information contenue dans la référence à laquelle elle doit être comparée. Ce qui, à première vue, peut sembler un inconvénient ou une limitation est en réalité un avantage.

Si le signal de *test* a été produit par le même scripteur que celui qui a produit la *référence*, les séquences doivent être semblables. Si ces séquences sont semblables, alors l'information que contient la *référence* doit faciliter la procédure d'inférence dans l'image de *test* et donc produire une meilleure séquence. Dans le cas où l'image de *test* n'a pas été produite par le même scripteur, l'information que contient la *référence* ne facilite en aucun cas l'inférence de la séquence; cette dernière peut ne pas refléter la séquence à l'origine de l'image. Mais puisqu'il s'agit dans ce cas d'une fausse signature, la séquence inférée peut

présenter de grandes différences avec celle de *référence*. Ces différences peuvent servir à rendre une première décision sur l'identité du scripteur.

Donc, l'utilisation du signal de référence pour guider l'inférence de la séquence peut offrir l'avantage d'effectuer une identification des *faux aléatoires*; certaines signatures pourraient être rejetées sur la base des résultats de la procédure d'inférence.

Puisque l'inférence de la séquence est guidée par les informations contenues dans la *banque de références*, il est important que cette banque soit constituée par des signaux qui contiennent de l'information sur la séquence. La meilleure solution consiste donc à utiliser comme signaux de *référence* des signaux *dynamiques*. Cela peut sembler en contradiction avec l'objectif global, qui est de concevoir un nouveau système de vérification *hors-ligne*. Pas vraiment, puisque tout système de vérification de signatures nécessite une phase d'abonnement pendant laquelle le nouvel abonné doit produire un ou plusieurs spécimens de sa signature et ce, dans un environnement contrôlé, afin de s'assurer de l'authenticité de ces spécimens. Ce n'est que pour cette étape d'abonnement que la signature de l'abonné doit être échantillonnée de la sorte.

Le système proposé procède donc en comparant une *image* de signature manuscrite avec un *signal dynamique* et tente d'inférer la séquence dans l'image à partir des informations contenues dans le signal dynamique. Pour qu'il puisse y avoir une comparaison possible

entre les deux signaux, ceux-ci doivent premièrement être transformés de façon à ce qu'ils soient comparables. Deux procédures différentes permettant de représenter chacun des signaux sous la forme d'un *graphe* constituent les étapes de *prétraitement* des signaux. Étant donnée la nature différente des signaux, il est exclu d'utiliser la même procédure de prétraitement dans les deux cas. Les signaux sont traités par des procédures distinctes, mais ayant le même objectif, soit celui de construire un graphe représentant le signal traité. Les deux graphes résultant des procédures de prétraitement constituent des représentations de plus haut niveau de l'information contenue dans les deux signaux. Ces étapes de prétraitement des signaux sont représentées par les modules 1 et 2 du *DFD* de la Figure 1.4.

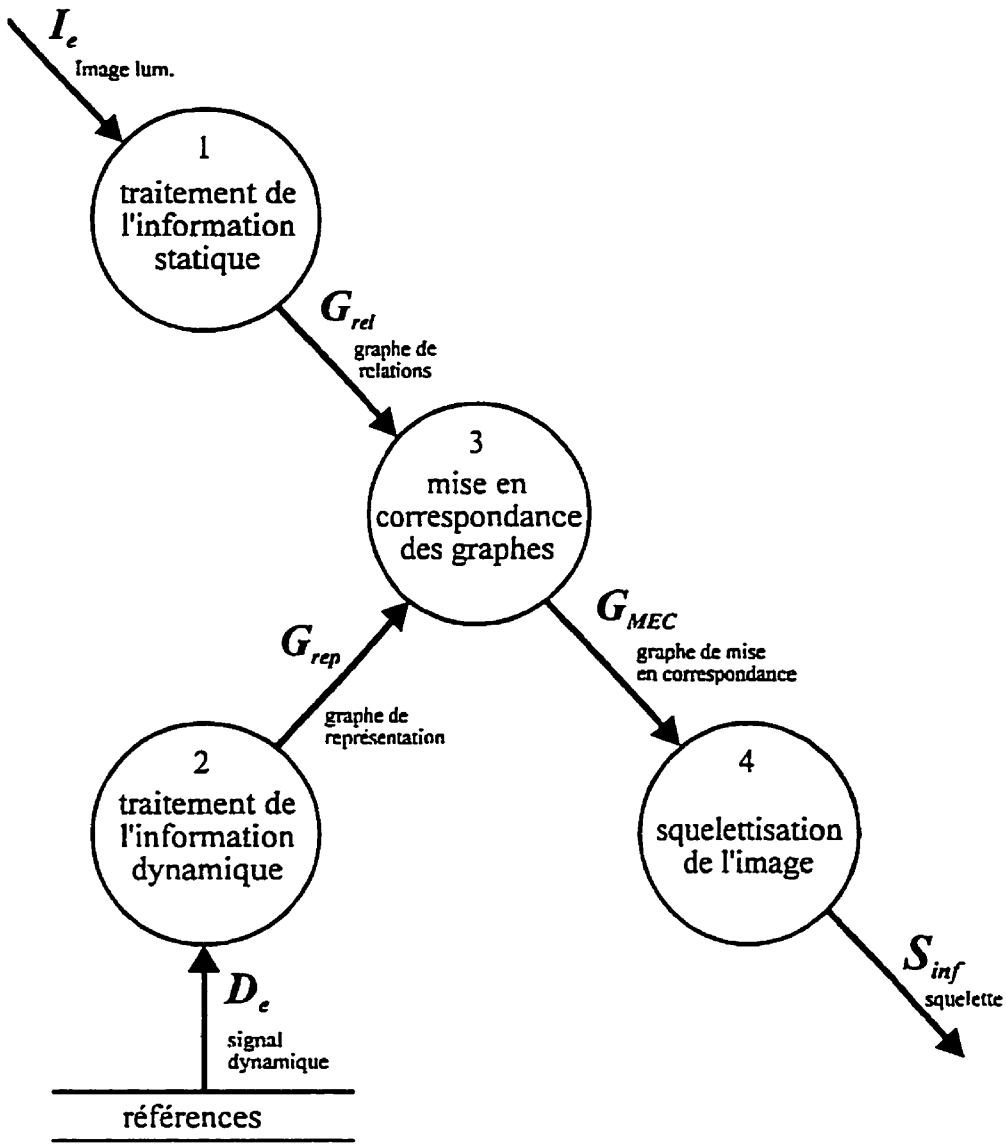


Figure 1.4 *DFD* synthèse de la technique proposée pour l'inférence de la séquence.

Une caractéristique fondamentale distingue cependant les deux graphes. Le graphe de représentation G_{rep} dérivé de la séquence dynamique constitue en fait une chaîne (qui est un cas particulier de graphe): chaque noeud n'a au maximum que deux voisins. Cela a

pour effet d'établir un chemin unique dans ce graphe. À l'opposé, dans le graphe de relations G_{rel}, les noeuds peuvent avoir plus de deux voisins. Cela est dû au fait que, dans ce graphe, il existe des noeuds représentant les croisements de traits. Il est alors impossible d'établir un chemin de façon triviale. La solution au problème du séquencement passe donc par la recherche d'un chemin dans ce graphe.

La recherche de ce chemin se fait par le biais de la procédure de *mise en correspondance des graphes*. Il s'agit d'identifier dans les deux graphes, les éléments qui représentent des portions de tracé similaires. En associant ainsi des éléments du graphe de relations G_{rel} avec des éléments du graphe de représentation G_{rep}, il devient possible d'établir un parcours dans le graphe de relations sur la base des associations faites avec les éléments de G_{rep}. Le résultat de cette opération peut être représenté comme un sous graphe orienté de G_{rel}.

C'est à partir de ce sous-graphe orienté que la procédure de squelettisation s'effectue. L'utilisation de l'information contenue dans le sous-graphe au niveau de l'image permet de déterminer la progression du tracé. Cette même information rend possible la squelettisation par suivi de contours [CHOU92] en permettant d'identifier les points de départ nécessaires à l'application d'une telle technique.

1.3.3 Les caractéristiques spécifiques à la méthode proposée

Outre l'utilisation de la *banque de références* comme information *a priori* pour guider l'inférence de la séquence, la technique proposée se distingue des techniques existantes par un certain nombre d'autres caractéristiques.

Premièrement, il faut souligner le fait que la squelettisation s'effectue seulement après avoir établi une séquence dans l'image. Cela évite, entre autres, la phase de correction du squelette nécessaire lorsque la squelettisation se fait préalablement à l'identification de la séquence [BOCC93] [LEE 92].

Puisque l'identification de la séquence se fait avant la squelettisation, la technique utilise plutôt l'information des contours de l'image binarisée afin de former une représentation de haut niveau de la signature. De cette façon, il est possible d'identifier des portions de tracé plus précisément qu'en utilisant seulement un squelette, puisque les contours constituent l'enveloppe du tracé. Même si cette information supplémentaire semble redondante pour les portions simples de tracé, elle est tout à fait justifiée pour interpréter plus facilement les discontinuités du tracé.

L'utilisation des contours du tracé constitue un atout à un autre niveau: cela permet une estimation de la largeur moyenne du tracé. La valeur ainsi estimée est utilisée comme dénominateur commun pour le choix des différents paramètres nécessaires au traitement

de l'image de la signature. De cette façon tous ces paramètres deviennent dépendants du contexte et rendent les résultats indépendants du type d'outil d'écriture.

1.3.4 Plan du mémoire

La suite de ce mémoire sera donc consacrée dans un premier temps à la description de la méthode proposée pour l'inférence de la séquence et, dans un deuxième temps, à l'évaluation des performances de cette même méthode. Pour faciliter la lecture de ce document, étant donné la complexité du système proposé, le traitement est décomposé en quatre modules distincts. Chacun de ces modules est décrit dans un chapitre particulier.

Le **chapitre 2** décrit la procédure de *traitement de l'information statique*, soit la transformation de l'image de luminance I_e en un graphe de relations G_{rel}. Il y est question de la binarisation de l'image de luminance, de l'identification et du positionnement des différents points caractéristiques sur les contours du tracé, de la formation des zones caractéristiques ainsi que de la formation et de l'édition du graphe de représentation G_{rep}.

Parallèlement, le **chapitre 3** décrit la procédure de traitement de l'information dynamique. Il y est question de la transformation de la séquence échantillonnée D_e en un graphe de représentation G_{rep}. Les étapes intermédiaires, telles que le filtrage des échantillons, l'interpolation de la séquence, l'identification et le positionnement des points

caractéristiques ainsi que la formation du graphe de représentation G_{rep}, sont toutes décrites en détail.

La *mise en correspondance des graphes* fait l'objet du **chapitre 4**. Il y est question, entre autres, de la procédure de relaxation et de l'attribution de poids initiaux nécessaire à cette tâche, ainsi que de la procédure de mise en correspondance par croissance consistante. Ce dernier traitement vise à augmenter le nombre d'éléments du graphe de mise en correspondance G_{MEC} qui constitue le résultat de toutes ces opérations.

La dernière étape relative à l'inférence de la séquence, la *squelettisation de l'image*, est traitée au **chapitre 5**. Il y est question du positionnement de points par la technique de suivi de contours et de la méthode d'interpolation retenue pour relier ces points entre eux.

Le **chapitre 6** est consacré à l'analyse des performances du système proposé. Y sont décrites les différentes expériences conduites pour valider l'approche proposée ainsi que pour en identifier les points forts et faibles.

Finalement, le **chapitre 7** permet de conclure en rappelant les différents points innovateurs de la technique ainsi qu'en proposant certaines orientations pour d'éventuels travaux futurs.

2. TRAITEMENT DE L'INFORMATION STATIQUE

2.1 Description générale

La tâche commune à tous les systèmes de traitement d'images est l'identification et l'organisation de l'information pertinente présente dans l'image traitée. Pour augmenter les chances de succès d'une telle entreprise, il importe de bien connaître la nature des images à traiter et surtout à quel usage elles sont destinées.

Le but de ce travail est l'inférence d'une séquence de traits dans une image de signature manuscrite. L'image traitée est une image de luminance, digitalisée en 256 niveaux de gris. De cette image doit être extrait un graphe de relation G_{rel} dont les noeuds représentent des zones d'intérêt Z_{rel} (regroupant les *zones extremums* Z_{ext} et les *zones de discontinuité* Z_{disc}) dans l'image [Figure 2.2]. La Figure 2.1 montre le *DFD* schématisant le traitement à effectuer pour extraire ce graphe de l'image de luminance.

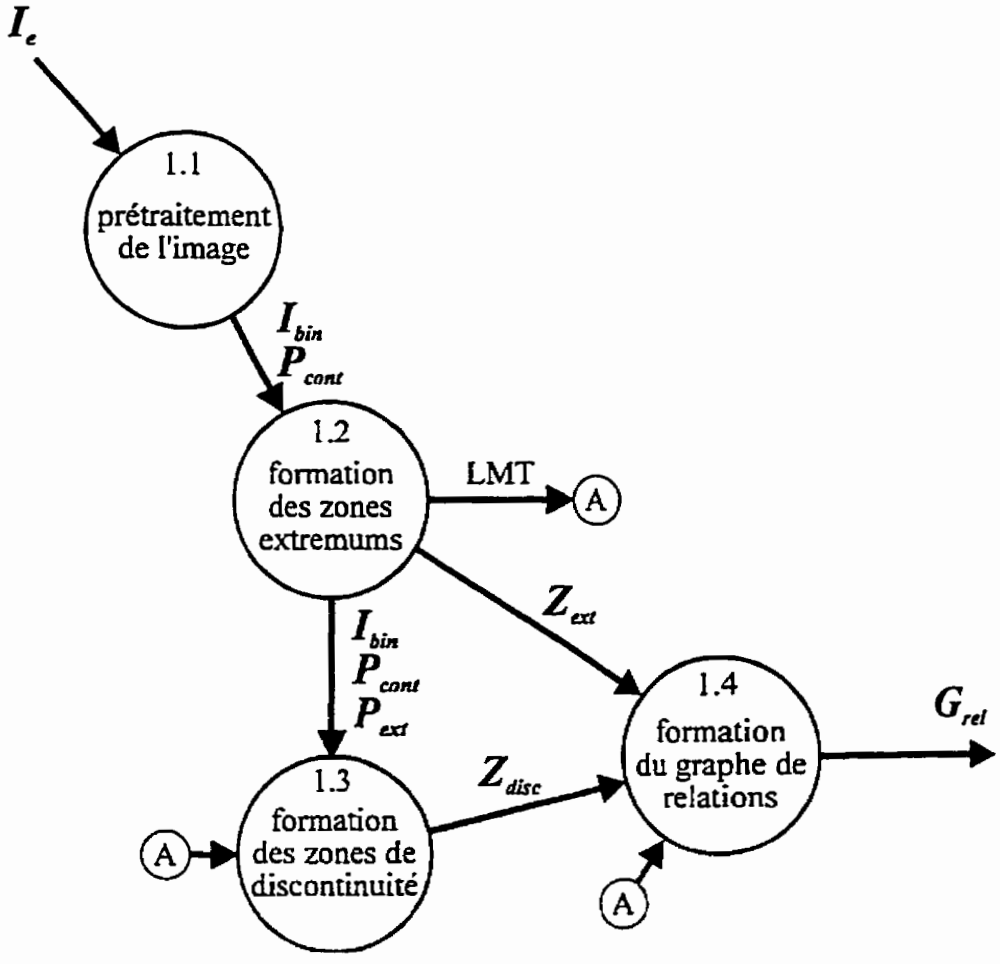
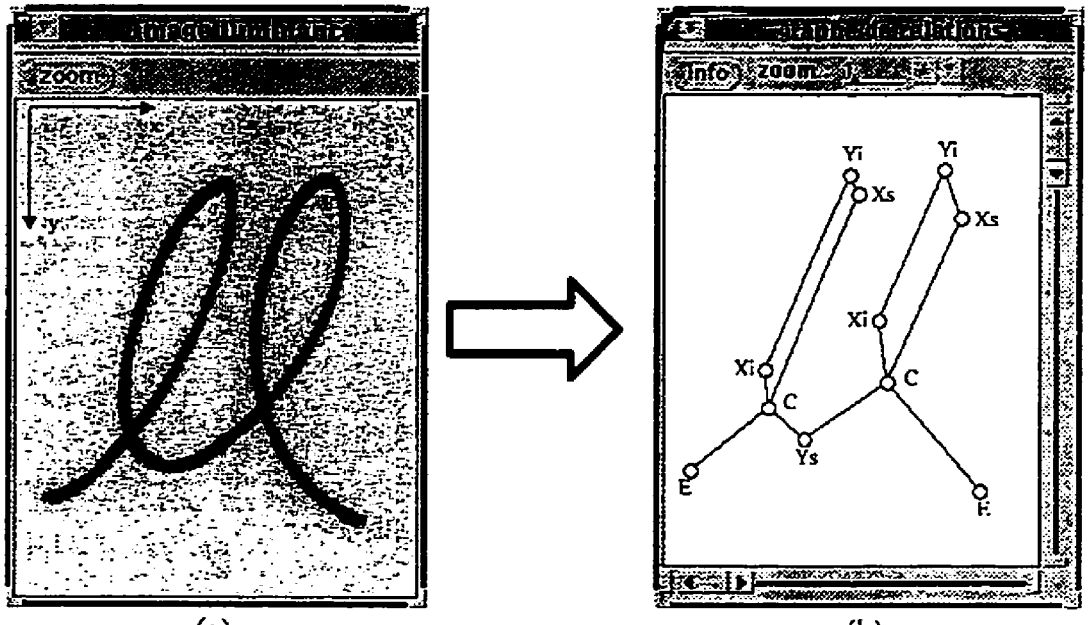


Figure 2.1 *DFD* schématisant le traitement de l'information statique.



(a) (b)

Figure 2.2 (a) Image de luminance I_e ; (b) le graphe de relations G_{rel} extrait de cette image.

Les zones d'intérêt sont formées de points caractéristiques identifiés sur les contours du tracé manuscrit. Dans un premier temps, la signature et ses contours doivent être identifiés dans l'image de luminance. Après cette étape de prétraitement, telle qu'illustrée à la Figure 2.3, l'information se réduit à une matrice identifiant les pixels appartenant au tracé (l'image binaire I_{bin}) et à des listes de points représentant les contours du tracé de la signature (P_{cont}).

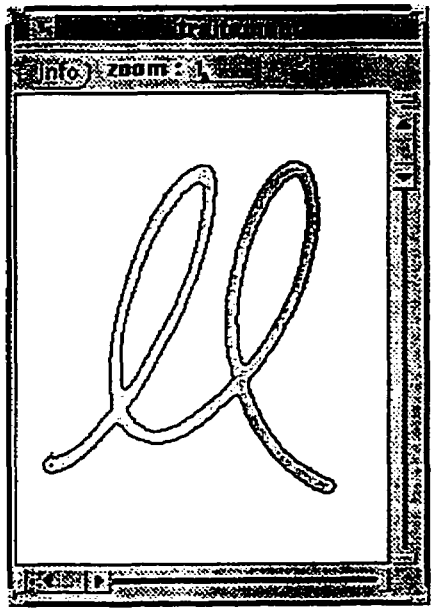


Figure 2.3 Résultat du prétraitement de l'image de luminance Ie: l'image binaire I_{bin} et les listes de points de contours P_{cont}.

C'est parmi les points des listes de contours P_{cont} que sont choisis les points caractéristiques qui sont à l'origine de la formation des zones d'intérêt Z_{rel} [Figure 2.4]. La formation de ces zones se fait en deux étapes distinctes selon la nature des zones à l'étude: premièrement les zones dites *extremums* (Z_{ext}), deuxièmement les zones de *discontinuité* (Z_{disc}).

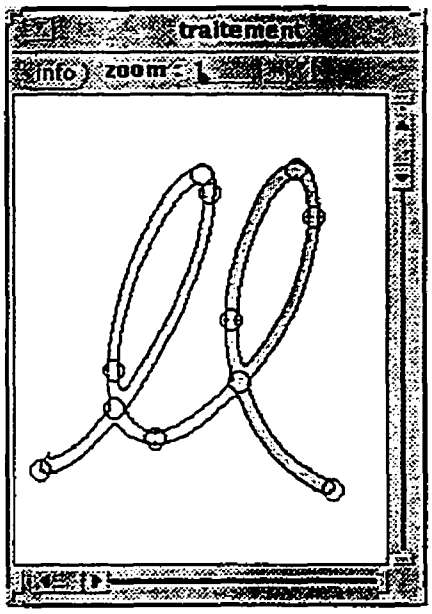


Figure 2.4 Image binaire I_{bin} dans laquelle sont identifiées les zones d'intérêt Z_{rel}, représentées par des cercles.

L'ensemble des zones formées permet la création du graphe de relations G_{rel} qui est l'aboutissement du traitement statique. Chaque zone d'intérêt Z_{rel} représente un noeud du graphe. Deux noeuds sont reliés par un arc si les zones qu'ils représentent sont elles-mêmes reliées par une portion de tracé dans l'image. Un exemple de graphe est donné à la Figure 2.2b.

Les sections qui suivent vont traiter plus en détail chacune des étapes permettant d'extraire de l'image de luminance I_e, le graphe de relations G_{rel} .

2.2 Prétraitement de l'image

Cette première étape du traitement statique a pour objet la binarisation de l'image ainsi que l'extraction des contours du tracé.

2.2.1 Nature des images traitées

Les images qui sont transmises au système sont des images dites de luminance : la valeur associée à chaque pixel de l'image est un indice de luminance qui traduit la quantité de lumière perçue lors de la digitalisation de l'image. Cet indice de luminance varie de 0 (noir) à 255 (blanc), les valeurs intermédiaires correspondent à autant de niveaux de gris. La référence à un point de l'image se fait à l'aide d'un système de coordonnées cartésiennes dont l'origine se situe dans le coin supérieur gauche de l'image [Figure 2.2].

Comme les images utilisées dans les différentes phases de développement du présent système ont été produites dans un environnement entièrement contrôlé (voir section 6.1), certaines hypothèses peuvent être émises, hypothèses qui vont permettre de justifier des choix ultérieurs.

1) Les signatures ayant été produites sur du papier blanc, le fond des images (« background ») est uniforme.

2) La production de chaque signature ayant été limitée à un rectangle de dimension fixe, l'orientation est considérée constante pour tous les spécimens produits par un même scripteur.

Dans un contexte différent, pour que ces deux hypothèses restent vraies, il faudrait que les images à traiter fassent l'objet d'un traitement préliminaire éliminant le fond des images [CHER95] et, si nécessaire, normalisant l'orientation des signatures.

2.2.2 Binarisation de l'image

2.2.2.1 Détermination du seuil

Binariser une image consiste à identifier tous les pixels formant un même objet. Généralement, un objet se distingue dans l'image par sa « couleur » (l'indice de luminance) qui va généralement créer un mode dans l'histogramme de luminance. C'est en déterminant les limites de ce mode (les seuils) que peut être identifié l'objet.

Les images de signatures manuscrites sont des images caractérisées par une distribution bimodale des luminances: l'histogramme de luminance d'une telle image se compose de deux bosses (les modes) séparées par une vallée [Figure 2.5]. Un des modes représente le fond de l'image et le second représente le tracé de la signature. Un seuil permet de séparer le fond du tracé : les pixels dont l'indice de luminance est inférieur au seuil sont associés

au tracé et les autres, au fond. Les modes d'une image de signature manuscrite ne sont généralement pas de proportions identiques. Le tracé de la signature occupant seulement une très faible surface de l'image en comparaison avec le fond, le mode associé au tracé est beaucoup plus faible que celui associé au fond, ce qui rend plus difficile le positionnement du seuil et, par le fait même, l'extraction du tracé de la signature.

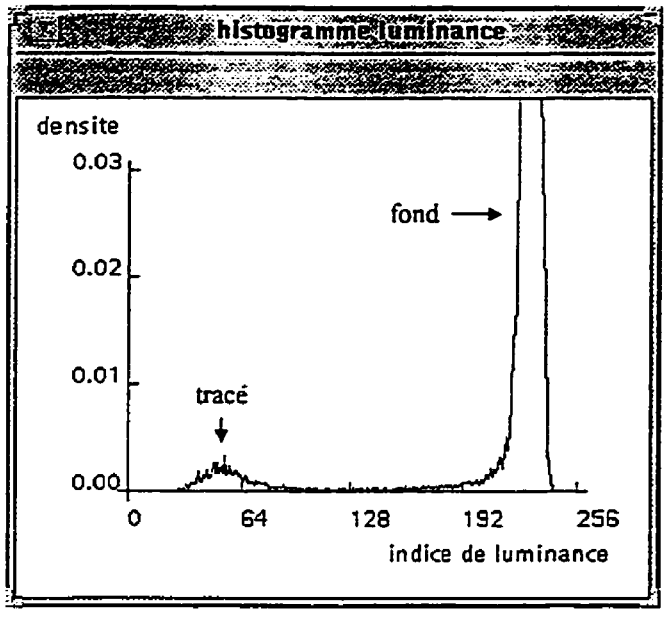


Figure 2.5 Histogramme de luminance associé à l'image en niveaux de gris présentée à la Figure 2.2 (a).

Les différentes techniques rencontrées dans la littérature pour choisir le seuil peuvent être classées en deux groupes. Il y a premièrement les techniques globales : une analyse de l'image dans son ensemble permet le choix d'un seuil unique pour toute l'image. L'autre groupe de méthodes propose le choix d'un seuil local pour chaque portion d'image. Ces dernières méthodes sont suggérées lorsque l'image présente de grandes variations de

contraste ou un fond non uniforme. Dans le cas présent, comme aucune de ces conditions n'est rencontrée, le choix s'est porté sur une méthode globale. Une récente revue de la littérature [TRIE95] propose la méthode de Abutaleb [ABUT89] comme étant la meilleure méthode globale. Ce choix n'est pertinent que dans la mesure où les images traitées présentent un fond uniforme, ce qui est le cas ici.

Abutaleb propose une méthode de seuillage basée sur une mesure d'entropie bidimensionnelle. L'histogramme des niveaux de gris ainsi que l'histogramme de la moyenne des niveaux de gris des huit voisins permettent de générer une matrice $m \times m$ dans laquelle chaque élément (s,t) représente la fréquence d'occurrence d'un pixel de niveau de gris s et dont la moyenne des voisins est t. L'allure de cet histogramme bidimensionnel, pour une image de signature manuscrite, correspond à ce qui est présenté à la Figure 2.6. Deux modes peuvent être identifiés: celui le plus près de l'origine correspond aux pixels du tracé et celui le plus éloigné, aux pixels du fond de l'image. Le seuil de binarisation doit donc se situer entre ces deux modes.

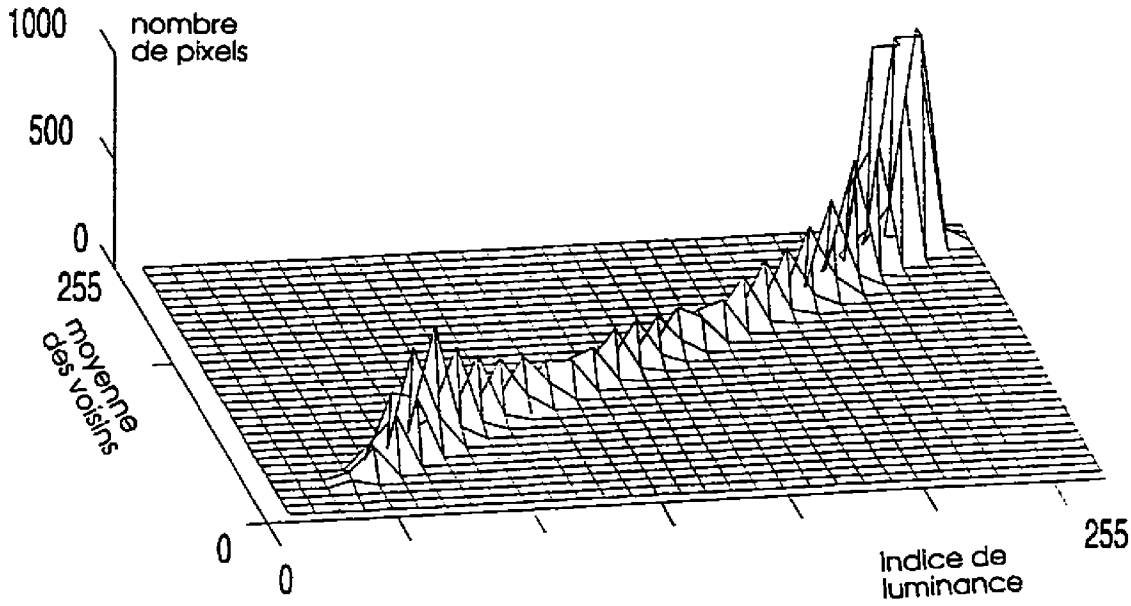


Figure 2.6 Histogramme bidimensionnel utilisé pour l'estimation du seuil de binarisation par une mesure d'entropie 2D.

Le seuil de binarisation est déterminé à l'aide d'un indice d'entropie $\psi(s,t)$. Cet indice d'entropie, tel que décrit par l'équation suivante, est calculé pour chacun des éléments p_{ij} de l'histogramme bidimensionnel.

$$\psi(s,t) = \ln[P_{st}(1-P_{st})] + \frac{H_{st}}{P_{st}} + \frac{H_{mm} - H_{st}}{1-P_{st}} \qquad (2\text{-}1)$$

où

$$P_{st} = -\sum_{i=1}^{s}\sum_{j=1}^{t} p_{ij} \qquad (2\text{-}2)$$

$$H_{st} = -\sum_{i=1}^{s}\sum_{j=1}^{t} p_{ij}\ln(p_{ij}) \qquad (2\text{-}3)$$

Le seuil de binarisation est l'indice pour lequel $\psi(s,t)$ est maximum. Cette technique détermine un seuil qui, en plus de poser une limite sur le niveau de gris du pixel (s_{max}), impose une contrainte sur la moyenne du niveau de gris de ses voisins (t_{max}). Est donc identifié comme appartenant au tracé, tout pixel dont le niveau de gris est plus petit que s_{max} et dont la moyenne des niveaux de gris des voisins est plus petite que t_{max} . En considérant ainsi le voisinage du pixel, cela augmente la robustesse de la méthode en présence de bruit.

2.2.2.2 Élimination du bruit

Même si la méthode de seuillage est efficace pour éliminer le bruit, il est possible que certains éléments indésirables soient quand même considérés comme faisant partie du tracé. Cela se produit lorsque l'image contient des taches de taille trop importante. Il faut alors inclure une connaissance a priori concernant la largeur des tracés. Des observations répétées montrent que la largeur moyenne des traits formant les signatures est d'environ 10 pixels. Théoriquement, le plus petit trait possible serait un cercle de diamètre égale à 10 pixels, ce qui correspond à un surface de 75 pixels. Donc, un objet de l'image binaire dont la surface, en pixels, est inférieure à τ_{pix} = 75 est considéré comme du bruit et est automatiquement éliminé.

2.2.2.3 Comblement des petits trous

Un autre problème qui peut survenir lors de la binarisation des images est celui dit des *petits trous*. Après le seuillage, il est possible que certains pixels du tracé aient été oubliés, formant ainsi des trous indésirables dans le tracé. Les petits trous ainsi formés seraient à l'origine de contours supplémentaires si rien n'était fait pour les éliminer. Le comblement de ces petits trous vise précisément à éviter ce problème.

Un trou est considéré *petit* si sa largeur est égale à un pixel, peut importe sa surface. Pour éviter d'avoir à mesurer la largeur de tous les trous de l'image, une technique alternative est proposée. Cette technique repose sur l'observation des contours du trou : un trou de largeur unitaire présente, à ses extrémités, un pixel qui est entouré de 3 pixels de contours (en considérant une connexité 4). En changeant les étiquettes des pixels ainsi identifiés et en répétant l'opération d'identification tant que des pixels changent d'étiquette, tous les petits trous sont comblés. La méthode est résumée dans le Tableau 2.1.

Cet algorithme s'applique à tous les pixels de fond qui rencontrent la condition de voisinage. Il se peut que certains pixels en bordure des contours, internes ou externes, rencontrent cette condition, même s'il ne s'agit pas de contours de petit trou. Ces pixels verront leur étiquette changée aussi. Ceci a donc comme effet d'éliminer certaines irrégularités sur les contours.

Tableau 2.1 Algorithme utilisé pour combler les petits trous

```
1: RÉPÉTER
2:     POUR tous les pixels p de l'image
3:         SI p est étiqueté comme fond de l'image ALORS
4:             Calculer n, le nombre de voisins de p appartenant au tracé (connexité 4)
5:             SI n ≥ 3 ALORS Changer statut du pixel
6:         FIN SI
7:     FIN POUR
8: TANT QUE le statut d'au moins un pixel a changé
```

Le résultat de ce dernier traitement donne l'image binaire I_{bin} utilisée pour le reste du traitement de l'information statique [Figure 2.1].

2.2.3 Extraction des contours du tracé

Les contours du tracé sont les points situés à la limite entre le tracé et le fond de l'image. Deux conditions doivent être satisfaites pour qu'un point soit considéré comme appartenant au contour d'un objet:

1) le point doit être étiqueté comme appartenant au tracé de la signature;

2) le point doit avoir au moins un de ses voisins (4-connexe) qui appartienne au fond de l'image.

Les points qui rencontrent ces conditions forment des listes 8-connexe. Ceci est dû à la seconde condition qui considère une connexité 4.

Afin de créer des listes de points représentant chaque contour fermé du tracé et que ces listes soient ordonnées et consistantes, une procédure d'identification doit être établie. La procédure définie concerne l'identification d'un point de départ et le mode de génération de la liste.

Dans un premier temps, il faut détecter un premier point qui rencontre les deux conditions citées précédemment. Ce premier point est trouvé par un balayage de l'image effectué de la gauche vers la droite et du haut vers le bas. Le point ainsi trouvé constitue le point de départ pour la liste $P_{cont}^{(i)}$ associée au i^e contour.

Le prochain point dans la liste $P_{cont}^{(i)}$ est toujours identifié à partir du dernier point trouvé et est choisi parmi les voisins de ce point. Le voisin retenu comme prochain point de contour doit évidemment satisfaire les deux conditions énoncées précédemment. Mais cela n'est pas suffisant car, parmi les huit voisins d'un point de contour, deux points satisfont toujours ces conditions. Il faut donc un critère supplémentaire afin de faire un choix

consistant parmi ces deux candidats. Le choix se fait en considérant le voisin précédent lors du déplacement d'un voisin à l'autre. Le point retenu est celui qui, lorsque le déplacement est effectué dans le sens anti-horaire, est précédé par un pixel appartenant au fond. Ainsi, tous les contours sont extraits de façon consistante : un observateur qui se déplacerait d'un point à l'autre du contour aurait toujours le tracé à sa gauche et le fond à sa droite.

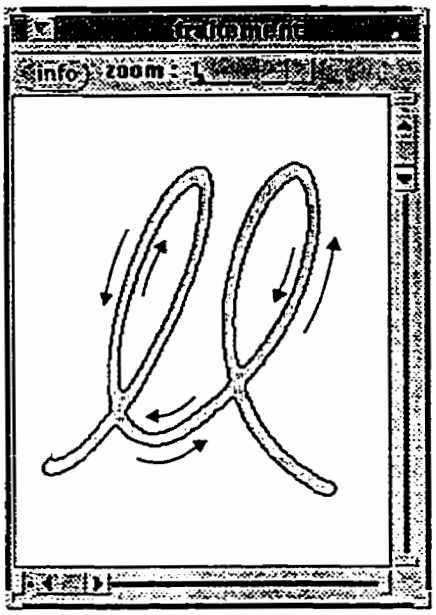


Figure 2.7 Exemple de contours extraits. Les contours externes sont extraits dans le sens anti-horaire et les contours internes, dans le sens horaire.

La liste des points d'un contour se termine lorsque le prochain point à insérer est le point de départ de celle-ci. La procédure d'extraction des contours est répétée tant que le

balayage de l'image identifie un point qui rencontre les deux conditions nécessaires et qui n'appartient pas encore à une liste de point de contour.

Ceci termine la partie prétraitement de l'image. De l'image échantillonnée I_e sont extraits le tracé de la signature (I_{bin}) et les contours de ce tracé (P_{cont}). C'est cette information qui sert à l'identification des différentes zones d'intérêt Z_{rel}, soit les zones extremums Z_{ext} et les zones de discontinuité Z_{disc} [Figure 2.1].

2.3 Formation des zones extremums

Les zones extremums se définissent comme des éléments du tracé représentant des extremums locaux de la signature. Les zones ainsi identifiées vont servir lors de la mise en correspondance avec les éléments semblables extraits de la séquence dynamique utilisée comme référence.

2.3.1 Localisation et identification des points extremums

Un extremum se définit comme une limite supérieure ou inférieure de la plage de variation d'une valeur. Le qualificatif local signifie que la plage de variation est définie localement plutôt que globalement. Dans le présent cas, les valeurs pour lesquelles sont appliquées la notion d'extremum sont les coordonnées cartésiennes des points des contours. Ceci crée donc deux catégories d'extremums - coordonnée X et coordonnée Y - comportant

chacune deux types d'extremum - inférieur et supérieur. Un point est donc considéré comme un extremum local inférieur (supérieur) si une de ses coordonnées est plus petite (plus grande) que celles de ses voisins immédiats sur le contour [Figure 2.8]. Les quatre type d'extremums possibles sont identifiées par :

Xi :	extremum inférieur en X	(à gauche)
Xs :	extremum supérieur en X	(à droite)
Yi :	extremum inférieur en Y	(en haut)
Ys :	extremum supérieur en Y	(en bas)

Chaque extremum est en plus qualifié en fonction de sa position relative par rapport au tracé. Un extremum est dit *absolu* si sa position par rapport au tracé peut être caractérisée par le même qualificatif que l'extremum, sinon il est dit *relatif*. Par exemple, un extremum **Xi** situé sur la portion de contour à gauche du tracé est un extremum absolu, tandis qu'un autre extremum **Xi** situé sur la portion de contour à droite du tracé est considéré comme relatif (noté **Xi'**). Ce qualificatif supplémentaire porte donc à huit le nombre d'extremums différents possibles. La Figure 2.9 donne un exemple de chacun d'eux.

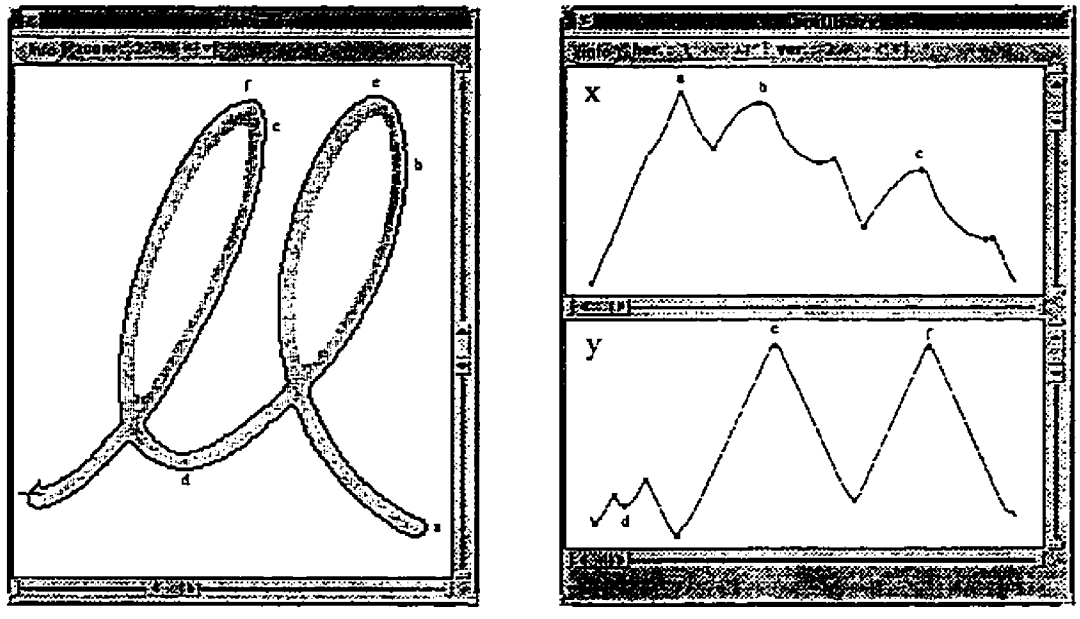


Figure 2.8 Profils des coordonnées (x,y) associées au contour externe du tracé représenté.

Les relations entre les extremums sont également identifiées par des qualificatifs décrivant les liens entre les extremums. L'*orientation* regroupe tous les extremums associés à un même axe (X ou Y) : deux extremums sont dits de *même orientation* s'ils sont tous deux des extremums en X (ou en Y). Le *type* est un sous-classement de l'orientation : des extremums sont de *même type* ou de *type contraire* selon qu'ils sont des extremums identifiant une même limite locale (supérieure ou inférieure) ou non. Si deux extremums sont de même type mais que leur position relative au tracé diffère, ils sont dits *complémentaires*.

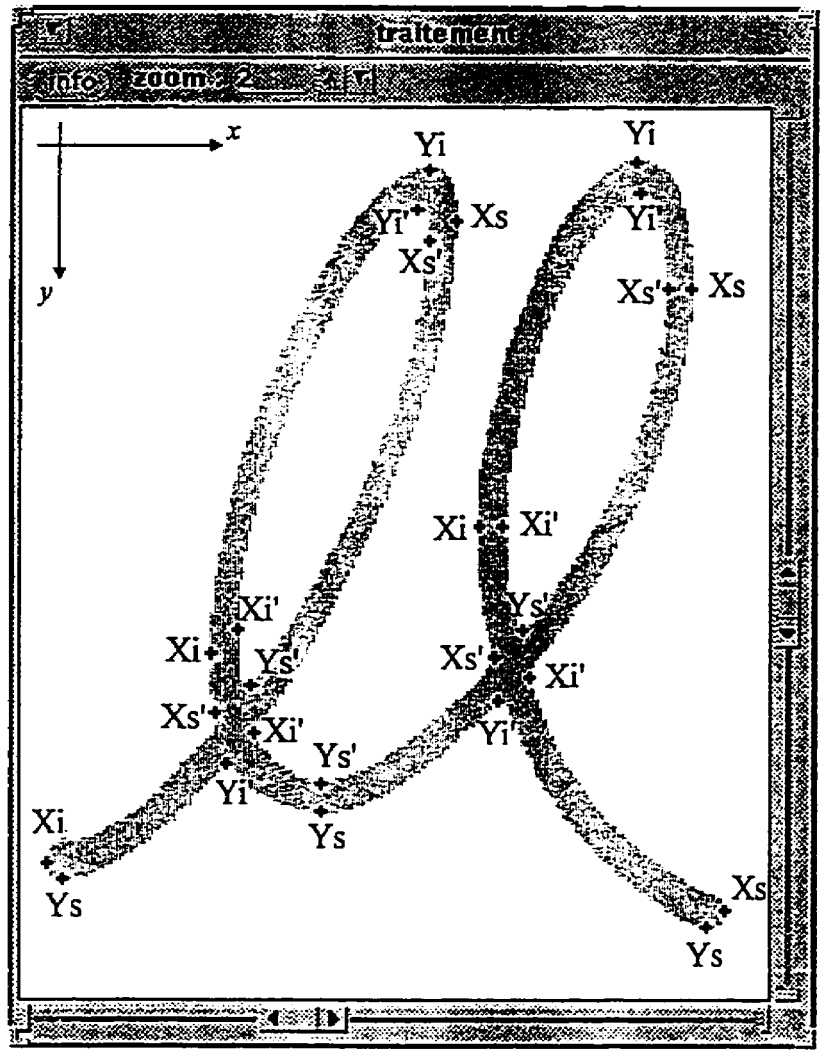


Figure 2.9 Image illustrant les 8 types d'extremum possibles. Un extremum *relatif* est identifié par le symbole *prime*.

2.3.2 Validation des points extremums

2.3.2.1 Identification des régions bruitées

Étant donné le caractère local du mécanisme d'identification des extremums, la présence de bruit sur les contours peut générer des extremums non pertinents. Il faut donc procéder à un filtrage des extremums afin de ne garder que ceux jugés pertinents. La pertinence d'un extremum dépend de la distance sur le contour qui le sépare de ses voisins de même orientation. La comparaison doit en effet se faire uniquement entre les extremums de même orientation (X ou Y), car il est impossible d'établir un quelconque critère de distance entre des extremums d'orientation différente.

Par souci de souplesse et d'adaptabilité, le choix de la distance seuil τ_{cont} est établi en fonction de la largeur moyenne du tracé *LMT*. Cette valeur est estimée à partir des couples d'extremums complémentaires pouvant être reliés par une droite incluse dans le tracé [Annexe A]. Le choix de τ_{cont} est basé sur l'hypothèse suivante:

> *Pour qu'une variation du tracé par rapport à sa direction soit significative, il faut que la variation de position, perpendiculairement à la direction, soit au moins égale à la largeur du tracé comme le montre la* Figure 2.10.

Toutes variations plus petites sont nécessairement attribuées à des mouvements indésirés de la main ou à des artefacts dus au fonctionnement de l'outil d'écriture. Donc des extremums de même orientation séparés par une distance sur le contour inférieure à la largeur du tracé sont nécessairement associés à du bruit. De cette hypothèse découle directement le choix du seuil pour la distance inter-extremum $\tau_{cont} = LMT$.

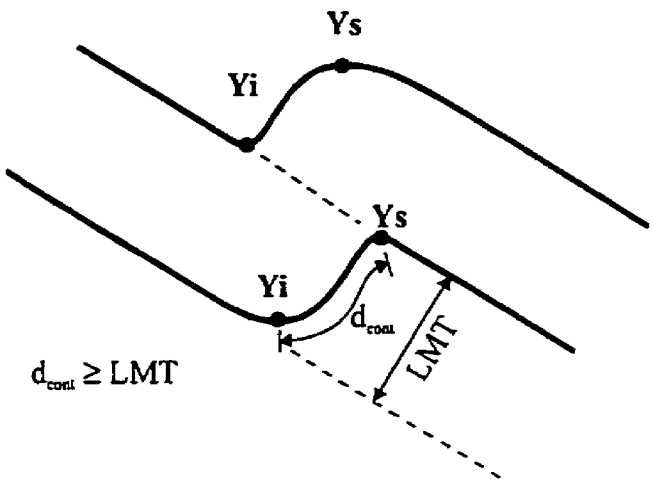


Figure 2.10 Illustration de la variation minimum considérée comme significative.

Si les distances sur le contour $d_{cont}(p_i p_j)$ et $d_{cont}(p_i p_k)$ qui séparent un extremum p_i de chacun de ses voisins de même nature p_j et p_k sont toutes deux plus grandes que τ_{cont} (qui est fixé à *LMT*), alors cet extremum est considéré pertinent. Les extremums de même orientation qui ne rencontrent pas cette condition et qui sont voisins forment une *région bruitée*. Des régions ainsi formées sera retenu un maximum de deux points par région, tel qu'expliqué dans les prochaines lignes.

2.3.2.2 Filtrage des régions bruitées

Le choix ainsi que le nombre de points à retenir dans chacune des régions bruitées sont basés sur des considérations spatiales:

1. La cohérence de la liste de points extremums;
2. La position par rapport à la région des extremums voisins de la région;
3. La position relative des extremums constituant la région.

Le critère de cohérence concerne la composition des listes de points. Pour qu'une liste de points extremums soit considérée cohérente, il faut qu'il y ait alternance entre les extremums d'une même *orientation*. En considérant la liste des extremums d'*orientation* commune, il ne peut y avoir deux extremums successifs de même *type*. Par exemple, suite à un extremum **Xs** (*orientation* X et *type* supérieur), le prochain extremum de même *orientation* doit obligatoirement être de *type contraire*, soit **Xi** (*type* inférieur). Les listes de points après filtrage doivent obligatoirement respecter cette règle.

La position des extremums voisins d'une région détermine le nombre de points à retenir dans cette région. Si la position des voisins permet de considérer la *région bruitée* comme un extremum, un seul point est retenu. Par exemple à la Figure 2.11 les régions bruitées A

et B peuvent être interprétées comme des extremums par rapport aux points voisins des régions sur le contours. Alors un seul point par région doit être retenu.

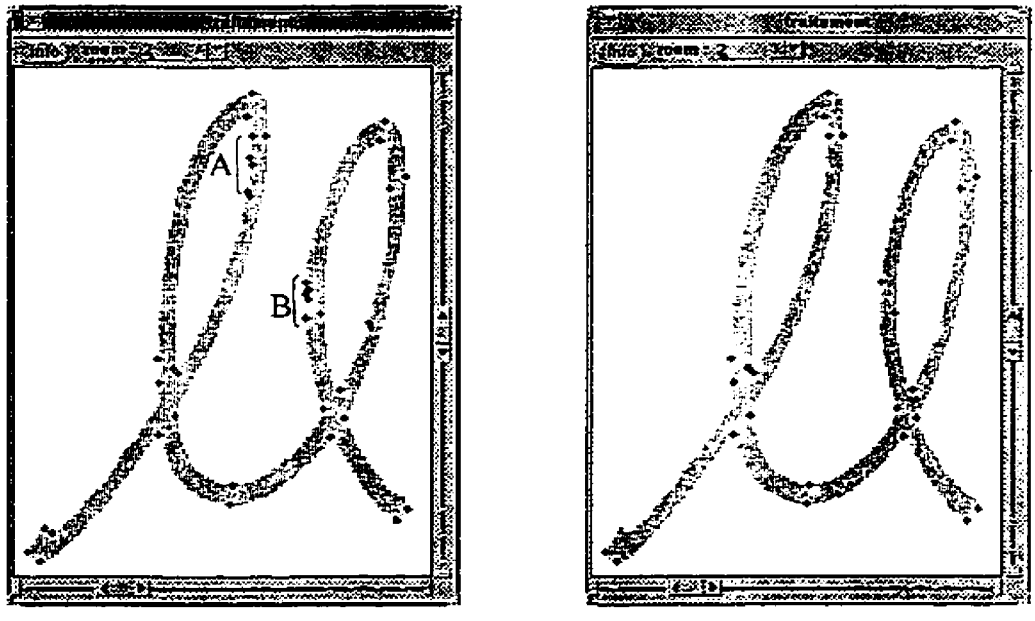


Figure 2.11 Les *régions bruitées* identifiées (A et B) pouvant être interprétées comme un extremum, un seul point par région est retenu.

Le point retenu est un extremum de même type que l'interprétation potentielle de la *région*; il représente également un extremum de même type par rapport aux autres points de la *région bruitée*. Par exemple, si la *région bruitée* peut être interprétée comme un **Yi** par rapport à ses voisins, le point retenu est un point de type **Yi**, et si plus d'un point **Yi** sont présents dans la *région bruitée*, le choix se porte sur celui qui représente un extremum **Yi** par rapport à tous les autres points appartenant à la *région bruitée*.

Dans le cas où la région à l'étude ne peut être interprétée comme un extremum par rapport aux extremums voisins, le nombre de points à retenir est 2 ou 0 selon la position relative des candidats potentiels. Une telle situation survient lorsque les voisins de la région sont des extremums contraires (un inférieur et un supérieur) comme c'est le cas pour les régions bruitées de la Figure 2.12. Pour que la liste des extremums reste cohérente, deux possibilités s'offrent: éliminer tous les extremums de la région bruitée ou bien en retenir deux qui soient de type contraire. Cette dernière possibilité correspond à la situation de la *zone bruitée* A [Figure 2.12] Les candidats potentiels sont les deux extremums contraires qui représentent des extremums par rapport aux autres points de la région bruitée. Ces deux points sont retenus seulement si leur ajout à la liste des extremums pertinents conserve la cohérence de la liste.

Une fois toutes les régions bruitées traitées, il ne reste dans les listes de points extremums P_{ext} que des extremums pertinents qui vont servir à la formation des zones extremums. C'est le sujet dont traite la prochaine section.

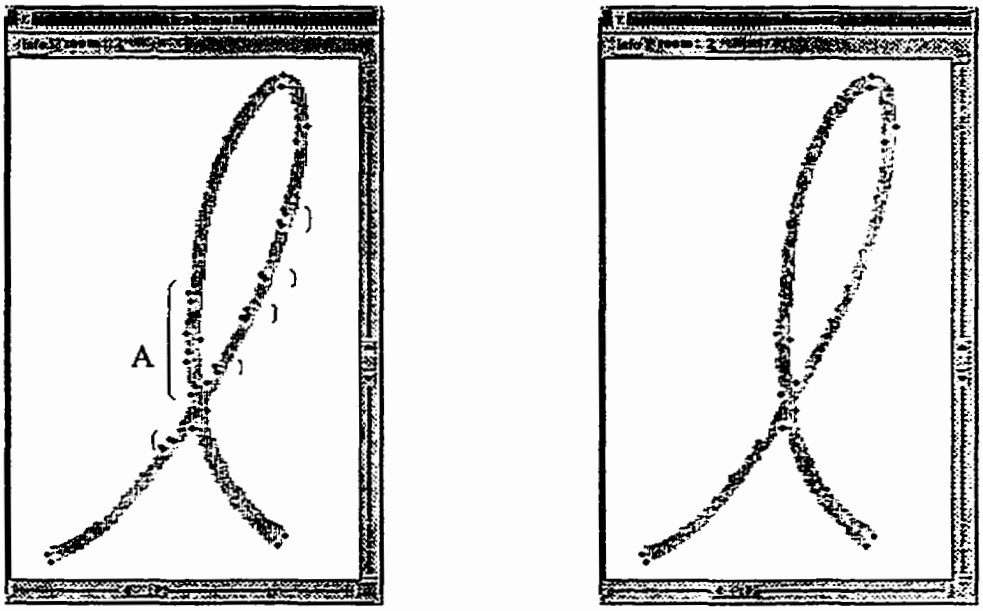


Figure 2.12 Exemples de régions bruitées ne pouvant être interprétées comme un extremum. Pour la région A deux points sont retenus. Pour toutes les autres, aucun point n'est retenu.

2.3.3 Formation des zones extremums

Une zone extremum est formée par deux points extremums rencontrant certains critères particuliers. Puisqu'une zone extremum doit définir une portion de tracé qui représente un extremum de la signature, le premier critère concerne l'orientation et le type des points: pour que la zone puisse être identifiée comme un extremum, il faut que les points la constituant soient des extremums de *même type*. De cette façon la zone peut être identifiée comme étant un extremum de ce type, les quatre types possibles étant **Xi**, **Xs**, **Yi** et **Ys**.

En plus d'être de même type, les points doivent être complémentaires: les constituants de la zone sont ainsi positionnés de part et d'autre du tracé. Cette restriction supplémentaire évite l'association d'extremums dont la présence résulte du croisement de traits ou d'un artefact du tracé.

Il est possible que, malgré ces conditions, pour un extremum donné, il y ait plus d'un candidat possible à la formation d'une zone. La décision finale est prise sur la base d'une métrique développée par l'auteur: l'*indice de visibilité* $v(p_i,p_j)$.

Décrit en détail à l'Annexe D, l'*indice de visibilité* $v(p_i,p_j)$ doit son nom à l'analogie faite avec des observateurs positionnés chacun à un point donné (p_i et p_j) et regardant dans des directions distinctes (φ_i et φ_j). La direction correspond ici à un vecteur perpendiculaire au contour où se situe le point. La valeur de l'indice est fonction à la fois de la distance qui sépare les points et des différences angulaires entre chacun des vecteurs perpendiculaires et la droite reliant les deux points p_i et p_j. La distance maximum d_{max} pour laquelle la valeur de cet indice est non nulle a été déterminée de façon théorique [Annexe E].

Donc, lorsqu'un choix doit être fait parmi les points caractéristiques, l'*indice de visibilité* permet de déterminer un ordre parmi les candidats. Les points qui présentent le plus grand *indice de visibilité* sont alors retenus. Dans le cas présent, seulement deux points doivent être retenus.

La zone peut alors être formée. Chaque zone est identifiée par un type, qui est le même que celui des points extremums la constituant, ainsi que par une coordonnée cartésienne déterminée par la moyenne des coordonnées des points extremums de la zone à l'étude. La Figure 2.13 montre, à l'aide de cercles, les zones extremums formées pour l'image de la Figure 2.2.

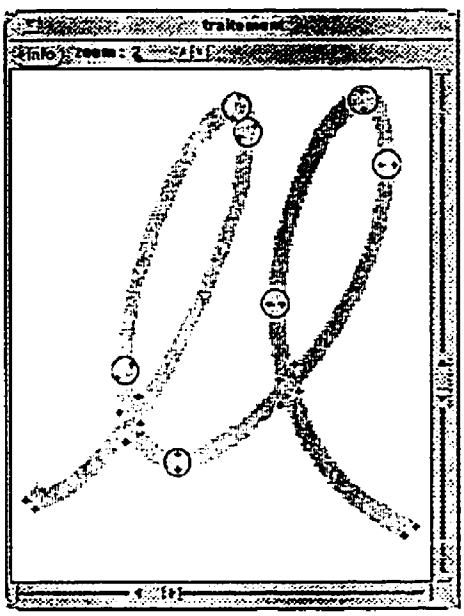


Figure 2.13 Illustration du résultat de la formation des Z_{ext} pour une image donnée.

2.4 Formation des zones de discontinuité

2.4.1 Localisation et identification des points de discontinuité

2.4.1.1 Nature des points identifiés

L'observation d'un tracé manuscrit et de ses contours montre que les profils de ces derniers sont toujours semblables au profil du tracé qu'ils délimitent sauf en des endroits biens précis, qui sont les *extrémités* et les *croisements* de traits. Dans le cas des *extrémités*, le contour décrit idéalement un demi-cercle dont le diamètre est égal à la largeur du trait. Dans les *croisements*, les contours voient leurs orientations changées de façon plus ou moins brusque selon leurs positions et l'angle avec lequel les traits se croisent.

Un espace de représentation qui permet d'observer et d'identifier ces comportements est celui de la courbure. En effet, les changements brusques d'orientation se traduisent par des valeurs de courbure qui se distinguent de la moyenne des valeurs de courbure des contours en général. L'identification de ces valeurs permet de positionner des points dits de discontinuité, et l'analyse des valeurs de courbure au voisinage de ces points détermine la nature de la discontinuité.

L'existence d'une discontinuité étant fonction de son voisinage, l'identification se fait en détectant d'abord des *régions de discontinuité*. De chacune de ces régions est ensuite extrait un seul point qui représente la discontinuité.

Les points de discontinuité sont classés en deux catégories. Il y a, premièrement, les *extrémités* qui se retrouvent au début ou à la fin d'une *composante* [PLAM92a], mais aussi dans ce que nous appellerons une *boucle dégénérée*. Une *boucle dégénérée* est une boucle dont le rayon est trop petit pour laisser apparaître le contour intérieur du tracé. Observée localement, une telle boucle peut être perçue effectivement comme une *extrémité*. Une observation plus globale permet cependant son identification.

La seconde catégorie de points regroupe toutes les discontinuités ne pouvant être considérées comme des *extrémités*. Plus précisément, il s'agit des points dits de *croisement*. Ces points identifient les jonctions de contours appartenant à des portions de tracés différents.

2.4.1.2 Estimation de la courbure des contours

Parmi les différentes méthodes de calcul de la courbure, deux familles peuvent être identifiées. Il y a la famille des *méthodes paramétriques* qui estiment la courbure par une fonction quelconque. Ces méthodes permettent une représentation de plus haut niveau ainsi qu'une réduction de la quantité d'information à manipuler. Il y a aussi les *méthodes différentielles* qui évaluent la courbure pour chaque point de la courbe (contour) en utilisant le voisinage du point pour effectuer une mesure. Avec les *méthodes différentielles*, il n'y a pas de réduction d'information, mais simplement un changement d'espace de représentation.

La méthode utilisée dans le présent système appartient à la seconde famille (*différentielle*). Ce choix s'explique par l'objectif visé qui est de positionner des points sur les contours en fonction de caractéristiques locales du tracé. La méthode retenue est celle proposée par O'Gorman [OGOR88a]. Cette méthode permet d'estimer la variation d'orientation θ_c pour chacun des points des contours. La valeur de θ_c correspond à la différence d'orientation entre deux droites situées de part et d'autre du point concerné. Les paramètres associés à ces droites sont leur longueur (W) et leur espacement (M). L'Annexe B décrit en détail le raisonnement suivi pour le choix de ces paramètres. Mentionnons simplement que M est fixé à 0 pour tous les contours et que W est déterminé à la fois par la valeur de LMT et par la longueur du contour concerné.

Étant donné que la variation d'orientation est calculée à l'aide de droites de longueur constante, cela est équivalent à un échantillonnage à pas constant. La variation d'orientation θ_c reflète donc directement la valeur de la courbure.

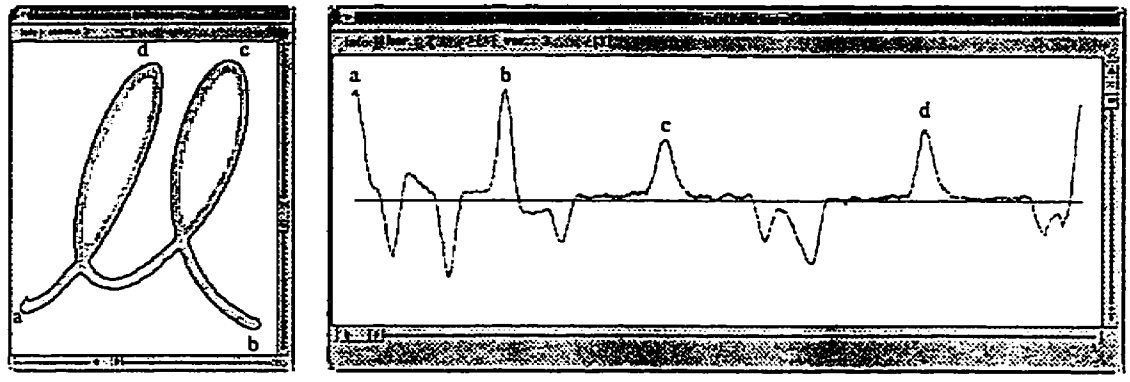


Figure 2.14 Exemple de profil de variation d'orientation calculé avec la méthode de O'Gorman. Puisque le pas d'échantillonnage est constant, ces valeurs reflètent directement la courbure.

Avec cette méthode la valeur de l'angle θ_c pour un point appartenant à une droite est 0°. Plus la courbure est grande, plus la valeur de l'angle s'approche de 180° ou de -180°, selon le sens de la courbure. Étant donné la méthodologie suivie pour l'identification des contours de l'image (section 2.2.3), il est possible d'affirmer que:

1) les points de contour près des *extrémités* présentent des variations d'orientation *positives* ($\theta_c > 0°$);

2) les points de contour près des *croisements* de traits présentent des variations d'orientation *négatives* ($\theta_c < 0°$).

Ces affirmations sont possibles étant donné la consistance avec laquelle sont construites les chaînes de points représentant les contours. Si elles permettent de distinguer un point

se situant près d'une *extrémité* d'un autre près d'un *croisement*, elles ne permettent cependant pas de choisir, parmi tous les points, ceux qui identifient vraiment des *extrémités* ou des *croisements*. Une procédure de filtrage est, de ce fait, nécessaire.

2.4.1.3 Filtrage des profils de courbure

Les points qui identifient les *extrémités* ou les *croisements*, soit les *points de discontinuités*, sont des points qui partagent certaines caractéristiques. L'observation du profil de variation d'orientation de la Figure 2.14 montre que la courbure près d'une discontinuité présente toujours un extremum local. Cependant, la seule valeur de l'angle de variation d'orientation de cet extremum n'est pas suffisante pour identifier correctement une *discontinuité*. Il faut tenir compte de l'amplitude locale de la variation d'orientation du point, en la comparant avec celle des points de son voisinage. Le voisinage considéré - la *région de discontinuité* - est défini à l'aide d'un opérateur morphologique.

L'opérateur retenu est la transformation en chapeau-haut-de-forme (« top-hat transform ») et son complément, la transformation en chapeau-bas-de-forme (« bottom-hat transform »), tel que proposé dans [LEYM89]. Cet opérateur est en fait une combinaison d'opérations de morphologie mathématique. À la base de la morphologie mathématique sont deux opérations complémentaires: l'*érosion* ($\ominus$) et la *dilatation* ($\oplus$) [SERR82]. Dans les deux cas, il s'agit d'observer le voisinage d'un point d'une courbe ou fonction

quelconque et d'en modifier la valeur selon une règle donnée. Le voisinage observé pour l'application de la transformation est défini par un *élément structurant*, dont la taille détermine les limites du voisinage, et la forme modifie les valeurs du voisinage sur lequel il est appliqué. L'élément structurant possède également un point de référence qui correspond au point pour lequel s'applique la transformation.

La règle suivie pour l'*érosion* [Figure 2.15c] consiste à remplacer chaque point de la courbe traitée par la valeur minimum des points de son voisinage. Inversement, la *dilatation* consiste à remplacer chaque point par la valeur maximum des points de son voisinage [Figure 2.15b].

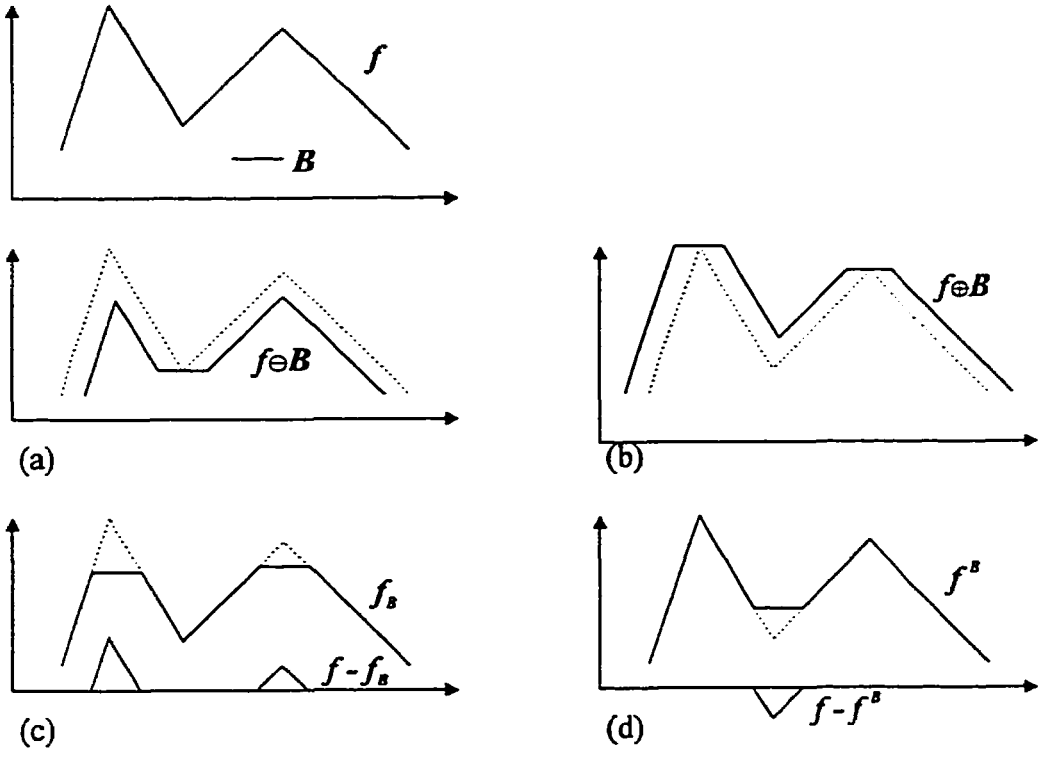


Figure 2.15 Illustration de différentes opérations morphologiques appliquées à une fonction f en utilisant l'*élément structurant* B : a) érosion $f \ominus B$, b) dilatation $f \oplus B$, c) ouverture f_B et chapeau-haut-de-forme $f - f_B$, d) fermeture f^B et chapeau-bas-de-forme $f - f^B$.

La combinaison de ces deux opérations de base génère de nouvelles fonctions. Parmi celles-ci, il y a l'*ouverture*, qui consiste à effectuer une dilatation sur une fonction préalablement érodée [Figure 2.15c]. Inversement, l'érosion d'une fonction dilatée porte le nom de *fermeture* [Figure 2.15d].

La transformation en chapeau-haut-de-forme est le résultat de la différence entre une fonction f et son ouverture f_B [Figure 2.15c]. Une telle opération permet d'isoler tous les

sommets d'une fonction. La largeur de chacun des sommets identifiés est au maximum égale à la largeur de l'élément structurant utilisé. L'amplitude dépend de la géométrie de la fonction. La transformation en chapeau-bas-de-forme, soit la différence entre une fonction f et sa fermeture f^B, permet d'isoler les vallées d'une fonction [Figure 2.15d]. Ces vallées répondent aux mêmes contraintes de largeur et d'amplitude.

Ces deux transformations appliquées aux profils de variation d'orientation permettent l'identification des régions du tracé qui peuvent potentiellement contenir des *points de discontinuité*. Les sommets de la transformation chapeau-haut-de-forme représentent les régions *extrémités* et les vallées de la transformation en chapeau-bas-de-forme, les *croisements*.

L'élément structurant utilisé est le même pour les deux transformations. Il s'agit d'un élément symétrique linéaire et horizontal, dont chacun des éléments vaut 1. La largeur de cet élément est déterminée de la même façon que le paramètre W utilisé pour le calcul de la courbure (section 2.4.1.2).

Évidemment, les régions identifiées ne représentent pas toutes des discontinuités. Une étape de filtrage est donc nécessaire.

2.4.2 Validation des régions de discontinuité

Le premier critère de validation repose sur les valeurs de courbure. Comme mentionné précédemment, les *extrémités* sont caractérisées par des valeurs de θ_c supérieures à 0°. Donc, tous les sommets dont la variation d'orientation est plus petite que 0° peuvent être rejetés. Il en est de même pour les vallées dont les variations d'orientation minimums sont plus grandes que 0°, puisque les vallées représentent les positions potentielles de points de *croisements*.

Cependant, étant donné le bruit présent autour de 0°, il est préférable de définir une plage de valeurs proscrites. La variation d'orientation étant estimée de façon à ce que l'angle θ_c atteigne un certain minimum près des *extrémités* [Annexe B], il est permis de hausser la valeur limite de 0° à 90° pour les sommets sans risque de commettre une faute. Pour les vallées, des observations répétées montrent que celles dont la variation d'orientation maximum est plus grande que -20° ne correspondent pas à des régions de discontinuité.

Après ce premier tri, les régions subsistantes sont triées à nouveau, mais cette fois, sur la base de l'amplitude de la variation d'orientation dans les régions de *discontinuité*. La taille de l'élément structurant utilisé est choisie de telle sorte que l'amplitude de la courbure dans les régions de discontinuité atteigne un certain niveau. Le choix du seuil d'amplitude $\tau_{\Delta\theta}$ se base sur le comportement de la courbure pour un croisement idéal. En fonction de

l'étude présentée à l'Annexe C, les régions de discontinuité dont l'amplitude est inférieure à $\tau_{\Delta\theta}$=15° sont rejetées.

Le dernier critère de validation est fonction de la position de la région de discontinuité sur le tracé. Si dans une région de discontinuité se trouve un point extremum qui appartient à une zone extremum identifiée, cette région est éliminée. Il est en effet inutile de positionner un point de discontinuité dans une portion de contour qui fait déjà l'objet d'une interprétation. Par contre, si la région renferme un extremum et que ce dernier n'appartient pas à une zone, alors c'est l'extremum qui est éliminé.

Les régions restantes sont considérées comme situées près des discontinuités du tracé. Pour chaque région non étiquetée, le point dont la variation d'orientation représente un extremum local est retenu. C'est ce point qui est considéré comme point de discontinuité. Il est identifié comme un point *extrémité* si la valeur de θ_c associée est supérieure à 0° et comme un point de croisement dans le cas contraire. Les points ainsi identifiés appartiennent à P_{disc}, la liste des points de discontinuités. La prochaine figure [Figure 2.16] illustre les discontinuités et les régions de discontinuité correspondantes identifiées dans l'image de la Figure 2.2a.

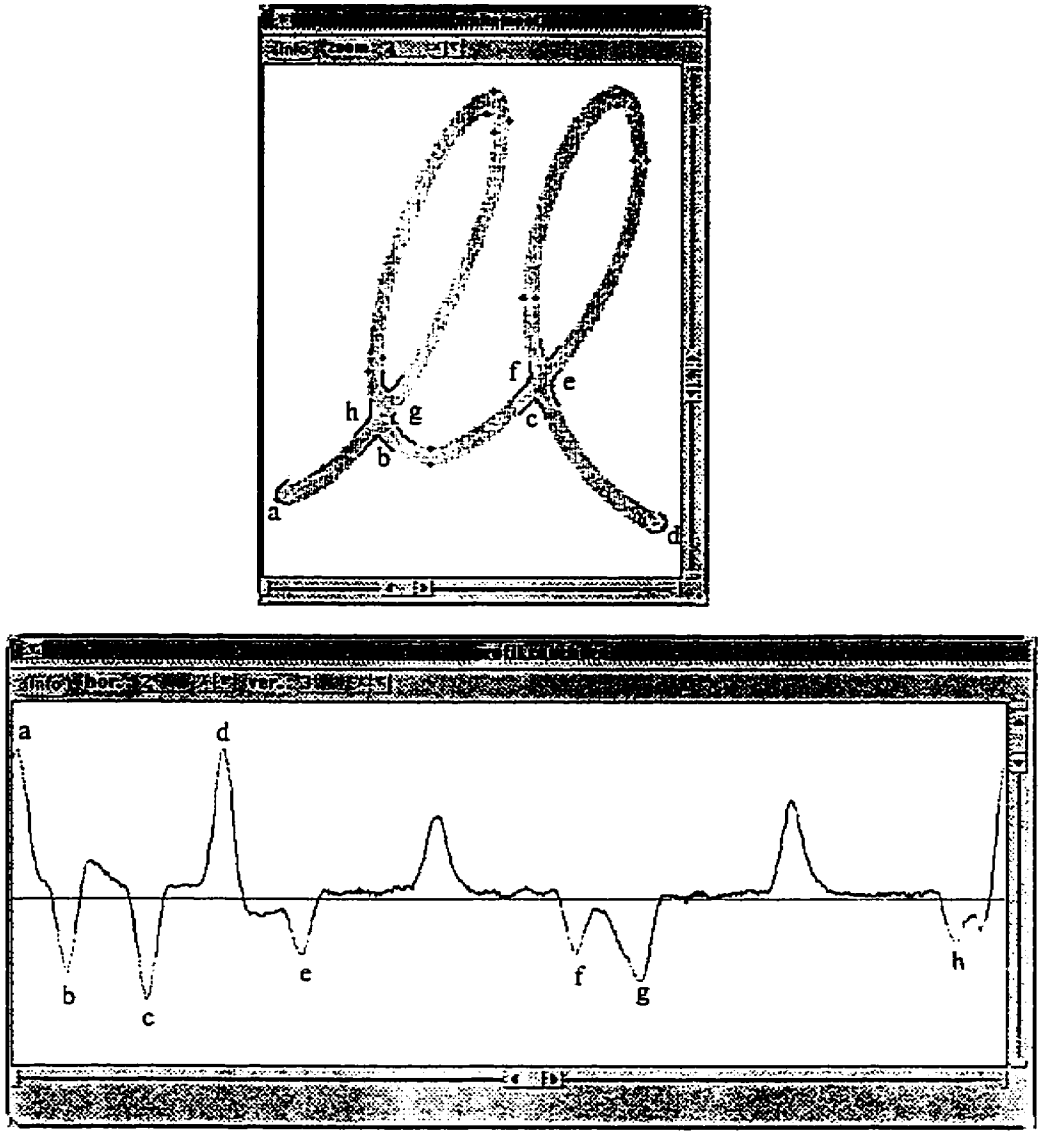


Figure 2.16 a) Exemple de *points de discontinuité* identifiés et des *régions de discontinuité* correspondantes. b) Profil de courbure correspondant au contour extérieur de l'objet présent en (a). Les portions de profil identifiées correspondent à des *régions de discontinuité*.

2.4.3 Formation des zones de discontinuité

2.4.3.1 Critères de formation

Les *zones de discontinuité* sont formées à partir des points de discontinuité et de tous les extremums qui ne forment pas une zone extremum. Contrairement aux zones extremums, les *zones de discontinuité* peuvent être constituées par un nombre de points caractéristiques variables. Il ne s'agit donc pas simplement d'identifier deux points qui, parmi un ensemble de candidats, maximisent un critère donné. Il faut identifier des ensembles de points dont la cardinalité n'est pas connue a priori, et dont chaque élément peut être relié à au moins un autre du même ensemble, en fonction d'un critère donné.

Le critère à rencontrer ici est un *indice de visibilité* $v(p_i,p_j)$ [Annexe D] non nul. Tel que mentionné précédemment, cet indice est basé sur la distance qui sépare deux points de contour, ainsi que sur les angles entre la droite reliant les points et la perpendiculaire par rapport au contour à chacun de ces points. La valeur de cet indice est automatiquement nulle si la droite qui relie deux points de discontinuité n'est pas totalement incluse dans le tracé. Le paramètre d_{max}, correspondant à la distance maximum pour laquelle $v(p_i,p_j)$ est différent de zéro, est fixé à 20 *LMT*. Ce choix, justifié à l'Annexe E, est fait en considérant les points de discontinuité d'un croisement à angle aigu.

Chaque ensemble de points de discontinuité est formé par des points qui ont chacun au moins un lien avec un autre point de l'ensemble. Le corollaire de ceci est qu'aucun point de discontinuité n'a de lien avec un point d'un autre ensemble de points localisés ailleurs dans l'image. Ainsi, chaque zone de discontinuité est formée à partir des points d'un seul ensemble, un ensemble pouvant éventuellement servir à former plus d'une zone. La formation des ensembles permet ainsi de réduire l'espace de recherche pour la méthode de sélection des points de discontinuité pouvant être associés à une zone.

2.4.3.2 Méthode de sélection des points

À partir des ensembles identifiés, il faut choisir les points qui forment une même zone. Étant donné les critères de formation des ensembles, pour que des points forment une zone de discontinuité, il doit exister entre chaque paire de points un indice de visibilité non nul. Autrement dit, une zone est formée par des points qui représentent une *clique maximale* de l'ensemble concerné.

Dans un graphe, une *clique* représente un sous-graphe dans lequel chaque noeud est directement relié à chacun des autres noeuds de la clique. Une *clique* est dite *maximale* lorsqu'il est impossible d'y ajouter un autre noeud [BOLL79]. Les ensembles de points peuvent être représentés sous la forme d'un graphe: chaque point représente un noeud, et un indice de visibilité non nul forme un arc entre les points concernés.

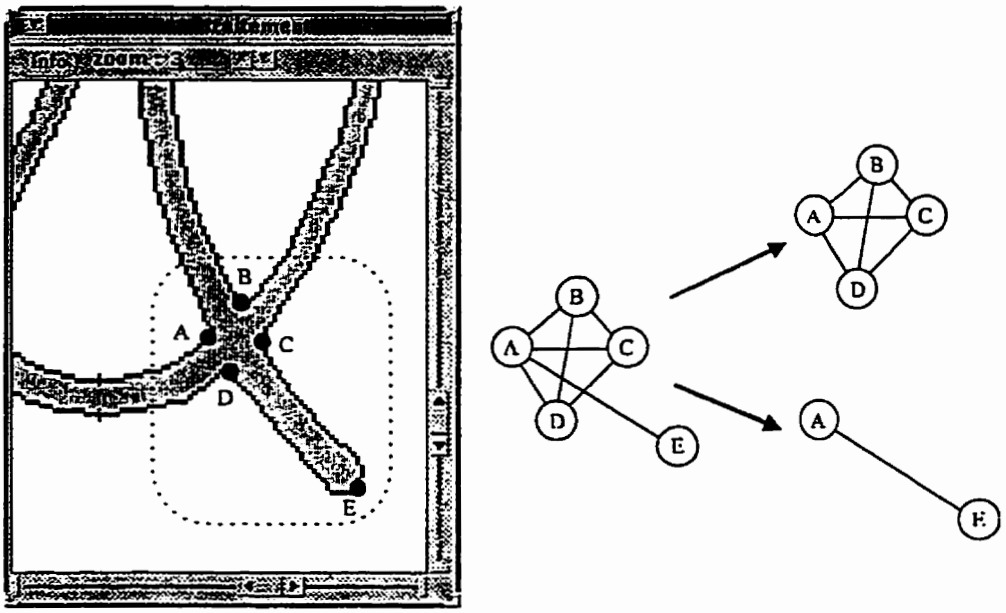


Figure 2.17 Illustration du concept de clique maximale. Dans cet ensemble de 5 points, 2 cliques maximales sont identifiées.

Comme le montre la Figure 2.17, un ensemble de points de discontinuité peut inclure plus d'une clique maximale. Il faut donc choisir, parmi toutes les cliques maximales, celles qui représentent vraiment des zones de discontinuité. Ce choix s'effectue dans le respect des règles suivantes:

1) Les cliques retenues doivent être *exclusives*. Une clique est exclusive lorsqu'elle ne partage pas d'élément avec une autre;

2) Les cliques retenues ne doivent pas se *chevaucher*. Les droites reliant les points d'une clique ne peuvent croiser celles d'une autre clique.

L'exemple de la Figure 2.17 illustre bien ces deux règles. Les points A,B,C,D et E forment un ensemble contenant deux cliques maximales (les sous-ensembles A,B,C,D et A,E). Ces deux cliques ne respectent aucune des deux règles puisqu'elles partagent un élément commun (le point A) et que la droite AE croise les droites BD et CD. Les deux cliques maximales ne pouvant donc pas être toutes deux choisies, un choix doit être fait.

La sélection des cliques se fait dans un ordre bien précis. Les cliques sont analysées en commençant par les plus grandes, c'est-à-dire celles dont la cardinalité est la plus élevée. Dans le cas de cliques de même taille, c'est l'indice de visibilité maximum de chaque clique qui détermine l'ordre: elles sont classées par ordre décroissant de la valeur de cet indice.

Donc, par défaut, la plus grande clique maximale avec le plus grand indice de visibilité est retenue. Les autres cliques sont analysées une à une. Une clique est retenue seulement si, en l'associant avec chacune des cliques déjà retenues, aucune des deux règles précédemment énoncées n'est enfreinte. Par exemple, dans la Figure 2.17, seule la plus grande clique est retenue.

Chaque clique retenue forme une zone de discontinuité. Il est possible que certains points caractéristiques ne soient associés à aucune zone. Pour les fins du traitement ces points sont également considérés comme des zones. Dans la Figure 2.17, par exemple, le point E

n'est associé à aucune clique maximale retenue: il est donc considéré comme appartenant à une zone distincte.

2.4.3.3 Identification des zones de discontinuité et des boucles dégénérées

La composition des zones de discontinuité varie en nombre et en type de points. Parmi toutes les compositions possibles, certaines contenant des points *extrémité* peuvent être identifiées de façon particulière.

Premièrement, si une zone n'est formée que d'un seul point et que ce point est une *extrémité*, alors cette zone est identifiée comme étant une *zone extrémité* [Figure 2.17]. Ce type de zone est le seul qui se compose d'un seul point et pouvant faire l'objet d'une interprétation.

Une zone formée d'une *extrémité* et d'un point de croisement est identifiée comme une *boucle dégénérée*. Les zones caractéristiques de la Figure 2.18 pointées par une flèche sont autant d'exemples de *boucles dégénérées*. L'observation de ce type de boucles permet de constater qu'en effet, un seul point de croisement est généralement détecté et qu'il fait face au point *extrémité*.

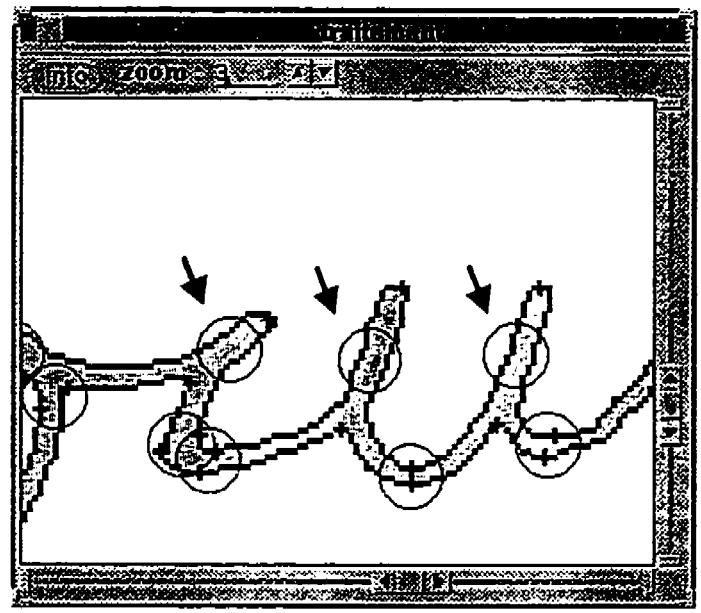


Figure 2.18 Les zones identifiées par des flèches représentent des boucles dégénérées.

Toutes les autres zones sont simplement identifiées comme des zones de discontinuité, et c'est le contexte qui donnera la bonne interprétation à ces zones. Les zones qui sont formées d'un seul point autre qu'un point *extrémité* sont identifiées comme des zones de *point solitaire*. Puisque seules les *zones extrémités* peuvent être composées d'un seul point, celles étiquetées *point solitaire* font l'objet d'une attention particulière dans le graphe de relations.

2.5 Formation du graphe de relations

2.5.1 Composition du graphe de relations

Le graphe de relations G_{rel} permet de structurer l'information extraite de l'image traitée. Il doit mettre en relation les différentes zones identifiées et ce, en s'inspirant du contexte dans lequel elles se trouvent.

Les noeuds du graphe de relations G_{rel} sont donc des zones préalablement identifiées. Ce sont les quatre types de zones extremums Z_{ext} et les zones de discontinuités Z_{disc}. Parmi ces dernières, il y a, entre autres, les *points solitaires*, qui représentent des zones non complètes et qui doivent être complétées. L'ajout de points à ces zones se fait à l'aide du *graphe de relations* G_{rel}.

Pour qu'il y ait un arc entre deux noeuds, il faut qu'il existe un lien entre les zones qu'ils représentent. Dans le présent cas, le lien est synonyme de contour: si une portion de contour permet de relier deux zones directement (sans passer par une tierce zone), alors un arc est formé entre les noeuds correspondants. Comme le nombre de portions de contours pouvant satisfaire cette condition entre deux zones varie, le lien doit être qualifié en fonction de ce nombre.

2.5.2 Identification des arcs

L'identification des arcs est fonction du nombre de liens existant entre deux zones. La première distinction à faire parmi les arcs dépend de la parité des liens. Les liens représentent des portions de contours et les zones, des portions de tracé, et entre deux portions d'un même trait, il y a généralement deux segments de contours. Donc, dans un cas idéal, il y a toujours un nombre *pair* de liens entre deux zones. Le nombre de liens est *impair* dans les cas où la binarisation est incorrecte, ou lorsque les zones ne représentent pas correctement les portions d'intérêt du tracé, par exemple les zones de *point seul*. Donc un arc est dit *complet* lorsque le nombre de liens est pair et *incomplet* sinon.

Une distinction est aussi établie entre les arcs complets, en fonction du nombre de paires de liens. Une paire de liens représente généralement une portion de tracé. Donc, le nombre de paires de liens traduit le nombre de portions de traits qui existent entre deux zones. Un arc qui représente deux liens est donc un arc complet *simple*, un arc qui représente 4 liens est dit complet *double* [Figure 2.19] et ainsi de suite. Il est plutôt rare de rencontrer des arcs qui représentent plus de 4 liens, mais ce n'est pas impossible.

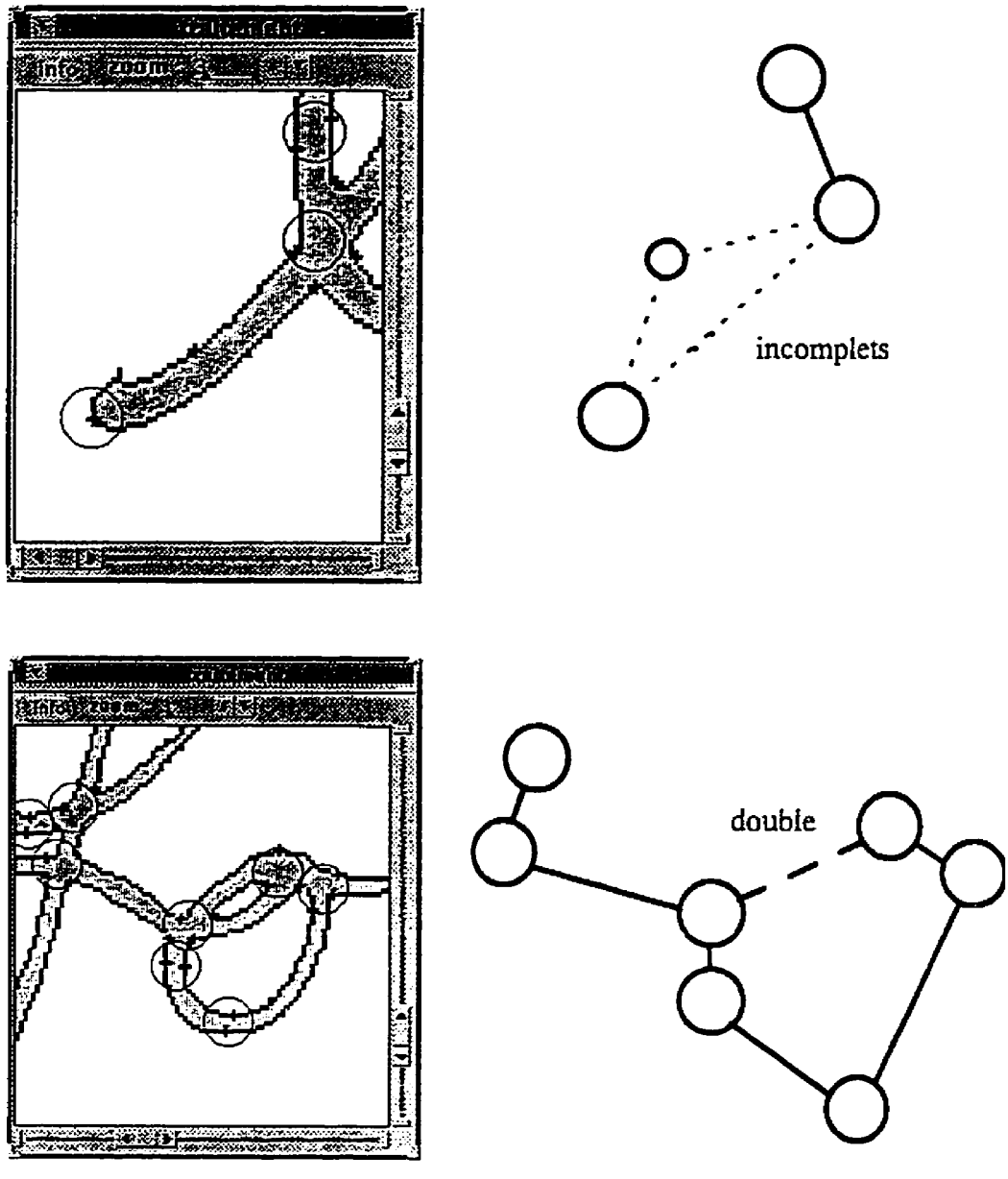
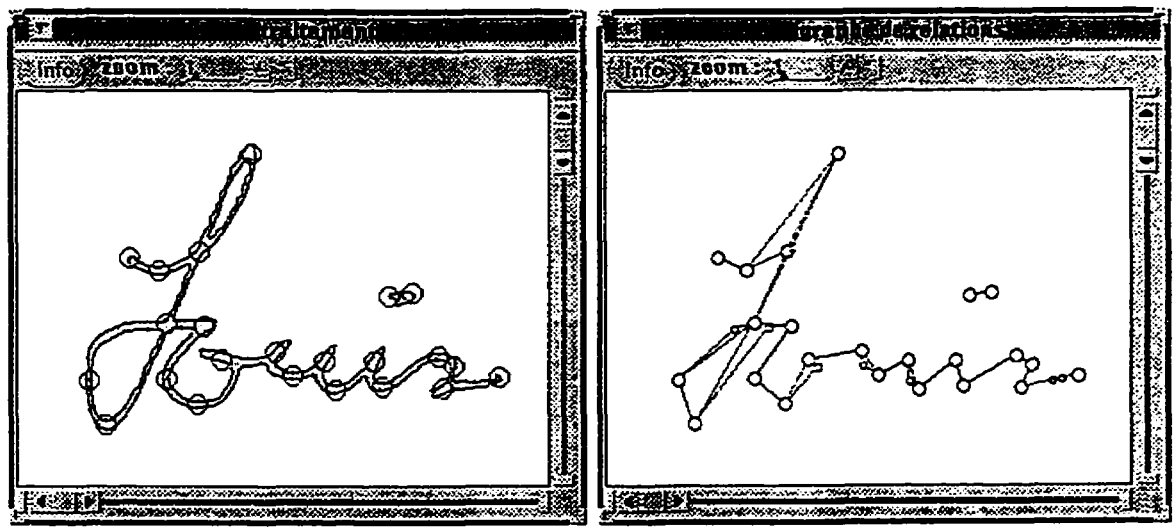


Figure 2.19 Exemple montrant les différents types d'arc. Tous les arcs qui ne sont pas *complets* (traits pleins), peuvent être soit *incomplets* (pointillés) ou *multiples* (tirets).

Idéalement, le G_{rel} doit être constitué uniquement d'arcs complets. Mais comme des erreurs d'interprétation sont toujours possibles, le résultat n'est pas toujours parfait

comme en témoigne la Figure 2.20. Il faut donc procéder à des modifications dans le graphe de relations.

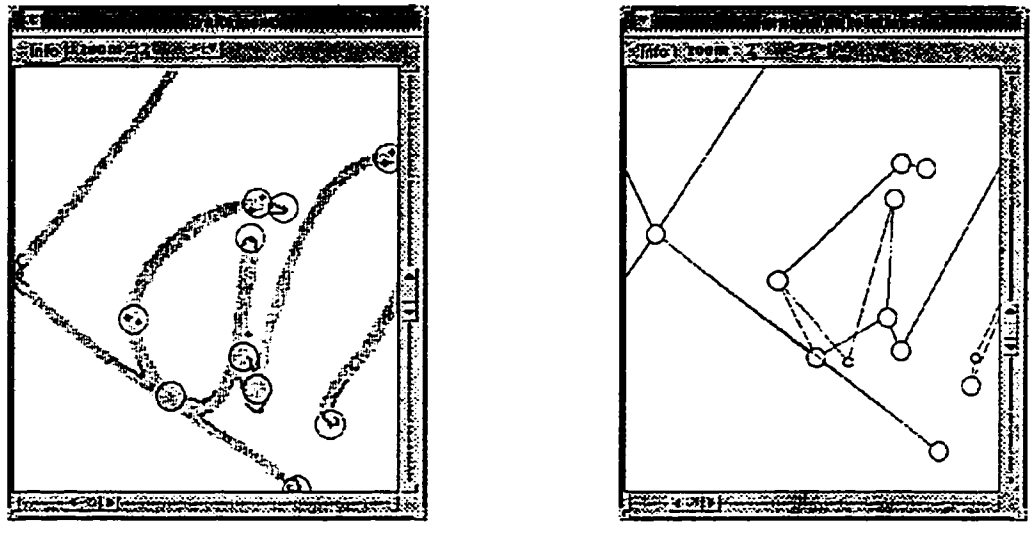


(a) (b)

Figure 2.20 Exemple de G_{rel} montrant certaines erreurs d'interprétation (b) et l'image binaire correspondante incluant les différents points caractéristiques (a).

2.5.3 Édition du graphe de relations

L'édition du graphe de relations G_{rel} a comme objectif premier l'élimination de tous les arcs incomplets. La seul façon de changer le statut d'un arc est de modifier une des deux zones qu'il relie, de façon à rendre le nombre de liens pair entre ces zones. La Figure 2.21 montre une portion de tracé affectée par la phase d'édition.



(a) (b)

Figure 2.21 Représentation d'une portion de tracé devant faire l'objet d'une édition. La portion présentée est modifiée par l'application des deux règles d'édition.

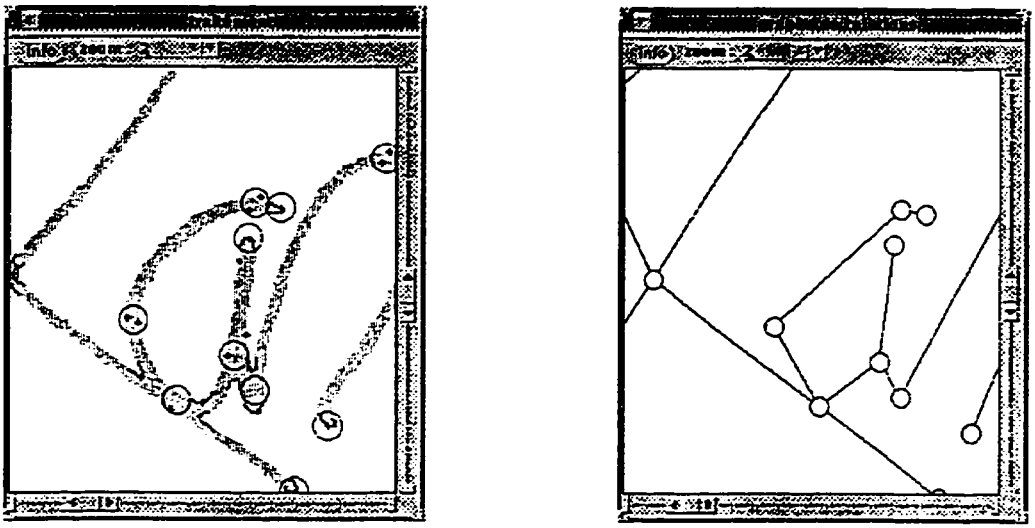
Une première possibilité consiste à fusionner des zones qui présentent des arcs incomplets. Si ces arcs incomplets relient une tierce zone, alors l'arc entre cette zone et le résultat de la fusion est nécessairement complet. Cette situation implique généralement une zone de type *point seul* et une zone de discontinuité. Pour que deux zones puissent être fusionnées, elle doivent satisfaire aux conditions suivantes:

1) Les zones doivent être *reliables*. Les centres des zones doivent pouvoir être reliés par une droite complètement incluse dans le tracé;

2) Les zones doivent être à une distance inférieure à $4 \times LMT$. Cette valeur étant motivée par des observations multiples.

Une autre situation causant la présence d'arcs incomplets résulte de l'inexistence d'un point caractéristique. Si les mécanismes de détection oublient un de ces points, il en résulte généralement une zone incomplète. Le point manquant a pour effet de créer des liens incomplets entre trois zones: la zone à laquelle il manque un point et deux des zones voisines. La solution à ce problème est de trouver, sur le lien incomplet entre les zones voisines, le point manquant. Le point choisi est celui qui minimise la distance par rapport au centre de la zone à compléter et qui peut être relié au centre de la zone par une droite totalement incluse dans le tracé.

Après avoir tenté d'ajouter des points à toutes les zones possibles, il se peut que certaines zones de *points solitaires* subsistent toujours. Tout ayant été fait pour les modifier, il ne reste qu'à les éliminer. Ceci contribue à l'élimination d'arcs incomplets. Si des liens incomplets subsistent toujours, les deux traitements précédents sont repris et ce, tant qu'ils apportent des changements à la configuration du graphe de relations G_{rel}.

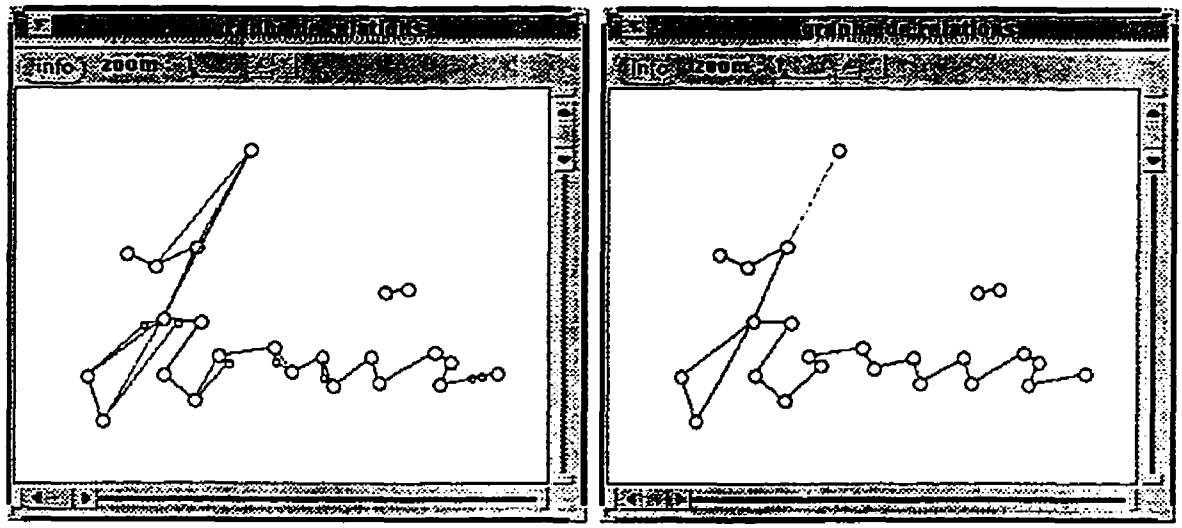


(a) (b)

Figure 2.22 Résultat de l'application des règles d'édition sur la portion de tracé présenté à la Figure 2.21. Dans cet exemple, il y a eu fusionnement de zones et ajout d'un point.

Dans le graphe résultant de la procédure d'édition, il est possible de retrouver des zones dont la position par rapport aux voisins ne représente pas un extremum. Ceci se produit généralement avec des zones de discontinuité formées de deux points et qui ont seulement deux voisins. De telles zones devraient représenter un extremum. Si tel n'est pas le cas, il est inutile de les conserver: elles sont éliminées de même que les points caractéristiques les constituant.

Le graphe résultant de toutes ces modifications est considéré comme le graphe de relations G_{rel} final [Figure 2.23]. C'est en utilisant ce graphe que la procédure de mise en correspondance est appliquée [Chapitre 4].



(a) (b)

Figure 2.23 G_{rel} de la Figure 2.20 avant (a) et après (b) édition.

3. TRAITEMENT DE L'INFORMATION DYNAMIQUE

3.1 Description générale

L'objectif du présent système est d'inférer la séquence dans une *image* de signature manuscrite en utilisant comme connaissance a priori la séquence exprimée par un *signal dynamique*. Le fait de comparer des signaux de nature différente constitue une des caractéristiques particulières du système proposé.

Pour que des signaux de nature différente puissent être comparés, ils doivent être exprimés dans un espace de représentation commun. Tout comme de l'*image* est extrait un graphe de relations G_{rel} [chapitre 2], il s'agit ici de former un *graphe de représentation* G_{rep}, représentant le *signal dynamique*. Les différentes étapes nécessaires à la transformation de l'information dynamique en un graphe de représentation sont résumées par le *DFD* de la Figure 3.1.

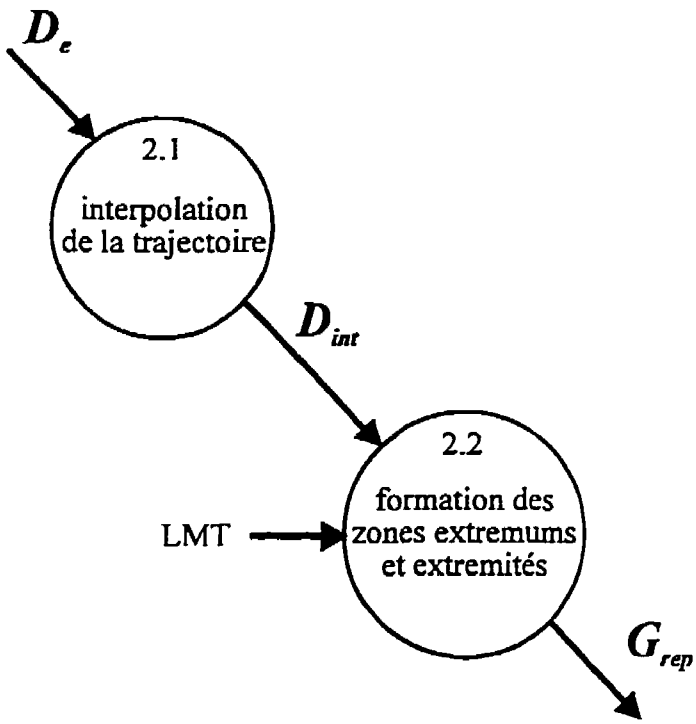


Figure 3.1 *DFD* montrant les différentes étapes de traitement de l'information dynamique.

L'information dynamique est soumise au système sous la forme d'une *liste de points échantillonnés* D_e. Ces points représentent les échantillons de position captés par une tablette à digitaliser. Un exemple de points résultant de cet échantillonnage est donné à la Figure 3.2.

Puisque les échantillons sont captés à une fréquence de 100 *Hz*, les points échantillons ne forment pas une liste continue de points. La première étape du traitement consiste donc à

interpoler une trajectoire entre chacun de ces points, de manière à créer une liste continue: la *liste de points interpolés* D_{int} [Figure 3.3].

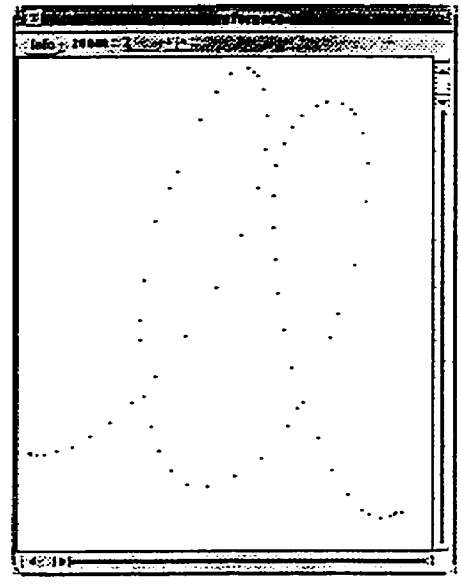


Figure 3.2 Exemple de points échantillonnés.

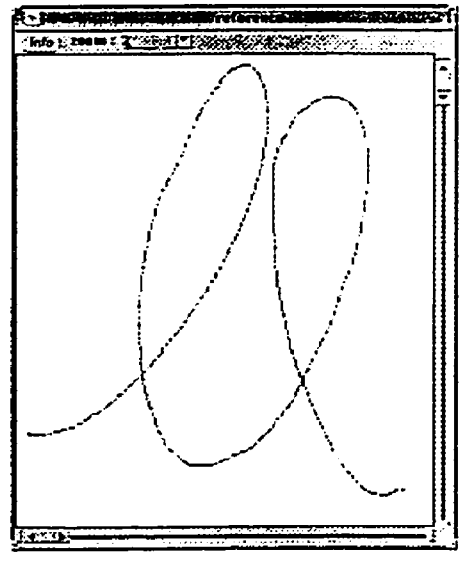


Figure 3.3 Résultat de l'interpolation entre les échantillons présentés à la figure précédente.

Sur la séquence interpolée D_{int} sont ensuite identifiés des *points caractéristiques* P_{car} [Figure 3.4]. Le vocable *caractéristique* regroupe les *extremums* ainsi que les *extrémités de trait*. Les *zones caractéristiques* Z_{car} formées à partir de ces points représentent les noeuds du *graphe de représentation* G_{rep} [Figure 3.5]. Ce graphe constitue l'aboutissement du traitement de l'information dynamique.

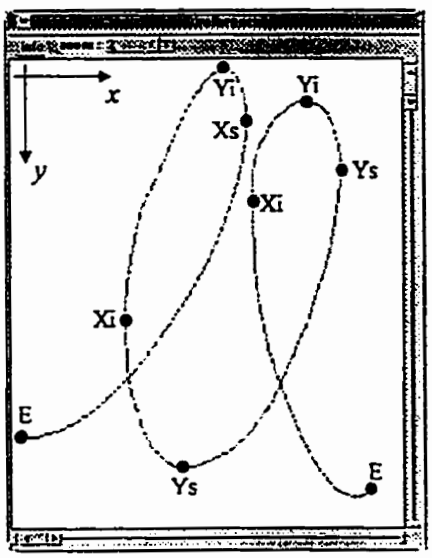


Figure 3.4 Points caractéristiques P_{car} identifiés sur la trajectoire interpolée. Il s'agit de points *extremums* et d'*extrémités de trait*.

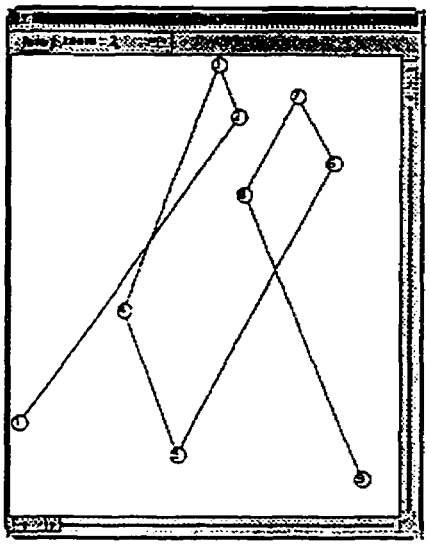


Figure 3.5 Graphe de représentation G_{rep} formé à partir des points caractéristiques P_{car} identifiés.

3.2 Interpolation de la trajectoire

3.2.1 Nature des données

L'information dite statique est le résultat de l'échantillonnage effectué par la tablette instrumentée lors de la production d'une signature. Le résultat de cet échantillonnage est une liste de coordonnées (x,y) représentant la position du stylo à des instants précis. Les signaux utilisés sont le résultat d'un échantillonnage à une fréquence de 100 *Hz* avec une résolution de 1000 *ppi*.

Étant donné la différence de résolution et de référentiel entre les instruments d'acquisition utilisés pour les images et les signaux dynamiques, il est nécessaire d'uniformiser les données. Le changement de référentiel doit procéder à une mise à l'échelle, une translation

de l'origine du système d'axe et une inversion de l'axe y [Figure 3.6]. Étant donné qu'une inversion d'axe doit être faite, il est nécessaire de définir une fonction de transformation pour chacun des axes du système. Ces équations sont les suivantes:

$$x' = \frac{R_s}{R_d}(x - x_{\min}) \tag{1}$$

$$y' = \frac{R_s}{R_d}(y_{\max} - y) \tag{2}$$

Dans ces équations, les termes R_s et R_d correspondent aux résolutions des systèmes d'acquisition des images et des signaux dynamiques respectivement (R_s=300, R_d=1000). Le terme $x_{\min}$ correspond à la coordonnée minimum en x des points échantillons D_e [Figure 3.6a]. De même, le terme $y_{\max}$ correspond à la coordonnée maximum en y des points échantillons D_e.

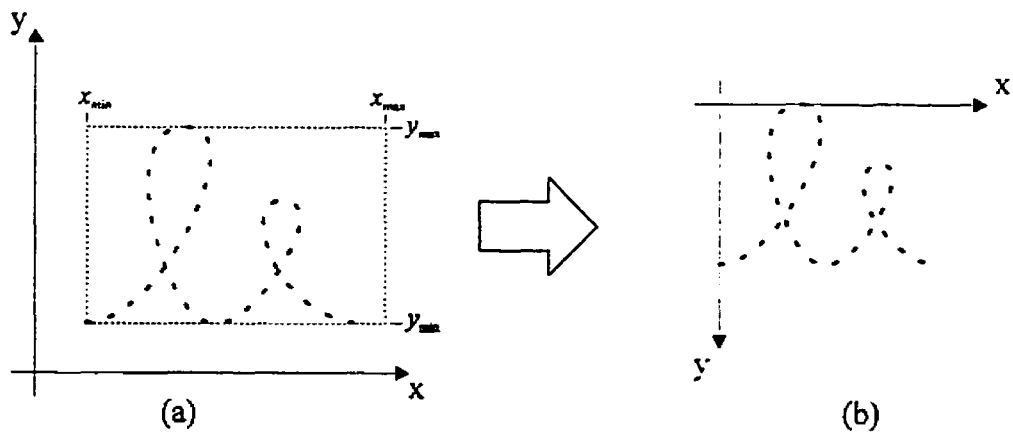


Figure 3.6 Illustration de la procédure de changement de référentiel pour les données des signaux dynamiques.

En plus de la position, la tablette fournit une donnée indiquant la pression exercée sur la pointe du stylo. Certains échantillons peuvent avoir un indicateur de pression nulle: ce sont des échantillons représentant un déplacement du stylo sans qu'il y ait contact avec la tablette, mais assez près de la surface pour qu'ils soient détectés. Cette donnée permet donc de distinguer les échantillons qui correspondent à des productions de traits de ceux qui correspondent à des déplacements entre deux traits. Par exemple, à la Figure 3.7, les échantillons compris entre les points A et B correspondent à des échantillons représentant un déplacement aérien du stylo.

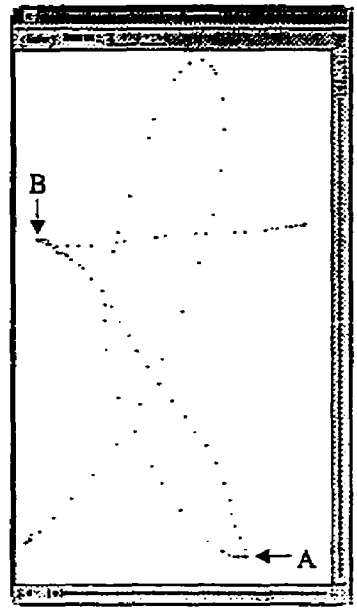


Figure 3.7 Échantillons captés par la tablette à digitaliser lors de la production de l'allographe *t*.

3.2.2 Filtrage des données

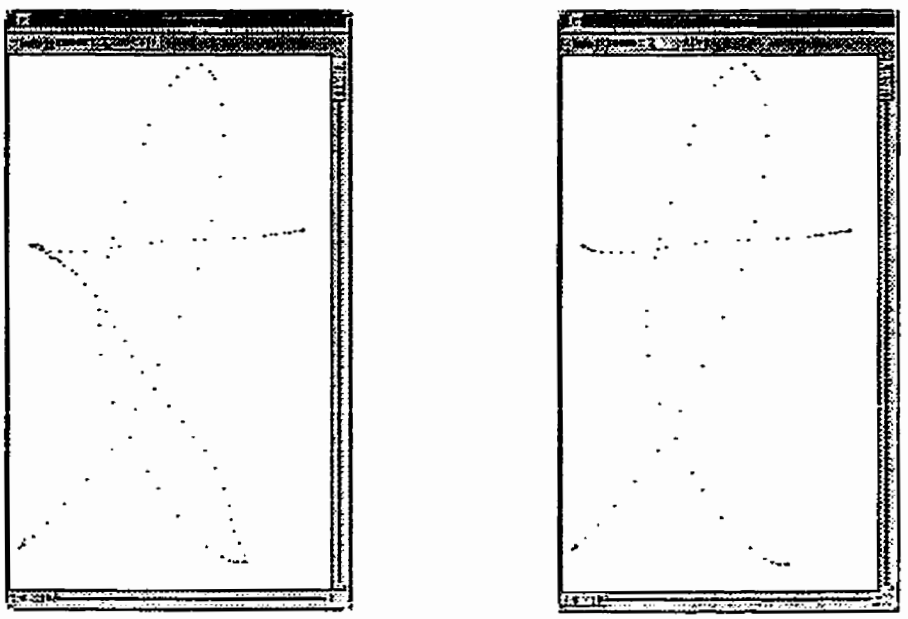
Étant donné que seul le tracé présente de l'intérêt ici, le premier rôle du filtrage consiste à éliminer de la liste des échantillons D_e tous ceux correspondant à des déplacements sans contact avec la tablette. Ces points correspondent aux échantillons dont l'indicateur de pression est nul. La liste résultante ne contient donc que les points représentant la production du tracé.

La présence d'échantillons dont la pression est nulle indique que le tracé à l'étude est composé de plusieurs *composantes* [PLAM92a]. Il faut donc, dans la liste des échantillons D_e, bien identifier les échantillons indiquant le début ou la fin d'une composante. Ces échantillons se reconnaissent par la présence d'un point dont la pression est nulle, soit juste avant dans la liste D_e pour les débuts de composante, ou juste après dans la liste D_e pour les fins de composante. Une fois les points de pression nulle éliminés, un point identifiant une fin de composante est toujours suivi d'un point indiquant le début de la suivante, sauf bien entendu si le point de fin correspond à la fin du tracé.

Le second rôle de la procédure de filtrage consiste à éliminer les échantillons successifs identiques. Lorsque le stylo reste immobile pendant l'acquisition, plusieurs échantillons successifs ont les mêmes coordonnées. Dans un système de vérification utilisant la vitesse comme espace de représentation, cela constitue une information importante. Le présent système fait appel à la cinématique uniquement pour établir une séquence qui traduit la

progression du tracé dans l'espace. Les échantillons redondants sont donc éliminés sans que cela nuise au séquencement. Un échantillon est considéré redondant si sa coordonnée (x,y) est identique à celle de l'échantillon précédent dans la liste D_e.

Le résultat final est la liste fitrée D_e' qui ne contient que les points nécessaires et suffisants à l'identification de la trajectoire d'un tracé manuscrit. L'exemple de la lettre *t* est repris à la Figure 3.8 afin d'illustrer l'action du filtrage des échantillons.



(a) (b)

Figure 3.8 Échantillons avant (a) et après (b) filtrage.

3.2.3 Choix de la méthode d'interpolation

L'interpolation entre les points est réalisée en utilisant la méthode des splines cubiques. Cette technique procède en calculant une fonction de degré 3 entre chacun des points de contrôle fournis. Les points de contrôle utilisés ici sont les échantillons captés par la tablette. Une condition de continuité est imposée à chacun des points de contrôle de sorte que la dérivé première existe et que la dérivé seconde est nulle [GERA89].

L'interpolation se fait séparément pour les deux coordonnées. Les échantillons forment deux fonctions: les points associés à une coordonnée sont exprimés en fonction de n, le rang de l'échantillon dans la liste. L'interpolation est faite entre chaque paire de points de façon récursive tant que les points ne sont pas voisins dans l'espace XY (en considérant une connexité 8). La procédure d'interpolation est décrite de façon détaillée dans l'Annexe E.

Le choix de la technique des splines cubiques est motivé par plusieurs raisons. Premièrement, cette technique d'interpolation fait passer la courbe interpolée par tous les points de contrôle disponibles. Étant donné que ces points sont, ici, des échantillons de la trajectoire, il est nécessaire que la séquence résultant de l'interpolation passe par ces points.

Le comportement de la courbe interpolée entre deux points dépend uniquement des conditions de continuité imposées aux points de contrôle. Et la condition dépend en partie de la position des voisins du point pour lequel elle est calculée. Étant donné que les points de contrôle considérés sont des échantillons pris à intervalles constants et à une fréquence suffisamment élevée, il est permis de croire que leur répartition est suffisamment représentative du comportement du tracé. Cette caractéristique de l'information traitée constitue une autre raison pour le choix de la méthode des splines cubiques.

La procédure d'interpolation a comme résultat la création d'une liste de points représentant la séquence interpolée D_{int}. Cette liste de points contient, en plus des points échantillons de la liste D_e', des points décrivant la trajectoire du tracé entre chacun des échantillons [Figure 3.3]. L'interpolation n'est évidemment pas faite entre le point indiquant la fin d'une composante et celui marquant le début d'une autre.

3.3 Formation des zones caractéristiques

Contrairement au traitement de l'information statique, une seule étape de formation de zones est effectuée ici étant donné la facilité avec laquelle celles-ci sont formées. Ces zones représentent soit des *extremums*, soit des *extrémités*.

3.3.1 Localisation et identification des points caractéristiques

Dans la représentation dynamique, le signal n'est pas perturbé par des croisements de traits. Les croisements dans l'image sont justement à la base du problème du séquencement; ainsi, lorsqu'il n'y a pas de croisement dans une image de signature, la solution devient triviale.

Puisqu'il n'y a pas de croisements associés aux signaux dynamiques, les seules discontinuités identifiables sont celles représentant les extrémités de traits. Comme il ne s'agit pas ici d'identifier des points sur les contours du tracé mais bien sur la séquence représentant le tracé, ces points sont facilement identifiables: ce sont les points identifiant le *début* et à la *fin* des composantes.

La définition d'un extremum est ici la même que celle utilisée pour l'identification des points extremums (section 2.3.1). Les extremums identifiables sont:

Xi : extremum inférieur en X

Xs : extremum supérieur en X

Yi : extremum inférieur en Y

Ys : extremum supérieur en Y

Puisque les extremums sont identifiés sur la séquence et non sur les contours, la notion de complémentarité est caduque. Il est en effet impossible de qualifier l'extremum en fonction de sa position par rapport au tracé, puisque le tracé se résume ici à une liste de points.

La liste des points caractéristiques P_{car} contient donc à la fois les points *extremums* et les *extrémités de trait*. La Figure 3.4 montre les différents points caractéristiques P_{car} identifiés pour la séquence interpolée de la Figure 3.3.

3.3.2 Validation des points caractéristiques

Le processus d'acquisition peut engendrer du bruit dans la séquence de points. Ce bruit se traduit par la création d'extremums supplémentaires. Il faut donc procéder à une validation des extremums identifiés de façon à ne conserver que ceux jugés pertinents.

La pertinence est déterminée en fonction de la distance qui sépare un extremum de ses voisins extrema de même orientation dans la liste de points. Tout comme pour la validation des extremums statiques, la distance seuil τ_{int} est égale à *LMT*, l'estimation de la *largeur moyenne du tracé*. La justification du choix de ce seuil est la même que celle faite pour le filtrage des extremums statiques [Annexe C]. Pour qu'une variation dans la trajectoire d'un trait soit significative, il faut que cette variation, par rapport à l'axe du déplacement, représente au moins une fois la largeur du tracé.

L'estimation de la *largeur moyenne du tracé LMT* doit nécessairement se faire à partir d'une *image* de signature manuscrite. Deux solutions s'offrent ici pour le choix de l'image à utiliser pour l'estimation de *LMT* : il peut s'agir soit de l'image correspondant au signal dynamique traité, soit de l'image dans laquelle doit être inférée une séquence. La première solution est possible ici, parce que chaque spécimen de signature composant la banque de données utilisée [chapitre 6] est emmagasiné sous les deux formes de représentation (*image* et *signal dynamique*). La seconde solution est également envisageable étant donné que la banque de données est composée de signaux produits dans un environnement contrôlé: l'utilisation du même outil d'écriture pour la production de tous les spécimens permet de supposer que la valeur du paramètre *LMT* sera relativement stable.

Le seuil τ_{int} permet d'identifier des *régions bruitées*. Une *régions bruitée* est définie par des extremums voisins qui ont chacun au moins un voisin de même orientation à une distance sur la séquence inférieure au seuil τ_{int}. Le mécanisme de sélection des points dans les régions est le même que celui proposé à la section 2.3.2.2 pour le traitement de l'information statique.

3.3.3 Formation des zones caractéristiques

Étant donné la nature de l'information, la formation des zones caractéristiques est plutôt triviale. Puisque le tracé est représenté par une suite de points, les *zones caractéristiques* Z_{car} correspondent aux différents *points caractéristiques* P_{car} identifiés sur la liste de

points de la séquence interpolée D_{int}. Pour chaque point caractéristique, une zone du même type que le point est créée.

3.4 Graphe de représentation

La fonction du graphe de représentation G_{rep} est d'offrir une représentation structurée de l'information extraite de la séquence de points D_{int} représentant le tracé. Ce graphe doit être construit de façon à ce que l'information qu'il véhicule soit compatible avec celle du *graphe de relations* G_{rel} résultant du traitement de l'information statique [section 2.5].

Les *zones caractéristiques* Z_{car} identifiées constituent l'essentiel de l'information extraite. Puisqu'elles représentent les mêmes portions de trait que les zones identifiées dans l'image, elles constituent ici aussi les noeuds du graphe.

La présence d'un arc entre deux noeuds indique que ces noeuds sont voisins dans la séquence. Étant donné que la notion de croisement de traits n'est pas considérée dans la représentation dynamique, il n'y a pas de zones de croisement et chaque noeud a donc, au plus, deux voisins. Il n'y a pas ici d'étape d'édition du graphe, comme c'est le cas avec le graphe de relations G_{rel}, étant donné que les zones ne peuvent être incomplètes.

Le *graphe de représentation* G_{rep} est donc le résultat final du traitement de l'information dynamique. Contrairement au *graphe de relations* G_{rel} extrait de l'image, il est possible

d'établir un séquencement dans le *graphe de représentation* G_{rep}. Cette séquence est dictée par l'ordre d'apparition des *points caractéristiques* P_{car} dans la séquence de points D_{int}. C'est ce séquencement qui permettra, après la mise en correspondance des deux graphes, d'établir une séquence dans l'image de la signature traitée. La Figure 3.5 montre le *graphe de représentation* G_{rep} résultant du traitement dynamique de la séquence échantillonnée D_e montrée à la Figure 3.2.

4. MISE EN CORRESPONDANCE DES GRAPHES

4.1 Description générale

Suite aux étapes précédentes, l'information à traiter se résume à deux graphes: le *graphe de relations* G_{rel}, résultat du traitement de l'information statique, et le graphe de représentation G_{rep}, résultat du traitement de l'information dynamique. Ces deux graphes représentent le positionnement spatial de différents éléments caractéristiques de la signature et les relations qui existent entre ces éléments. Étant donné la présence dans le *graphe de relations* G_{rel} d'éléments représentant des discontinuités telles que des *croisements* et des *boucles dégénérées*, il est difficile d'identifier une séquence entre les noeuds du graphe qui traduirait la progression spatiale du tracé de la signature.

Cette information existe dans le graphe de représentation G_{rep}, qui est justement un graphe orienté représentant la progression du signal dynamique dans l'espace. La mise en correspondance du *graphe de relations* G_{rel} avec le graphe de représentation G_{rep} vise donc l'identification d'un parcours dans G_{rel} qui représenterait la progression du tracé de la signature. Les différentes étapes qui mènent à cette mise en correspondance sont résumées par le *DFD* de la Figure 4.1.

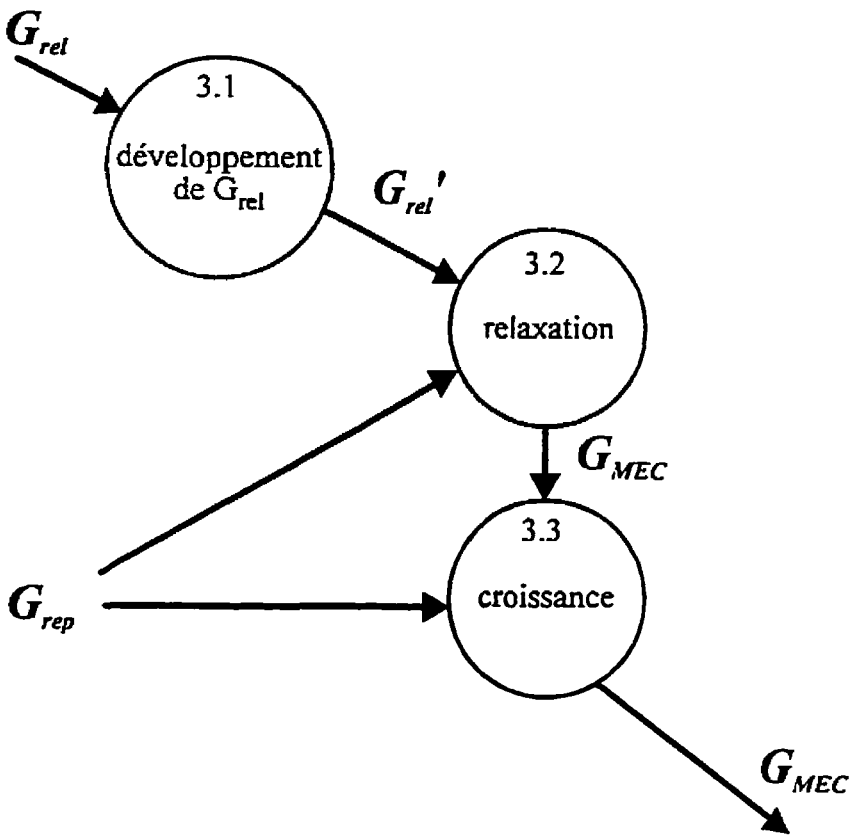


Figure 4.1 *DFD* résumant les étapes de traitement pour la mise en correspondance.

Une des conditions *sine qua non* pour qu'il y ait mise en correspondance entre les graphes est que ces graphes soient composés d'éléments identiques. Ce n'est pas le cas ici puisque le *graphe de relations* G_{rel} peut contenir des éléments de discontinuité (les *croisements* et les *boucles dégénérées*). La première tâche à accomplir est donc le *développement* de G_{rel} afin d'éliminer les éléments de discontinuité. En fait, ces éléments ne sont pas éliminés, mais plutôt *interprétés* afin de les remplacer par des éléments qui se retrouvent dans les

deux graphes, soient des *extremums* ou des *extrémités*. Les interprétations sont faites en fonction de la position des noeuds voisins des discontinuités. Le résultat du développement est donc un *graphe de relations développé* G_{rel}' qui n'est composé que d'*extremums* et d'*extrémités*.

Après l'étape de *développement*, il est possible de procéder à la mise en correspondance. Cette tâche a pour but d'associer un noeud du graphe de représentation G_{rep} à chaque noeud du *graphe de relations développé* G_{rel}', de façon à établir un parcours dans ce dernier. Le problème peut donc être vu comme un problème d'*étiquetage consistant* («consistant labeling»), problème commun à quantité de système d'interprétation d'images.

Le problème de l'étiquetage consistant peut se résumer ainsi: des *étiquettes* doivent être assignées à des *objets* à raison d'une *étiquette* par *objet*, et ce en respectant certains critères. Mais il est des cas où plus d'une *étiquette* s'applique à un même *objet*. C'est notamment le cas ici, puisque la mise en correspondance se fait entre les noeuds de même type et qu'il y a plus d'un noeud de même type dans chacun des graphes. Une synthèse des différentes techniques développées pour surmonter ce problème a été fait dans WALL88.

Parmi toutes les méthodes existantes, celle retenue pour effectuer la mise en correspondance est l'étiquetage par *relaxation* [SHAP90]. Ce choix est d'abord motivé

par l'aspect local de la solution puisque cette méthode permet l'utilisation de l'information contextuelle. Sans oublier le caractère parallèle de la méthode qui traite tous les éléments de façon simultanée, ce qui a également l'avantage de diminuer la complexité algorithmique de la solution.

Une fois la procédure de *relaxation* terminée, il est possible que certains noeuds n'aient pas été mis en correspondance. Un second traitement est donc implanté pour tenter de former davantage de paires de noeuds. Il est ici question de mise en correspondance par *croissance consistante*, car de nouveaux couples de noeuds sont ajoutés seulement s'ils peuvent être rattachés à des séquences de noeuds déjà mis en correspondance.

Le résultat de ces étapes de mise en correspondance est le *graphe de mise en correspondance* G_{MEC}, qui est en fait une version orientée du *graphe de relations développé* G_{rel}' et dans laquelle certains noeuds sont ignorés. C'est à partir de ce nouveau graphe que la séquence est inférée dans l'image de la signature.

4.2 Développement du graphe de relations

4.2.1 Caractéristiques du développement

Le but du *développement* est d'obtenir un *graphe de relations développé* G_{rel}' qui possède les mêmes caractéristiques structurelles que le *graphe de représentation* G_{rep} afin de permettre la mise en correspondance entre les éléments de ces deux graphes.

Les éléments qui font l'objet du *développement* sont ceux qui ne peuvent exister que dans G_{rel}, soit les noeuds représentant des *discontinuités* (les *croisements* et les *boucles dégénérées*) ou des *extrémités*. Tous les noeuds de ces types présents dans G_{rel} doivent être remplacés par un ou plusieurs noeuds *extremums* ou *extrémités*. Le développement d'un noeud se fait en considérant la position spatiale du noeud par rapport à chacun de ses voisins immédiats dans le graphe. Les voisins d'un noeud à développer sont considérés individuellement ou par paire selon le type du noeud à développer et le nombre de ses voisins.

Puisque dans le *graphe de représentation* G_{rep}, tous les noeuds ont au maximum deux voisins, une des caractéristiques du *développement* est de regrouper, après développement, les voisins de chaque noeud en un maximum de deux groupes. Par exemple [Figure 4.2], lorsqu'un noeud *V1* a comme voisin un noeud **C** qui doit être développé et, qu'après développement, **C** est remplacé par plus d'une interprétation, le

nombre de voisins de *V1* peut augmenter. Puisque ces nouveaux voisins sont le résultat du développement d'un seul et même noeud, ici C, ils forment un seul groupe de voisins.

Pour ce qui est des noeuds développés, les interprétations qui en résultent ont comme voisins les noeuds impliqués dans l'interprétation qui a mené à leur création. Puisque ces interprétations sont faites en considérant un maximum de deux voisins, chacune des interprétations a elle aussi un maximum de deux voisins. Par exemple [Figure 4.2a], l'interprétation du noeud C par rapport à ses voisins *V1* et *V2* donne naissance au noeud Ys. Ce dernier a donc comme voisins les noeuds *V1* et *V2*; chacun de ses voisins peut aussi devenir un groupe de voisins s'il doit également être développé.

Donc, le résultat du développement est un *graphe de relations développé* G_{rd}' qui n'est composé que de noeuds *extremums* et *extrémités*, et dont les arcs émanant d'un même noeud sont regroupés en un maximum de deux groupes, chacun correspondant à un voisin des deux voisins possibles. Les prochaines sections expliquent en détail comment procéder au développement de chacun des types de noeud.

4.2.2 Développement des croisements

Les noeuds dits de *croisement* témoignent de la rencontre de deux ou de plusieurs traits. Lorsque deux traits se croisent, il est possible que le point de rencontre de ces traits

corresponde à un extremum d'un des deux traits, ce qui a pour effet de le confondre avec la zone de croisement.

Le développement d'un noeud de croisement consiste donc, dans un premier temps, à considérer toutes les paires de voisins comme les noeuds *afférent* et *efférent* d'un trait passant par le croisement traité. La position du noeud par rapport à chaque noeud d'une paire de voisins détermine s'il doit être remplacé par un autre noeud et, si oui, par quel type de noeud. Si le croisement peut être interprété comme un *extremum* de même type par rapport aux deux voisins, alors il est remplacé par cet *extremum*, sinon les deux voisins sont considérés voisins entre eux. Par exemple [Figure 4.2a], le *croisement* **C** constitue un extremum **Xs** par rapport aux voisins *V1* et *V3*, alors il est remplacé par cet extremum et ce dernier est relié aux noeuds *V1* et *V3*. Mais ce même noeud **C** ne représente aucun extremum par rapport aux voisins *V1* et *V4*, ce qui explique la présence d'un arc entre ces deux noeuds après le développement.

Ce mode d'interprétation procède selon l'hypothèse que chaque *croisement* est composé par des superpositions de traits. Autrement dit, tous les voisins d'un *croisement* peuvent être groupés par paires représentant un même trait. Mais lorsque le nombre de voisins d'un *croisement* est impair, il y a nécessairement un noeud qui n'appartient pas à une paire. Il est probable que cette zone cache une *extrémité* de trait. Donc, dans le cas où un

noeud *croisement* a un nombre impair de voisins, la zone est interprétée, en plus, comme l'*extrémité* potentielle de chacun des traits la reliant à ses voisins [Figure 4.2b].

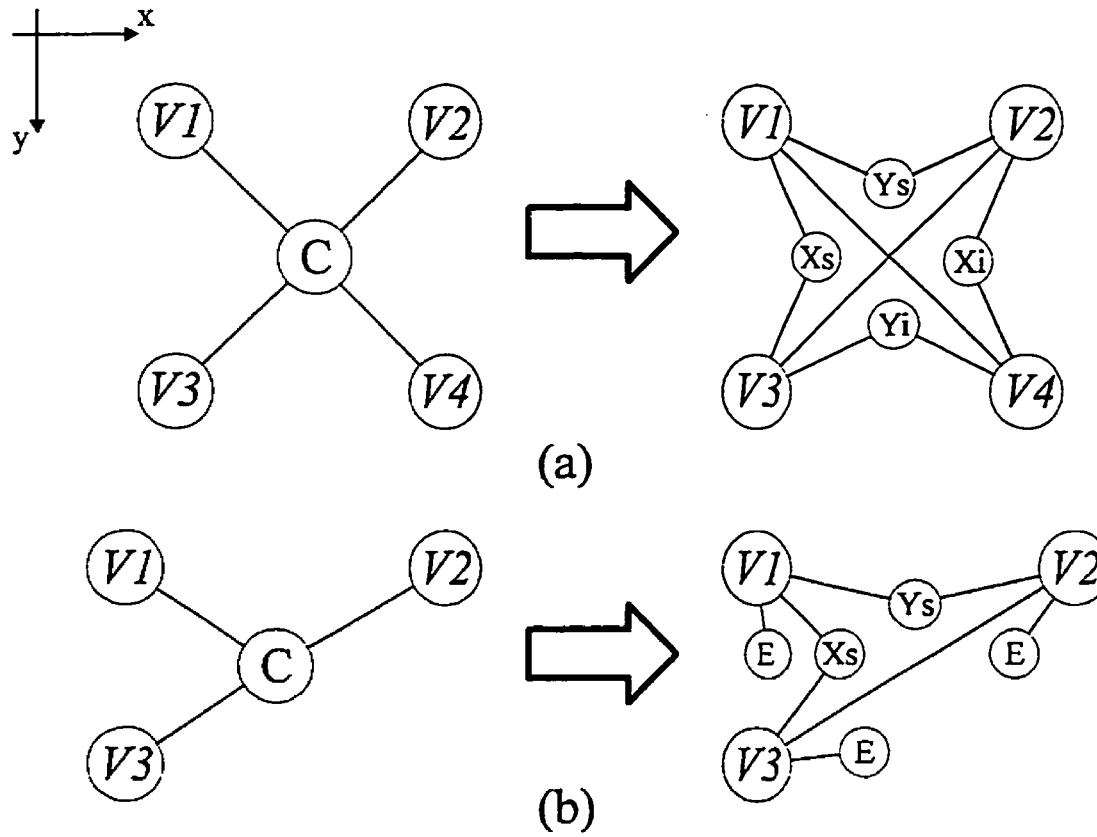


Figure 4.2 Exemples théoriques illustrant l'opération de développement de *croisements*.

Étant donné que la zone de *croisement* peut cacher un *extremum* ou une *extrémité*, chacune des interprétations est positionnée à la même place que la zone de *croisement* développée. Les coordonnées indiquant le centre du *croisement* servent également de centre pour toutes les interprétations de cette zone. Ces coordonnées vont servir à la mise en correspondance.

4.2.3 Développement des boucles dégénérées

L'étiquette *boucle dégénérée* fait référence à des zones de tracé correspondant à des boucles trop petites pour laisser paraître le contour intérieur de la boucle, ce qui rend donc impossible l'identification des zones *extremums*. Le cas extrême d'une telle dégénérescence est le rebroussement de tracé, ce qui délimite une chaîne telle que définie par Plamondon [PLAM92a]. Puisqu'elle n'est composée que de deux points caractéristiques, une zone de *boucle dégénérée* ne peut avoir que deux voisins. C'est en comparant sa position par rapport à chacun de ses voisins que les interprétations possibles sont trouvées.

L'interprétation de la position d'une boucle dégénérée par rapport à l'un de ses voisins engendre un maximum de deux extremums. Parmi les interprétations trouvées, au moins une est commune aux deux voisins. Dans certain cas, les deux interprétations sont les mêmes. Le nombre d'extremums qui remplacent la boucle dégénérée varie donc entre deux et trois, selon la position des voisins.

La Figure 4.3 illustre ces deux situations. En (a) les interprétations par rapport à *V1* sont **Yi** et **Xs** et celles par rapport à *V2* sont **Xi** et **Yi**. Comme une des interprétations est commune aux deux voisins, la *boucle dégénérée* est remplacée par trois extremums: **Xi**,

Yi et **Xs**. En (b) les interprétations sont les mêmes pour les deux voisins: **Yi** et **Xs**. Deux *extremums* correspondants remplacent donc la *boucle dégénérée*.

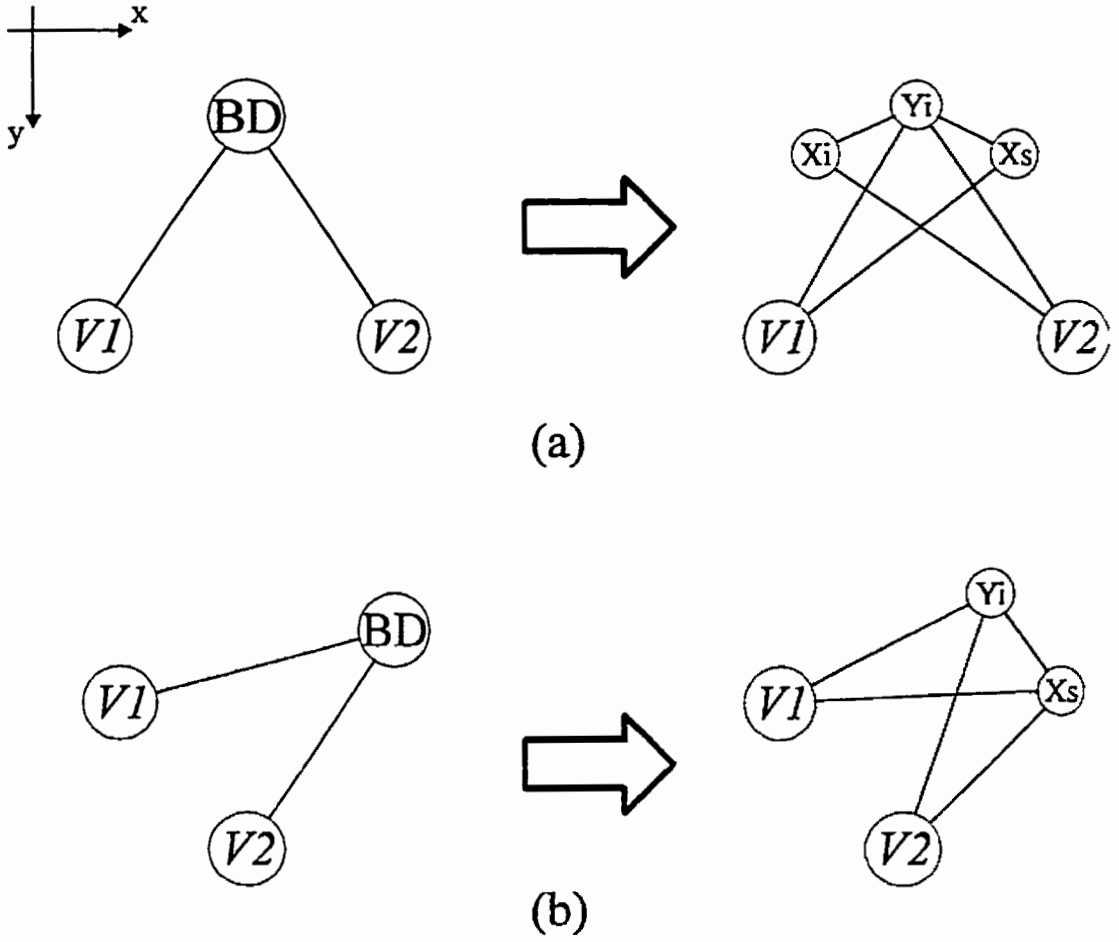


Figure 4.3 Exemples théoriques illustrant l'opération de développement de *boucles dégénérées*.

Contrairement aux *croisements* où les centres des interprétations sont toujours positionnés à la même place que le centre de la zone développée, le cas des boucles dégénérées est plus complexe. Il est en effet nécessaire d'estimer la position de chacune des

interprétations, puisqu'il ne s'agit pas ici de superposition de zones mais bien d'une fusion de zones causée par un manque d'information. Dans un premier temps, il faut identifier un point de contour pour chaque interprétation possible, point qui va servir au positionnement final de la zone interprétée.

L'identification des points de contour peut se faire de deux façons selon le comportement du contour dans la zone de *boucle dégénérée*. Une première solution consiste à trouver, lorsque c'est possible, des points de contour qui correspondent à des *extremums* de même type que les interprétations. Ces extremums sont cherchés à partir du point *extrémité* de la *boucle dégénérée* concernée. La Figure 4.4 illustre deux *boucles dégénérées* (formées des *discontinuités* **D** et des *extrémités* **E**). Le développement de la zone en (a) donne deux interprétations, Yi et Xs, qui trouvent leur équivalent comme point de contour. Il en va différemment pour les interprétations Xs et Yi de la zone en (b).

Pour ce qui est de l'interprétation Xi, aucun extremum de ce type n'existe sur le contour près de la *boucle dégénérée*; cela peut résulter d'une erreur de binarisation. Il faut donc procéder autrement dans ce genre de situation. La solution proposée pour identifier un point sur le contour consiste à choisir le point qui se trouve à égale distance des points E et D formant la boucle dégénérée [Figure 4.4b]. La procédure à suivre est de calculer pour tous les points, en débutant par le point voisin de E, les distances euclidiennes d_1 et d_2, et de retenir le point pour lequel d_1 et d_2 sont égales.

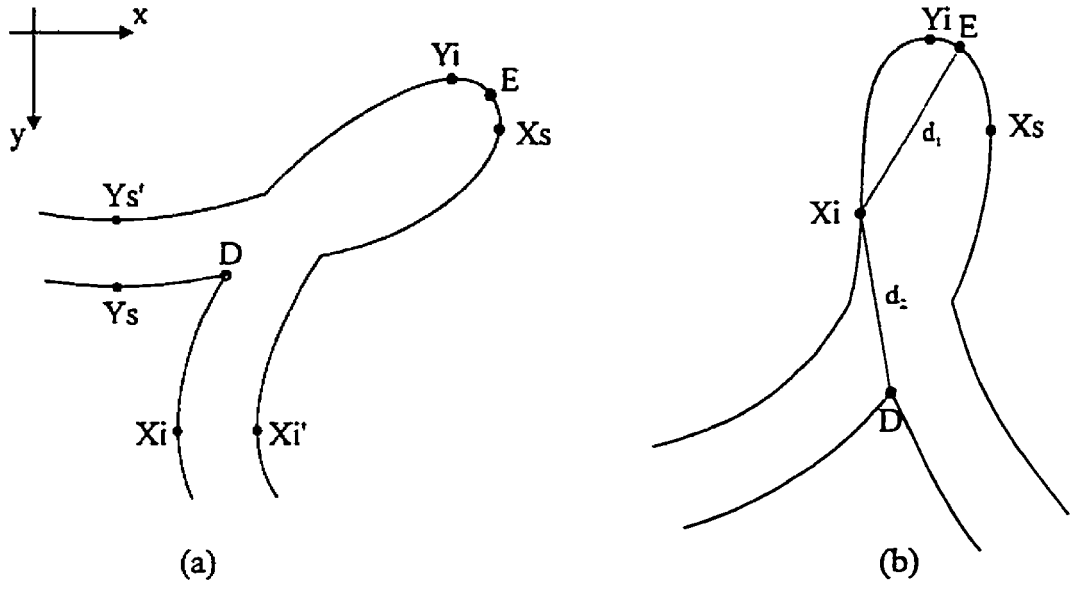


Figure 4.4 Exemples théoriques illustrant le positionnement des points sur les contours de *boucles dégénérées*.

Les points ainsi identifiés ne correspondent évidemment pas à des centres de zones puisqu'ils sont situés sur les contours du tracé. Il faut donc, à partir de ces points, trouver des points dans le tracé, qui eux vont correspondre aux centres des *zones interprétées*. En admettant l'hypothèse que l'outil d'écriture utilisé produit un trait symétrique par rapport à l'axe du déplacement, le centre d'une zone caractéristique se situe au centre du trait. En admettant que l'estimation de la largeur moyenne du tracé par *LMT* est exacte, le centre d'une zone caractéristique se situe à une distance de *LMT/2* de tout contour. Le point central d'une zone doit donc se situer à une distance d_c=*LMT/2* du contour, à moins que le tracé à ce point soit de largeur inférieure à LMT, auquel cas le point se situe au milieu du tracé. Ceci est résumé par l'équation suivante:

$$d_c = \min(\frac{LMT}{2}, \frac{lt}{2}) \qquad (4\text{-}1)$$

La position du centre de la zone est calculée à partir du point correspondant préalablement identifié en suivant la droite perpendiculaire au tracé [Figure 4.5]. C'est cette même droite qui sert à l'estimation locale de la largeur du tracé *lt*: cette mesure correspond à la distance séparant les deux points où cette droite croise le tracé. Si cette distance est supérieure à *LMT*, alors le point central est placé à une distance de *LMT/2* du contour, sinon il est situé à la distance égale à la moitié de la largeur du tracé à ce point, soit *lt*/2.

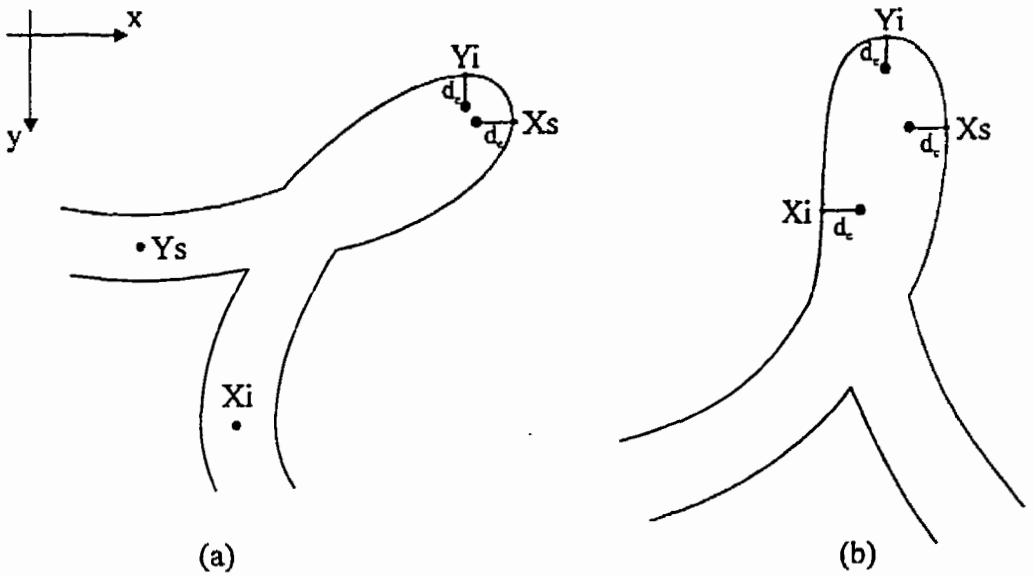
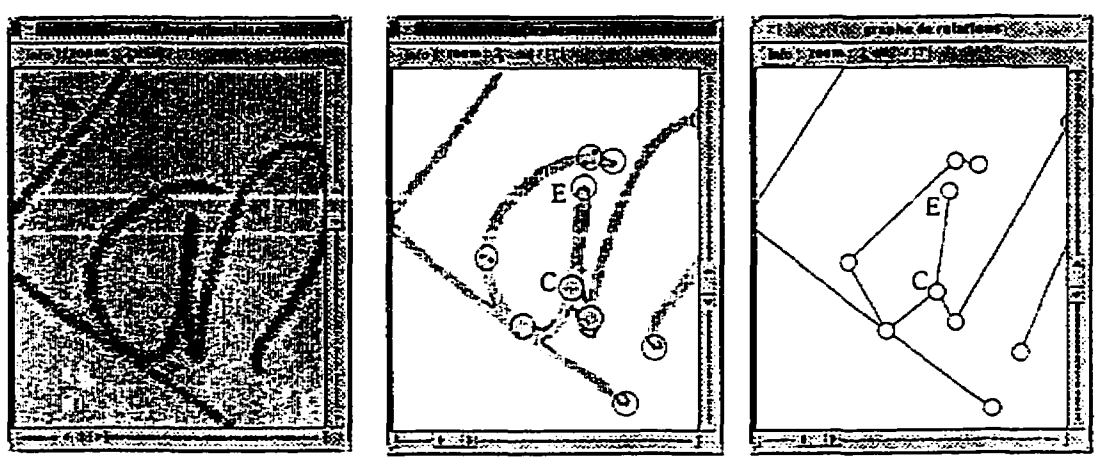


Figure 4.5 Exemples théoriques illustrant le positionnement des centres des zones résultant du développement de *boucles dégénérées*.

4.2.4 Développement des extrémités

Même si les noeuds *extrémités* sont des noeuds qui existent dans le graphe de représentation G_{rep}, et qu'ils ne devraient donc pas faire l'objet d'une phase de développement, certains d'entre eux sont tout de même soumis à cet exercice. Il s'agit des noeuds *extrémités* qui ont comme voisin un noeud de *croisement* (les extrémités ont toujours un seul voisin). Il faut en plus que le *croisement* voisin ait lui-même un nombre impair de voisins, ce qui laisse présager un rebroussement de trait. Il est en effet possible que de telles configurations soient le résultat d'une *boucle dégénérée* mal identifiée, comme c'est le cas dans la figure qui suit.



(a) (b) (c)

Figure 4.6 Exemple de *boucle dégénérée* identifiée comme une *extrémité* E voisine d'un *croisement* C. Détails de l'image de luminance (a), de l'image binarisée (b) et du graphe de relations (c).

Dans un cas comme celui de la Figure 4.6, la zone *extrémité* **E** est développée en considérant sa position spatiale par rapport au *croisement* **C** voisin. L'interprétation de la zone comme une *extrémité* est cependant conservée, puisqu'il est possible qu'il s'agisse véritablement d'une *extrémité*. La mécanique du développement reste cependant la même que celle décrite dans les sections précédentes, comme le montre l'exemple de la prochaine figure.

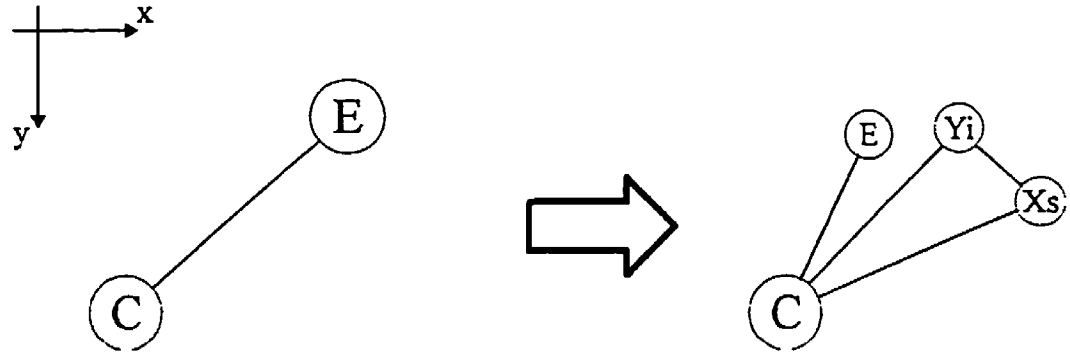


Figure 4.7 Exemple théorique illustrant l'opération de développement d'une *extrémité*. Pour qu'une telle zone soit développée, il faut obligatoirement qu'elle soit voisine d'un *croisement*.

Les trois possibilités de développement décrites jusqu'ici permettent d'obtenir le *graphe de relations développé* G_{rel}' qui ne contient que des noeuds de type extremum et extrémité tel que montré à la Figure 4.8. Ce graphe est impliqué dans la mise en correspondance avec le graphe de représentation G_{rep} associé à l'information dynamique utilisée comme référence.

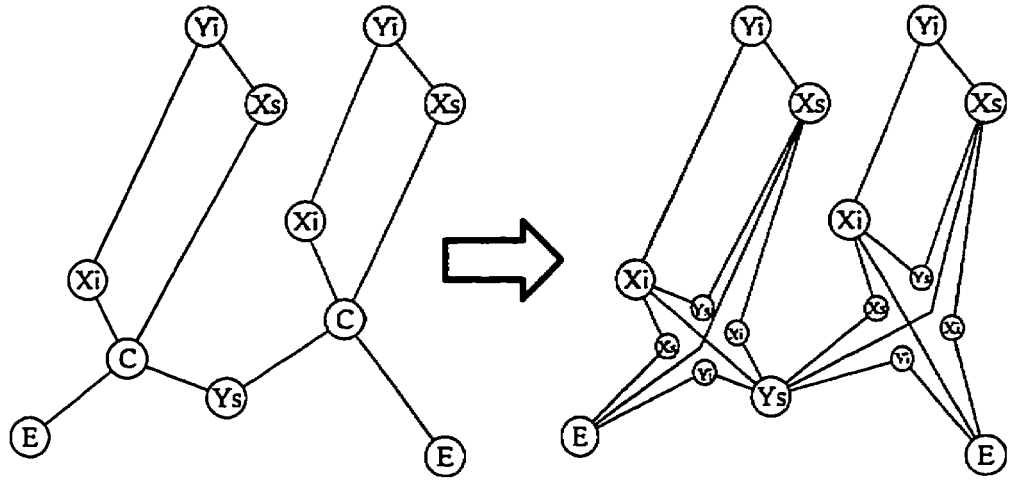


Figure 4.8 Exemple d'un *graphe de relations développé* G_{rel}'. Ce graphe est le résultat du développement du graphe présenté à la Figure 2.2 b.

4.3 Mise en correspondance par relaxation

4.3.1 Description de la procédure de mise en correspondance

La tâche de la mise en correspondance s'apparente ici à un problème d'étiquetage consistant. Un tel problème existe lorsque les quatre composantes suivantes sont identifiables:

1) un ensemble d'*objets* à étiqueter;

2) un ensemble d'*étiquettes* pour chaque *objet*;

3) une relation de voisinage entre les *objets*;

4) une relation de contrainte entre les *étiquettes* pour chaque paire d'*objets* voisins.

Dans le cas présent, les *objets* sont les noeuds du *graphe de relations développé* G_{rel}' et les *étiquettes* sont les noeuds du graphe de représentation G_{rep}. La mise en correspondance consiste donc à chercher une *étiquette* pour chaque *objet*. La relation de voisinage entre les *objets* correspond aux arcs de G_{rel}'. De même que la relation de contrainte entre les *étiquettes* correspond à la présence d'arcs entre les noeuds de G_{rep}.

La technique retenue pour résoudre ce problème est celle de l'étiquetage par *relaxation* qui permet de rendre une décision sur le choix d'une étiquette parmi un ensemble [HUMM83]. Cette technique s'inscrit dans le cadre d'un algorithme de mise en correspondance [Tableau 4.1]. Les différentes tâches associées à ce traitement sont décrites dans les prochaines sections.

Tableau 4.1 Algorithme synthèse de la procédure de mise en correspondance par relaxation

1:	**RÉPÉTER**
2:	Attribution des poids initiaux
3:	**RÉPÉTER**
4:	Mise à jour des poids par relaxation
5:	**TANT QUE** les conditions d'arrêt non rencontrées
6:	Validation des mises en correspondance
7:	**TANT QUE** de nouvelles mises en correspondance sont effectuées

4.3.2 Attribution des poids initiaux

Dans un premier temps, des ensembles d'étiquettes Λ_i sont formés pour tous les objets i. Dans chacun de ces ensembles, une étiquette supplémentaire est ajoutée: il s'agit de l'étiquette nulle $\varnothing$. Cette étiquette doit être ajoutée afin de rendre possible l'identification des objets pour lesquels aucune mise en correspondance ne peut être faite entre deux noeuds. Chacune des *étiquettes* $\lambda \in \Lambda_i$ se voit attribuer un *poids initial* $p_i(\lambda)$ traduisant le degré d'appartenance de l'étiquette λ à l'ensemble Λ_i.. Ce poids peut être binaire (0 ou 1, *appartient* ou *n'appartient pas*) ou défini sur un continuum (par exemple entre 0.0 et 1.0), auquel cas une fonction d'appartenance doit être définie. Dans le cas présent, le poids initial est basé sur trois facteurs combinés en une fonction d'appartenance. Ces trois facteurs sont:

1) le type de l'*objet* et de l'*étiquette*;
2) la différence de position spatiale relative entre l'*objet* et l'*étiquette*;
3) la différence entre les angles formés par la position des voisins de l'*objet* et de l'*étiquette*.

Premièrement, en ce qui concerne le type, il faut que l'objet et l'étiquette soient tous deux de même type. Une fonction binaire $t_i(\lambda)$ est donc définie comme suit:

$$t_i(\lambda) = \begin{cases} 1 & si\ type(\lambda) = type(i) \\ 0 & autrement \end{cases} \qquad (4\text{-}2)$$

Pour ce qui est de la différence de position spatiale, elle correspond à la mesure $dp_i(\lambda)$ qui exprime la distance euclidienne des différences de positions spatiales entre l'objet i et l'étiquette λ. Puisque l'objet et l'étiquette appartiennent à des espaces différents, les positions doivent être normalisées afin de rendre possible leur comparaison. Les facteurs de normalisation utilisés sont [Figure 4.9]:

d_{Xs} la taille en X du signal I_{bin} (signal statique)

d_{Ys} la taille en Y du signal I_{bin}

d_{Xd} la taille en X du signal D_{int} (signal dynamique)

d_{Yd} la taille en Y du signal D_{int}

Ces valeurs, utilisées comme dénominateurs dans les mesures de distance, permettent d'exprimer des positions variant entre 0.0 et 1.0, ce qui rend donc les comparaisons possibles.

Les positions sont exprimées par rapport aux centres de masse de chacun des signaux. En utilisant un tel référentiel, plutôt qu'un des coins du rectangle circonscrit au signal, les effets de la variation du signal sont minimisés. Ces différentes valeurs, illustrées à la Figure 4.9, sont:

μ_{Xs} la moyenne en X du signal I_{bin} (signal statique)

μ_{Ys} la moyenne en Y du signal I_{bin}

μ_{Xd} la moyenne en X du signal D_{int} (signal dynamique)

μ_{Yd} la moyenne en Y du signal D_{int}

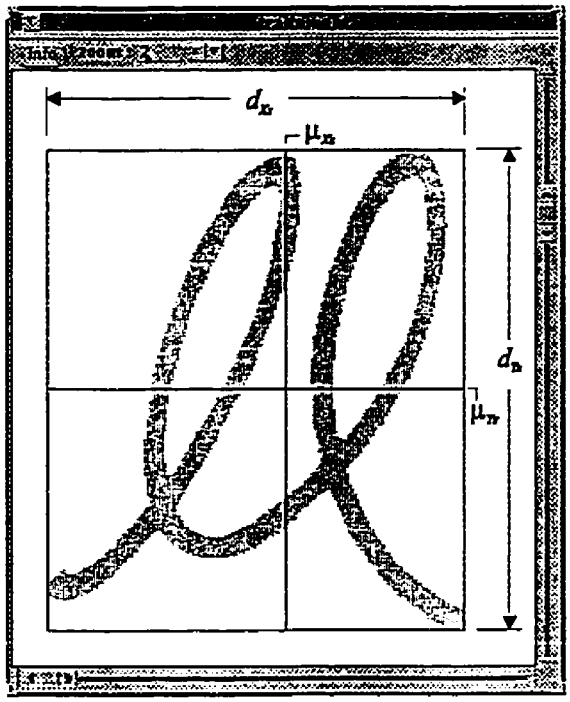


Figure 4.9 Représentation graphique des différentes valeurs tirées de l'image, utilisées pour l'estimation et la normalisation des positions spatiales. Les paramètres correspondants sont extraits de la séquence de la même façon.

La valeur de $dp_i(\lambda)$ dépend donc des positions des noeuds par rapport au centre de masse; les positions sont normalisées par rapport à la taille des signaux. Ces positions sont exprimées par rapport à chacun des deux axes (X et Y), pour un total de quatre positions estimées. Une distance euclidienne est ensuite calculée à partir de ces quatre positions.

Cette distance traduit la différence de position entre les deux noeuds. La valeur du facteur $dp_i(\lambda)$ dépend directement de cette distance comme l'indique l'équation qui suit:

$$dp_i(\lambda) = \sqrt{\frac{\left|\frac{x_i - \mu_{Xs}}{d_{Xs}} - \frac{x_\lambda - \mu_{Xd}}{d_{Xd}}\right|^2 + \left|\frac{y_i - \mu_{Ys}}{d_{Ys}} - \frac{y_\lambda - \mu_{Yd}}{d_{Yd}}\right|^2}{2}} \qquad (4\text{-}3)$$

La division de la somme des carrés par 2 restreint les valeurs de $dp_i(\lambda)$ à la plage 0 à 1. La valeur est minimum si la position de l'*étiquette* correspond exactement à celle de l'*objet*, et maximum lorsque la position de l'*étiquette* est diamétralement opposée à celle de l'*objet* (par exemple, une *étiquette* située dans le coin supérieur droit du signal dynamique et un *objet* situé dans le coin inférieur gauche du signal statique).

Le troisième facteur concerne les angles formés par l'*objet* et l'*étiquette* avec leurs voisins respectifs. Les deux angles impliqués dans cette mesure sont formés par deux droites se rejoignant au centre de l'objet (ou l'étiquette) et passant chacun par le centre de chacun des voisins de l'objet (ou l'étiquette). Le facteur d'angle est le résultat de la différence de ces angles, comme le montre l'équation suivante:

$$da_i(\lambda) = \frac{|\theta_s - \theta_d|}{\pi} \qquad (4\text{-}4)$$

Dans cette équation, θ_s représente l'angle associé à l'objet et θ_d celui associé à l'étiquette. Étant donné que cette mesure n'est invoquée que lorsque l'objet et l'étiquette sont de même type, la différence entre les deux angles ne peut excéder π. La division de la différence par cette valeur limite donc la variation de $da_i(\lambda)$ entre 0 et 1.

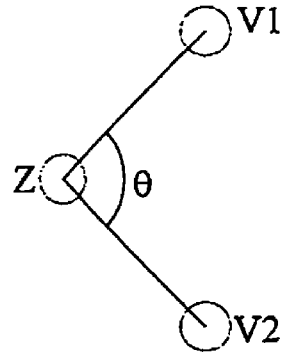


Figure 4.10 Illustration de la signification des angles θ_s et θ_d utilisés pour l'évaluation du facteur angulaire $da_i(\lambda)$. L'angle θ est associé au noeud Z qui a comme voisins les noeuds V1 et V2. La méthode de mesure est la même dans les deux graphes.

Reste maintenant à combiner les trois facteurs $t_i(\lambda)$, $dp_i(\lambda)$ et $da_i(\lambda)$ afin d'obtenir la fonction d'appartenance qui détermine la valeur initiale de $p_i(\lambda)$. La combinaison retenue est:

$$p_i(\lambda) = t_i(\lambda)\left\{0.5*\left[1-dp_i(\lambda)\right]^2 + 0.5*\left[1-da_i(\lambda)\right]\right\} \qquad (4\text{-}5)$$

Donc, le poids initial est automatiquement nul si l'étiquette λ et l'objet i ne sont pas de même type. S'ils sont de même type, alors le poids varie entre 0 et 1: il est maximum si les

facteurs $dp_i(\lambda)$ et $da_i(\lambda)$ sont nuls. Étant donné le poids égal donné à ces deux facteurs, chacun est responsable pour la moitié de la valeur de la fonction d'appartenance. L'expérience ayant montré que les variations de positions sont plus significatives, la décision a été prise d'élever au carré le terme de l'équation contenant le facteur de position $dp_i(\lambda)$. De cette façon, la tolérance sur les écarts de positions est diminuée sans pour autant augmenter le poids de ce facteur dans le résultat.

En ce qui a trait au poids de l'étiquette nulle $\varnothing$, il dépend du nombre d'étiquettes ayant un poids non nul. Puisque la mise en correspondance va se faire nécessairement avec une des n étiquettes ayant un poids non nul ou avec l'étiquette $\varnothing$, la probabilité que cette dernière soit retenue est donc de $1/(n+1)$. C'est donc cette valeur qui est utilisée comme poids initial.

La somme de tous les poids associés à un même *objet* doit être égale à l'unité. Ceci est obtenu par une simple normalisation des poids et a pour but la comparaison des poids de différents *objets* entre eux. Les poids normalisés $p_i(\lambda)^N$ sont donc obtenus par l'opération suivante:

$$p_i(\lambda)^N = \frac{p_i(\lambda)}{\sum_{\lambda=1}^{m} p_i(\lambda)} \qquad (4\text{-}6)$$

$$\sum_{\lambda=1}^{m} p_i(\lambda)^N = 1$$

4.3.3 Mise à jour des poids par relaxation

L'étape suivante, qui est le coeur de la procédure de relaxation, consiste à effectuer une mise à jour des poids initiaux en considérant les voisins de l'objet concerné. La relation de voisinage utilisée ici pour identifier les voisins est la même que celle utilisée pour le *graphe de relations développé* G_{rel}' (i.e. l'existence d'un arc entre deux noeuds). Pour tous les voisins j de l'objet i, une relation de contrainte $r_{ij}(\lambda,\lambda')$ est calculée pour tous les couples d'étiquettes $\lambda \in \Lambda_i$ et $\lambda' \in \Lambda_j$. Cette relation de contrainte, valant 0 ou 1, indique si les étiquettes λ et λ' peuvent être attribuées aux objets voisins i et j respectivement sans que cela contrevienne aux règles de consistance préétablies. Dans le cas présent, la règle utilisée concerne la présence d'un arc entre les deux étiquettes. Pour que $r_{ij}(\lambda,\lambda')$ soit non nul, il faut simplement que λ et λ' soient voisins dans le graphe G_{rep}. La valeur ainsi obtenue sert à pondérer le poids initial $p_j(\lambda')$ dans le calcul du facteur de mise à jour $q_i(\lambda)$ tel que défini par l'équation suivante:

$$q_i(\lambda) = \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}(\lambda,\lambda') p_j(\lambda') \qquad (4\text{-}7)$$

Cette équation montre que la valeur de $q_i(\lambda)$ correspond à la somme des poids des voisins pour lesquels il existe une étiquette valide λ' qui est elle même voisine de λ. La double sommation indique que ce facteur dépend des m étiquettes des ensembles des n voisins de l'objet i. Dans le cas présent, les voisins sont regroupés en deux groupes, $V1$ et $V2$, et un

seul voisin par groupe est susceptible d'être retenu. L'équation pour le calcul du facteur de mise à jour devient donc:

$$q_i(\lambda) = \max_{j \in V1} \sum_{\lambda'=1}^{m} r_{ij}(\lambda,\lambda')p_j(\lambda') + \max_{j \in V2} \sum_{\lambda'=1}^{m} r_{ij}(\lambda,\lambda')p_j(\lambda') \qquad (4\text{-}8)$$

Ce facteur, qui est un nombre positif, permet la mise à jour des poids de la façon suivante:

$$p_i(\lambda)^{(n-1)} = \frac{p_i(\lambda)^{(n)}[1+q_i(\lambda)]}{\sum_{l=1}^{m} p_i(l)^{(n)}[1+q_i(l)]} \qquad (4\text{-}9)$$

Le dénominateur de cette équation permet de normaliser les nouveaux poids afin que leur somme soit encore égale à l'unité. Dans le cas où $q_i(\lambda)$ est nul cette équation devient la fonction identité puisque le dénominateur vaut l'unité: les poids restent donc inchangés. Plus la valeur de $q_i(\lambda)$ est grande, plus l'importance de $p_i(\lambda)$ augmente par rapport aux autres poids.

Donc, à chaque étape de relaxation certains poids augmentent et d'autres diminuent jusqu'à ce qu'il y ait polarisation des poids près de 0 et 1. Puisque chaque objet a un ensemble de poids initiaux qui lui est propre et donc des indices de mise à jour qui peuvent varier beaucoup, il est impossible de déterminer d'avance le nombre d'itérations nécessaires avant que la polarisation se produise. Le nombre d'itérations est donc établi

dynamiquement en fonction de la plus grande variation de poids $\Delta p^{(n)}$ observée pour une même itération n et de la variation de cette valeur maximum:

$$\Delta p^{(n)} = \max_{\forall i}(p_i(\lambda)^{(n)} - p_i(\lambda)^{(n-1)}) \qquad (4\text{-}10)$$

Les itérations de la procédure de relaxation sont interrompues lorsque la plus grande variation de poids observée est inférieure à 0.05 et que cette valeur maximale affiche une tendance à la baisse. Mathématiquement, cela correspond aux deux conditions suivantes:

$$\Delta p^{(n)} < 0.05 \qquad (4\text{-}11)$$

$$\Delta p^{(n)} - \Delta p^{(n-1)} < 0 \qquad (4\text{-}12)$$

La seconde condition sert uniquement à éviter que la procédure s'arrête après seulement une ou deux itérations, car dans les premières itérations, il est possible que la variation maximale soit inférieure au seuil de 0.05 sans pour autant que la polarisation apparaisse.

Lorsque les deux conditions sont satisfaites, la procédure de relaxation est suspendue. Pour chaque *objet*, les éléments de l'ensemble des *étiquettes* sont triés par ordre décroissant de poids.

4.3.4 Validation des mises en correspondance

Lorsque la procédure de relaxation est terminée, il faut choisir pour chaque *objet* une seule *étiquette* par ensemble. Le choix se porte d'abord sur les *étiquettes* présentant un poids élevé. Pour que la mise en correspondance soit confirmée, il faut que l'*étiquette* concernée n'ait pas été attribuée à un autre *objet*. Si tel est le cas, le choix se porte sur l'étiquette suivante de la liste, jusqu'à ce qu'une étiquette valide soit trouvée. Autrement dit, l'étiquette retenue est celle qui présente le poids le plus élevée *et* qui n'est pas appariée avec un autre objet. Une étiquette fait cependant exception: il s'agit de l'étiquette nulle $\varnothing$. Cette dernière peut être attribuée à plus d'un objet sans que cela pose problème.

Étant donné qu'il est possible qu'une même étiquette constitue le meilleur choix pour plus d'un objet, il est important de définir une procédure pour la confirmation des mises en correspondance. Cette marche à suivre concerne premièrement l'ordre de traitement des objets. Cet ordre est le suivant :

1) les *extrémités*;
2) les *extremums*;
3) les *interprétations*.

Les *interprétations* sont les noeuds créés lors du *développement* de G_{rel}. En traitant ces noeuds en dernier, cela donne préséance aux noeuds représentant des zones clairement

identifiées du tracé. Si, parmi les objets d'une même catégorie, une étiquette représente le meilleur choix pour plus d'un objet, alors le choix se porte sur l'objet pour lequel le poids est maximum.

Parmi toutes les mises en correspondance effectuées, il faut identifier celles qui doivent être conservées. Les premières à être éliminées sont celles impliquant l'étiquette nulle. Ces appariements témoignent que l'objet concerné n'a pas trouvé une autre étiquette convenant à sa situation. Pour que les autres mises en correspondance soient retenues, il faut que l'objet concerné ait un voisin dans G_{rel}' qui soit aussi mis en correspondance. Le fait qu'un voisin soit aussi mis en correspondance renforce l'hypothèse d'exactitude de la mise en correspondance effectuée.

Tous les objets pour lesquelles la mise en correspondance est validée sont identifiés comme tel et ne sont plus impliqués dans les prochaines itérations de la procédure de mise en correspondance. Leur poids $p_i(\lambda)$ est également fixé à 1 de façon à augmenter leur importance relative lors de la mise à jours des poids des autres *objets*. Un *objet* non apparié voisin d'un *objet* apparié profitera de ce poids unitaire.

La procédure de mise en correspondance est reprise tant et aussi longtemps que de nouvelles mises en correspondance valides sont effectuées.

4.3.5 Formation du graphe de mise en correspondance

Lorsqu'aucune nouvelle mise en correspondance ne peut être effectuée, une première version du *graphe de mise en correspondance* G_{MEC} est créée. Les noeuds composant ce graphe correspondent aux couples formés par la procédure de mise en correspondance. Chaque noeud peut donc être relié à la fois à un noeud du *graphe de relations développé* G_{rel}' et à un noeud du *graphe de représentation* G_{rep}. Pour ce qui est de la présence d'un arc entre deux noeuds, il faut tout simplement que ces noeuds soient voisins dans le *graphe de représentation* G_{rep}.

Puisque chaque noeud de G_{MEC} correspond à un noeud de G_{rep} et que les arcs de G_{MEC} sont définis par ceux de G_{rep}, le *graphe de mise en correspondance* G_{MEC} constitue donc un sous-graphe de G_{rep}. Dans le cas où tous les noeuds sont mis en correspondance, alors G_{MEC} et G_{rep} sont identiques. Le *graphe de mise en correspondance* partage donc les mêmes caractéristiques que le *graphe de représentation*, particulièrement celle relative au fait que chacun des noeuds a un maximum de deux voisins.

4.4 Mise en correspondance par croissance consistante

4.4.1 Identification des sous-séquences

Une fois la mise en correspondance par relaxation effectuée, il peut subsister des noeuds du *graphe de représentation* G_{rep} pour lesquels aucun équivalent n'ait été identifié dans le

graphe de relations développé G_{rel}'. La conséquence de ces oublis est la création de sous-graphes (ou sous-séquences) dans le *graphe de mise en correspondance* G_{MEC}.

Lorsqu'un noeud de G_{rep} n'est pas mis en correspondance, les voisins de ce noeud, s'ils sont mis en correspondance, créent des noeuds dans G_{MEC} qui n'ont qu'un seul voisin, caractéristique normalement réservée aux noeuds *extrémité*. De tels noeuds constituent alors le *début* ou la *fin* d'une sous-séquence. Les sous-séquences sont donc caractérisées par une suite de noeuds voisins qui se termine par un noeud ne représentant pas une *extrémité* et n'ayant qu'un seul voisin.

Si des noeuds ne sont pas appariés, le *graphe de mise en correspondance* G_{MEC} se compose de sous-séquences de la séquence de référence exprimée dans G_{rep}. Mais il est possible que certains des noeuds non appariés puissent former de nouveaux noeuds dans G_{MEC} et ainsi s'ajouter à ces sous-séquences de façon consistante par rapport à la référence. La procédure de *mise en correspondance par croissance consistante* a donc comme objectif la formation de nouveaux noeuds à ajouter dans G_{MEC} et ce, en respectant les critères décrits ci-après.

4.4.2 Critères d'ajout

La procédure d'ajout de nouveaux noeuds dans le graphe de *mise en correspondance* G_{MEC} est guidée par trois critères précis. Chacun représente une condition pour la mise en correspondance entre deux noeuds. Ces conditions font appel aux notions suivantes:

1. le *voisinage* du noeud à apparier;
2. la *position* du noeud par rapport au voisin apparié;
3. le *type* du noeud à apparier.

Le premier critère concerne le *voisinage* des candidats. La condition imposée sur le *voisinage* d'un noeud à apparier permet une première sélection dans les candidats à une éventuelle mise en correspondance. Cette condition limite le choix des candidats aux voisins des noeuds formant des terminaisons de sous-séquences. Autrement dit, pour qu'un noeud non apparié fasse l'objet d'une tentative d'appariement, il doit avoir parmi ces voisins un noeud apparié correspondant à un noeud de G_{MEC} identifié comme un *début* ou une *fin* de sous-séquence. Le critère concernant le *voisinage* est satisfait si chacun des deux candidats à une éventuelle mise en correspondance possède un voisin apparié et que ces voisins appariés forment un seul et même noeud dans G_{MEC}. En imposant cette condition aux deux membres d'une éventuelle mise en correspondance, le noeud résultant peut nécessairement être relié à la sous-séquence concernée.

Le second critère impose une restriction sur la *position* des candidats par rapport à leurs voisins formant un noeud commun dans G_{3IEC} (résultat du critère concernant le *voisinage*). La condition imposée sur la *position* concerne la position relative de chacun des candidats par rapport à ces mêmes voisins. Cette position relative est exprimée de façon qualitative: le noeud est qualifié en fonction de sa position horizontale (gauche/droite) et verticale (haut/bas) par rapport au noeud. Cela revient à séparer l'espace en quatre quadrants. Le critère de position est satisfait si les deux candidats sont positionnés de la même façon (dans le même quadrant) par rapport aux voisins préalablement identifiés.

Le troisième et dernier critère concerne le *type* des noeuds à apparier. L'exigence sur l'identité des *types* est la même que celle imposée pour la mise en correspondance par relaxation: pour que deux noeuds soient mis en correspondance, ils doivent nécessairement être de même type. Contrairement aux deux critères précédents, celui-ci n'a pas à être satisfait dans tous les cas: il existe certaines situations qui font exception à ce critère. Elles seront mentionnées plus loin dans le texte.

4.4.3 Procédure de mise en correspondance

La *mise en correspondance par croissance consistance* procède en deux étapes. Dans un premier temps, des mises en correspondance sont tentées tant que de nouveaux noeuds sont créés à partir des couples de noeuds rencontrant les trois critères décrits à la section précédente. Lorsqu'aucune nouvelle mise en correspondance n'est possible, de nouvelles

tentatives en ce sens sont faites, mais cette fois en ne considérant que les deux premiers critères (le *voisinage* et la *position*). Cette deuxième étape de la procédure, en permettant des appariements entre des zones de types différents, permet de compenser d'éventuelles erreurs de segmentation ou différences de comportement du tracé ayant pour effet de créer une zone supplémentaire ou d'en faire disparaître une.

Puisque la procédure est effectuée en parallèle, il est possible qu'un même noeud du *graphe de relations développé* G_{rd}' soit retenu comme meilleur candidat pour la mise en correspondance avec deux noeuds différents du *graphe de représentation* G_{rep}. Dans un tel cas, un choix doit être fait entre les deux possibilités. Le choix se porte sur la combinaison présentant la moins grande *différence de position spatiale* dp_i [section 4.3.2] entre les deux noeuds.

Tous les couples de noeuds qui peuvent ainsi être formés représentent de nouveaux noeuds dans G_{MEC}. Lorsqu'un nouveau noeud est ainsi créé, il est ajouté à la sous-séquence correspondante, et la procédure de mise en correspondance est reprise en considérant ce nouveau noeud comme limite (*début* ou *fin*) de la sous-séquence concernée. Ceci a pour effet d'allonger les sous-séquences et même éventuellement d'en lier certaines.

L'algorithme du Tableau 4.2 résume l'algorithme utilisé pour la mise en correspondance par croissance consistante. L'algorithme est le même pour les deux étapes de la procédure, sauf pendant la deuxième étape, seuls les deux premiers critères d'ajout sont invoqués (*voisinage* et *position*). La première partie de l'algorithme (lignes 3 à 8) permet d'identifier le meilleur appariement pour chacun des noeuds non appariés du *graphe de relations développé* G_{rel}'. La seconde partie de l'algorithme (lignes 9 à 15) a comme fonction de choisir le meilleur appariement lorsque plus d'une possibilité fait appel à un même noeud du *graphe de représentation* G_{rep}.

Tableau 4.2 Algorithme synthèse de la procédure de mise en correspondance par croissance consistante.

```
1:  POUR chacune des deux étapes (n=3, n=2)
2:      RÉPÉTER
3:          POUR tous les noeuds N_rel non appariés de G_rel'
4:              SI N_rel respecte les n premiers critères ALORS
5:                  Identifier tous les noeuds N_rep compatibles avec N_rel
6:                  Choisir le meilleur appariement, si nécessaire (celui minimisant dp_i)
7:              FIN SI
8:          FIN POUR
9:          POUR tous les appariements possibles
10:             SI plus d'un appariement concerne un noeud N_rel ALORS
11:                 Choisir le meilleur (celui minimisant dp_i)
12:                 Éliminer les autres
13:             FIN SI
14:             Effectuer ajout dans G_MEC
15:         FIN POUR
16:     TANT QUE de nouvelles mises en correspondance sont effectuées
17: FIN POUR
```

Après tous ces traitements, la procédure de mise en correspondance est considérée comme terminée. Le *graphe de mise en correspondance* G_{MEC} rend possible l'identification d'une séquence dans le *graphe de relations développé* G_{rd}'. Cette séquence est directement inspirée de l'information fournie par le *graphe de représentation* G_{rep} qui fait office de référence.

Il est maintenant possible de procéder à la squelettisation de l'image, en se guidant à l'aide du *graphe de mise en correspondance* G_{MEC}. C'est le sujet du prochain chapitre.

5. SQUELETTISATION DE L'IMAGE

5.1 Définition du squelette

La procédure de squelettisation permet de positionner dans l'image le squelette des objets présents dans une image. Cette opération doit permettre de diminuer la quantité d'information nécessaire à la représentation des objets et ainsi faciliter leur identification que ce soit pour en faire la classification ou la reconnaissance.

La formation d'un squelette doit s'effectuer suivant certaines règles. Ces règles sont dérivées de ce que les humains considèrent comme étant les caractéristiques propres à un squelette. Rosenfeld [ROSE75] a proposé les règles suivantes qui correspondent à ces caractéristiques:

1. Le squelette doit être de largeur unitaire;

2. La connexité de l'image doit être conservée;

3. Les courbes, droites et intersections ne doivent pas être déformées.

Le point faible de la majorité des algorithmes de squelettisation correspond à une dérogation à la dernière règle. Les problèmes de déformation peuvent en partie s'expliquer par le caractère *local* et *non informé* des algorithmes. *Local* parce que le traitement d'un pixel se fait en ne considérant que ses huit voisins, et *non informé* parce que le contexte d'application n'est jamais considéré.

Une technique récente permet cependant de pallier cette lacune. Développée par Chouinard et Plamondon [CHOU92], cette nouvelle méthode de squelettisation est spécifiquement adaptée au contexte du traitement des caractères manuscrits. La procédure utilisée dans le présent projet pour la squelettisation est une adaptation de cette méthode.

5.2 Squelettisation de signaux manuscrits

5.2.1 Composition des signaux manuscrits

Les images de signaux manuscrits, de façon générale, sont constituées de traits plus ou moins complexes qui peuvent s'entrecouper de diverses façons. Cette observation permet de définir deux types de régions pour les objets présents dans une image de signaux manuscrits, soit les régions de *tracé isolé* et les régions d'*intersections de traits*.

Les régions de *tracé isolé* sont les éléments de l'objet qui représentent une portion de trait simple. Une définition plus formelle serait: toute portion d'un tracé pour laquelle les deux segments de contours la délimitant sont clairement connus.

Toute portion de l'objet qui ne rencontre pas cette définition peut être considérée comme étant une région d'*intersection de traits*. Lorsque deux traits se superposent, il devient en effet impossible d'identifier les contours appartenant à chacun des traits. Ces portions du tracé manuscrit peuvent être identifiées dans l'image par exclusion: toute portion de l'objet qui n'est pas identifiée comme *tracé isolé* est automatiquement identifiée comme une *intersection de traits*.

5.2.2 Squelettisation adaptée à l'écrit

La méthode proposée par Chouinard et Plamondon adapte justement la squelettisation selon qu'il s'agit de traiter une portion de *tracé isolé* ou une *intersection de trait*. Une autre particularité de la méthode de squelettisation présentée dans [CHOU92] réside dans l'approche séquentielle de l'algorithme plutôt que parallèle comme dans la plupart des solutions proposées antérieurement.

L'algorithme procède en déplaçant, sur les contours du tracé manuscrit, deux pointeurs situés de part et d'autre d'un même trait. Au fur et à mesure du déplacement de ces pointeurs, des points représentant le squelette sont identifiés et reliés entre eux. Cette façon de faire, par suivi de contours, permet la squelettisation des portions de *tracé isolé*.

Les *intersections de traits* sont détectées à l'aide d'une fenêtre de taille variable définie autour des deux pointeurs. Cela permet non seulement la détection mais aussi l'interprétation des *intersections de traits*. Le nombre limité de types possibles d'intersections dans le traçage de caractères manuscrits a permis de définir des intersections « génériques », permettant ainsi la squelettisation sur la base de particularités géométriques précises.

5.3 Squelettisation de la signature manuscrite

5.3.1 Définition de la méthode

L'algorithme de Chouinard et Plamondon est efficace, parce que son champ d'application concerne le traitement des caractères manuscrits. Pour ce qui est des signaux de signatures manuscrites, il serait plutôt difficile de définir toutes les intersections possibles. Par contre,

dans le présent système, les intersections sont déjà localisées et ce, par le biais des zones de discontinuité du *graphe de relations* G_{rel}.

Les intersections étant connues, il est possible de squelettiser les portions de *tracé isolé*. La procédure de suivi de contours peut donc s'appliquer directement pour squelettiser le tracé compris entre deux zones voisines dans le *graphe de mise en correspondance* G_{MEC}. Ce même graphe contient une information de haut niveau concernant la séquence. Cette information rend relativement triviale l'interprétation des intersections, puisque les points d'entrée et de sortie d'une intersection donnée peuvent ainsi facilement être identifiés.

Les points identifiés par les deux procédures précédentes, une fois réunis, constituent le squelette de la signature manuscrite. La réunion des points consiste simplement à former une seule liste de points ordonnés selon la séquence identifiée. Les étapes nécessaires à la réalisation de la squelettisation apparaissent dans le *DFD* présenté à la prochaine figure.

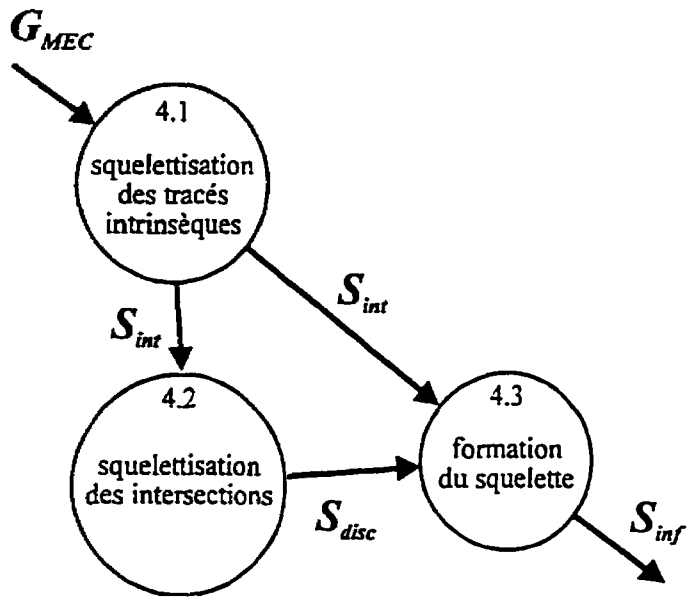


Figure 5.1 *DFD* montrant les étapes associées à la procédure de squelettisation de la signature manuscrite.

5.3.2 Squelettisation du tracé isolé

Toute portion de tracé associée à un arc dans le *graphe de relations* G_{rel} peut être considérée comme une portion de *tracé isolé*. Pour qu'un arc existe entre deux noeuds, il faut que les régions associées aux noeuds soient elles-mêmes reliées par une portion de tracé. Puisqu'une zone peut représenter un *croisement* ou un *extremum*, alors un arc

représente nécessairement une portion d'un trait contribuant à la formation d'un *extremum* ou d'un *croisement*.

La squelettisation du *tracé isolé* consiste donc en la mise en oeuvre de la procédure de suivi de contours entre chacun des noeuds du *graphe de mise en correspondance* G_{MEC} qui sont reliés par un arc. Pour réaliser cette tâche, il est nécessaire d'identifier un point sur chacune des portions de contours du trait à squelettiser. Les deux points ainsi identifiés constituent les positions de départ des pointeurs pour le suivi de contours proprement dit.

Ces deux points sont choisis parmi les points caractéristiques formant la zone de départ. Pour qu'un point soit retenu, il faut que le prochain point caractéristique sur le contour appartienne à la zone d'arrivée. Étant donné la façon dont ont été générées les listes de points des contours [section 2.2.3], pour un des points, il doit s'agir du point caractéristique *suivant* sur le contour et pour le second point, du *précédent* sur le contour.

Lorsque l'arc qui relie la zone de départ et la zone d'arrivée est un *lien simple*, le choix est alors trivial, car seulement deux points rencontrent les critères. Mais dans le cas d'un *lien*

multiple, il est possible que plus de deux points rencontrent les critères de choix. Une priorité doit donc être établie entre les candidats afin d'optimiser le choix.

Si plus d'un point peut être retenu comme point de départ, le choix doit être fait en priorité parmi les points de la zone qui n'ont pas servi de points d'arrivée lors de la squelettisation du tracé entre la zone actuelle et la précédente. Lorsqu'il est impossible de choisir un tel point, alors le choix se fera parmi les points d'arrivée. Un point d'arrivée est un point caractéristique sur lequel s'est terminée la procédure de suivi de contours appliquée à la précédente portion de trait.

Après identification des deux points de départ, les pointeurs doivent être positionnés avant de débuter la procédure de suivi de contours. Si un des points de départ est un point de *discontinuité*, alors le pointeur est déplacé sur le contour en direction de la zone d'arrivée jusqu'à ce qu'il atteigne la limite de la *région de discontinuité* [section 2.4.2] associée au point de discontinuité. La position des pointeurs est ensuite corrigée à nouveau afin de minimiser la distance spatiale qui les sépare. La façon de faire consiste à déplacer vers la zone d'arrivée chacun des pointeurs et de déterminer la position qui minimise la distance. Cette procédure est reprise jusqu'à ce que la distance entre les deux pointeurs atteigne un minimum. La procédure de suivi de contours peut maintenant débuter.

La squelettisation par suivi de contours est une procédure itérative qui consiste à déplacer sur des portions de contours deux pointeurs (*PG* et *PD*) et d'identifier après chaque déplacement un point à ajouter au squelette. Ce point correspond simplement au point milieu de la droite reliant les deux pointeurs. À chaque itération, un pointeur peut être déplacé d'une distance *p* sur le contour. La décision quant au déplacement d'un pointeur est prise en fonction de la distance spatiale entre les deux pointeurs pour chacune des positions possibles [Figure 5.2a].

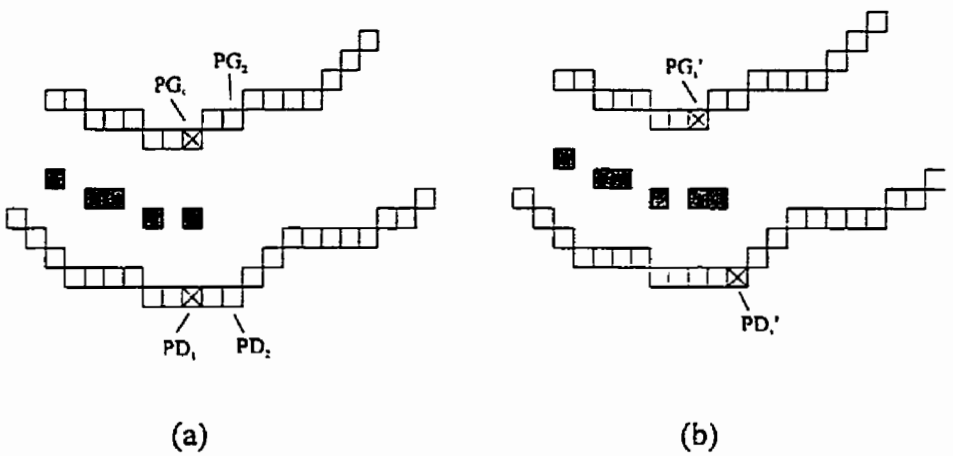


Figure 5.2 Illustration de la procédure de déplacement des pointeurs le long des contours. (a) La position courante est représentée par les pointeurs PG_1 et PD_1. (b) La distance minimum est associée au couple PG_1 et PD_2, qui deviennent la nouvelle position courante, PG_1' et PD_1'.

Puisque chaque pointeur peut ou non être déplacé d'un pas p sur le contour, trois combinaisons doivent être envisagées. La combinaison à retenir est celle offrant la plus petite distance euclidienne d entre les pointeurs. La distance minimum d_{min}, déterminée par:

$$d_{\min} = \min\left[d(PG_1, PD_2), d(PG_2, PD_1), d(PG_2, PD_2)\right] \qquad (5\text{-}1)$$

permet donc d'identifier la position des pointeurs pour la prochaine itération de la procédure, conformément à la prochaine équation:

$$(PG_1', PD_1') = (PG, PD) \mid d(PG, PD) = d_{\min} \qquad (5\text{-}2)$$

La nouvelle position ainsi déterminée, il ne reste qu'à déplacer les pointeurs et à identifier le prochain point à ajouter au squelette [Figure 5.2b].

Le pas p pour le déplacement des pointeurs a été fixé à 2, tel que proposé dans [CHOU90]. De cette façon, il est possible que les points de squelette résultant de deux itérations successives ne soient pas voisins. Cette situation est corrigée en procédant à une

interpolation linéaire entre ces points. Puisque le pas de déplacement est de 2, l'interpolation d'un seul point est suffisante pour joindre les points du squelette.

Certains autres points sont ajoutés au squelette pendant la procédure de squelettisation du *tracé isolé*. Il s'agit du centre de certaines *zones caractéristiques* correspondant à des noeuds du *graphe de mise en correspondance* G_{MEC}. Si un noeud de ce graphe représente un extremum, alors le point représentant le centre de la zone caractéristique associée est ajouté au squelette. Rappelons que le centre d'une zone caractéristique correspond à la moyenne des points caractéristiques formant la zone.

L'ajout de ce point supplémentaire est particulièrement important lorsque l'extremum concerné est le résultat du développement d'une discontinuité [section 4.2]. Le point ainsi ajouté devient un point de contrôle important pour guider l'interpolation lors de la squelettisation des *intersections de traits* [section 5.3.3].

Après cette étape de traitement, la procédure de suivi de contours est terminée. Chacune des étapes décrites jusqu'ici doit être reprise jusqu'à ce que tous les arcs du *graphe de mise en correspondance* G_{MEC} aient été traités. Ce traitement est effectué conformément à l'algorithme présenté au Tableau 5.1. Les résultats de cette procédure sont les portions de

squelette S_{int} représentant les portions de *tracé isolé*. Des exemples de résultats de traitement sont donnés dans la Figure 5.3.

Tableau 5.1 Algorithme utilisé pour la squelettisation par suivi de contours.

```
1: POUR tous les noeuds de G_MEC
2:     Identifier les points de départs
3:     Positionner les pointeurs
4:     TANT QUE les deux pointeurs ne sont pas rendus à la zone voisine FAIRE
5:         Identifier point milieu et l'ajouter au squelette
6:         Déterminer nouvelles position pour les pointeurs
7:     FIN TANT QUE
8: FIN POUR
```

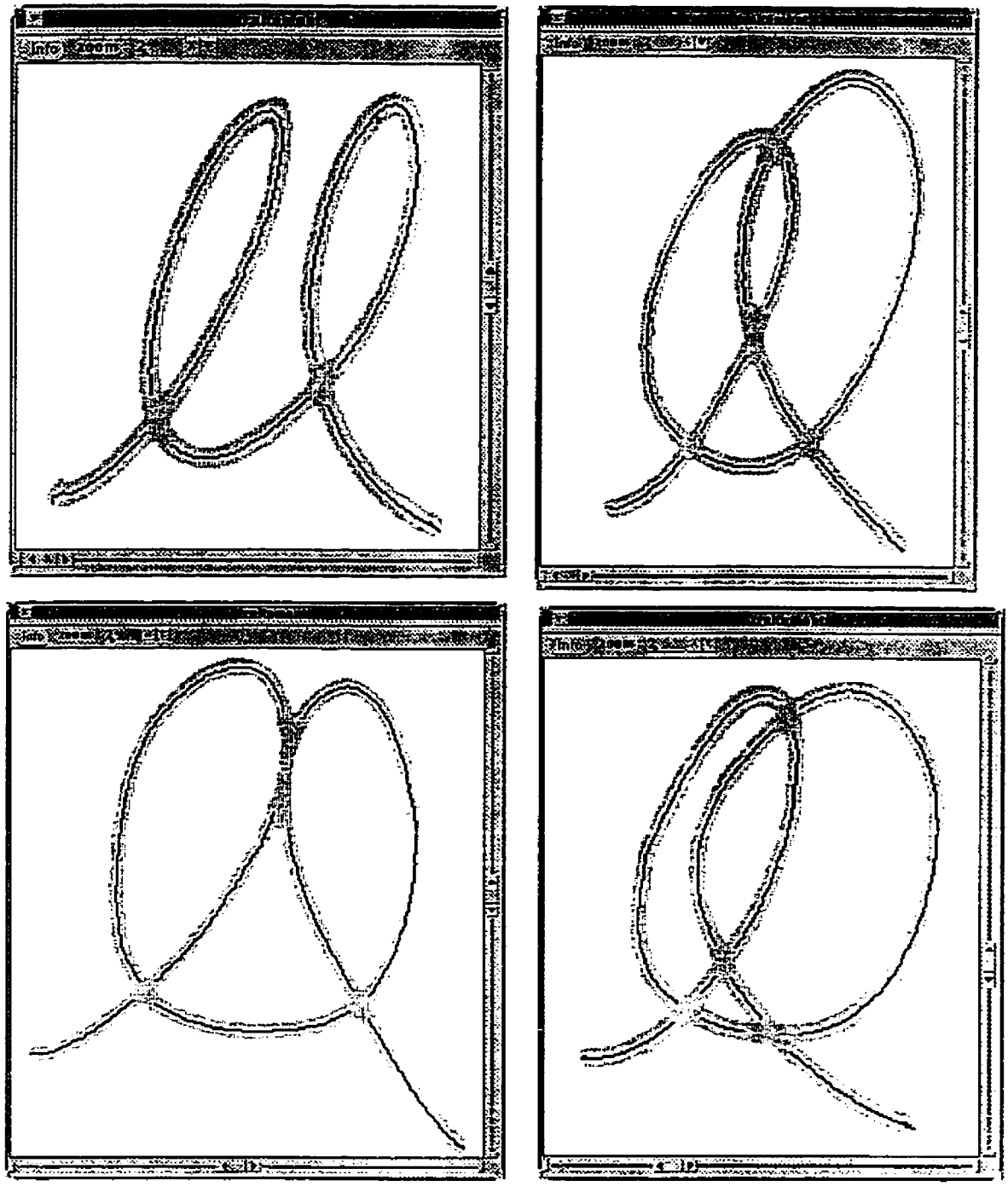


Figure 5.3 Exemples de squelettisation du *tracé isolé* par suivi de contours.

5.3.3 Squelettisation des intersections de traits

Maintenant que toutes les portions de *tracé isolé* sont squelettisées (S_{int}), il ne reste qu'à traiter les *intersections de traits*. Puisque dans une intersection de traits, les contours de chacun des traits ne sont pas entièrement connus, il est difficile d'appliquer la méthode de suivi de contours. Il faut donc s'en remettre à une autre technique.

La solution proposée consiste simplement à interpoler des points pour combler les intervalles dans le squelette correspondant aux régions d'*intersection de traits*. Le choix de la méthode d'interpolation doit être guidé par le contexte d'application.

Parmi les facteurs à considérer, notons premièrement que la nature des intervalles à combler ne permet pas de simplement procéder à une interpolation linéaire. Il est en effet très probable que des croisements se produisent entre des traits de forte courbure. Il est même possible que des intersections occultent des changements d'orientation plus importants, tels que des inversions du sens de rotation.

Ces comportements peuvent être en partie reproduits en procédant à l'interpolation d'une fonction du 3e degré, puisqu'une telle fonction autorise la présence d'un point d'inflexion. Pour interpoler une fonction de degré N, il est nécessaire de connaître $N+1$ points. Dans la

présente situation, cela oblige donc à identifier 4 points à partir des portions de squelette déjà connues.

Un second facteur à considérer est que la position et l'orientation des portions de squelette de part et d'autre de l'intervalle peuvent être telles qu'il soit impossible de définir une fonction, au sens mathématique du terme; il est donc nécessaire de procéder à l'interpolation séparément pour les deux coordonnées, comme c'est le cas lors de l'interpolation de la trajectoire du signal de référence [section 3.2]. Puisque la taille des intervalles à combler peut varier, cette situation pourrait correspondre à un cas d'interpolation entre des données réparties non uniformément. Mais il est impossible de quantifier l'intervalle par rapport à l'axe n, le nouvel axe par rapport auquel sont exprimées les coordonnés x et y.

Une façon de contourner ce problème consiste à choisir les quatre points de contrôle pour l'interpolation en fonction de la taille d de l'intervalle à combler [Figure 5.4 a]. Deux des points de contrôle (P_2 et P_3) sont les points qui terminent les segments à relier, et les deux autres sont ceux situés à une distance d de chacun de ces points. En choisissant ainsi les points de contrôle, l'intervalle entre ceux-ci sur l'axe n peut être considéré constant.

Pour combler l'intervalle représentant la région d'*intersection de traits*, il suffit de procéder à l'interpolation de points entre les points de contrôle P_2 et P_3 [Figure 5.4 b]. La procédure suivie est la même que celle présentée à l'Annexe F, sauf pour ce qui est de la méthode d'interpolation proprement dite.

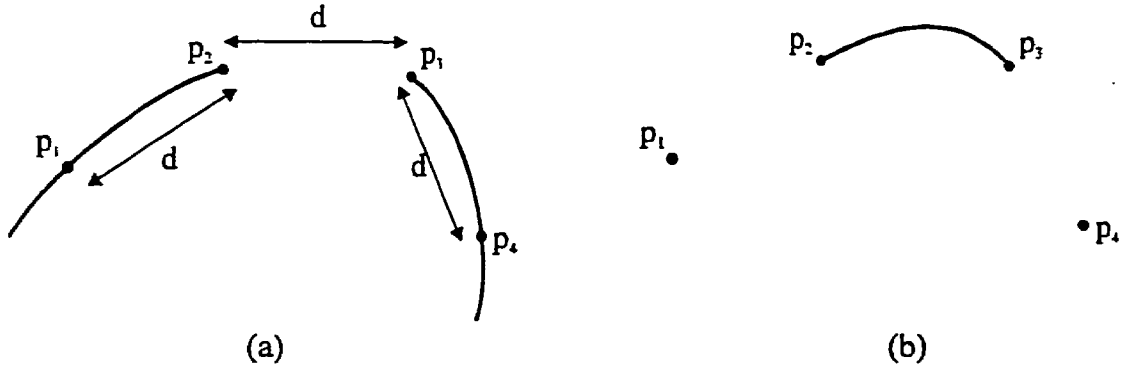


Figure 5.4 Illustration de la procédure d'interpolation pour les intersections de traits. (a) Positionnement des points de contrôle. (b) Interpolation d'un polynôme de degré 3 dans l'intervalle.

La méthode des splines cubiques permet l'interpolation d'une courbe continue entre une série de points de contrôle. Cette courbe est générée à partir de fonctions du 3e degré définies pour chacun des intervalles entre deux points de contrôle successifs. Les conditions de continuité aux points de contrôle sont dictées en fonction des points de contrôle voisins. Dans le cas de l'interpolation de la séquence de référence, cette technique

était tout indiquée puisque les échantillons de la tablette à digitaliser représentaient des points de contrôle répartis uniformément dans le temps.

La situation actuelle est tout autre. Il s'agit ici de combler un seul intervalle situé entre deux portions de squelette. La taille et la nature de cet intervalle dépendent de la *discontinuité* qui doit être squelettisée. Comme il est difficile d'identifier les points de contrôle nécessaires à l'utilisation des splines cubiques et qu'il n'y a qu'un seul intervalle à combler, il est plus approprié d'utiliser ici l'équation de Lagrange pour un polynôme du 3e degré.

Partant de l'affirmation qu'il existe une seule fonction de degré *N-1* pouvant passer par *N* points, l'équation de Lagrange permet d'identifier un polynôme unique pour un ensemble donné de points de contrôle. L'équation de Lagrange pour un polynôme de degré N-1 est la suivante:

$$P(x) = \frac{(x-x_2)(x-x_3)\ldots(x-x_N)}{(x_1-x_2)(x_1-x_3)\ldots(x_1-x_N)}y_1 + \frac{(x-x_1)(x-x_3)\ldots(x-x_N)}{(x_2-x_1)(x_2-x_3)\ldots(x_2-x_N)}y_2 + \ldots + \frac{(x-x_1)(x-x_2)\ldots(x-x_{N-1})}{(x_N-x_1)(x_N-x_2)\ldots(x_N-x_{N-1})}y_N \quad (5\text{-}3)$$

Cette équation est construite de telle sorte que chacun des membres de droite est un polynôme de degré N-1. La valeur de cette équation aux N points x_i, représentant les points de contrôle, est égale au y_i correspondant. Pour ce qui est des valeurs de x intermédiaire, elles sont déterminées par une combinaison linéaire de tous les polynômes. Des exemples d'interpolation sont donnés dans la figure qui suit. Il serait bon de mentionner que les interpolations de la prochaine figure résultent toutes d'une comparaison avec le signal dynamique présenté à la Figure 3.3.

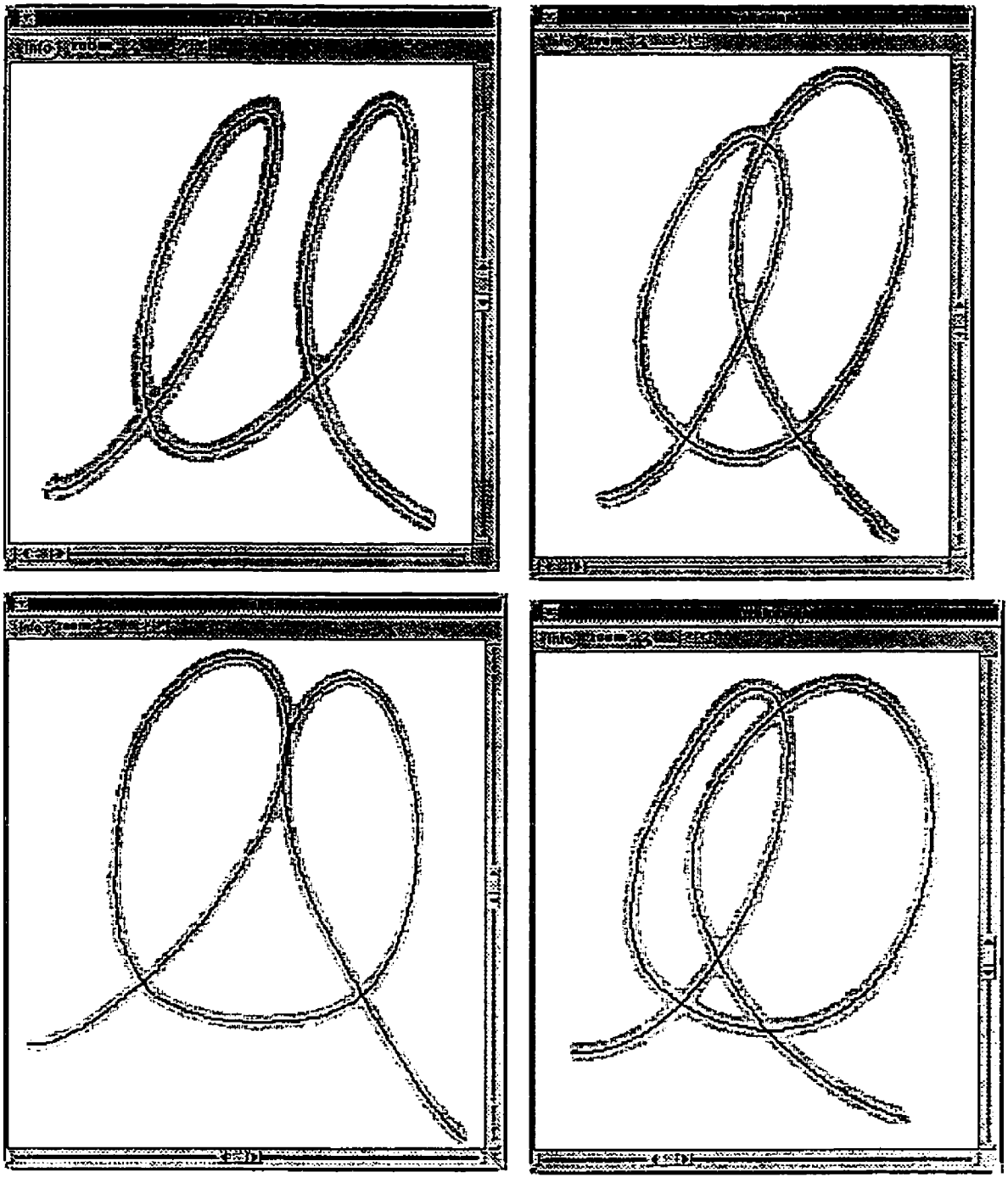


Figure 5.5 Exemples de squelettisation des *intersections de traits* par interpolation.

Une fois tous les intervalles comblés, il suffit de regrouper en un seul squelette les points identifiés par interpolation (S_{dist}) à ceux identifiés par le suivi de contours (S_{int}) [section 5.3.2]. Le résultat de cette opération constitue la *séquence inférée* S_{inf}. Cette liste de points représente la séquence inférée dans l'*image de luminance* I_e à partir de l'information *a priori* contenue dans la *dynamique de référence* D_e. Il s'agit là du résultat final de la procédure d'inférence de la séquence.

Puisque cela termine la description de la méthode d'inférence proposée, il est maintenant temps de procéder à l'évaluation de cette nouvelle technique.

6. ÉVALUATION DES PERFORMANCES

6.1 Composition et acquisition de la banque de données

Les performances du système proposé ont été évaluées à l'aide d'une banque de données différente de celle utilisée pendant la phase de développement. Cette façon de faire permet une évaluation sans biais des performances.

La banque de données utilisée pour la validation est composée des signatures de 20 scripteurs, chacun ayant produit 20 spécimens de sa signature. Chaque spécimen de signature est emmagasiné sous deux formes de représentation : le signal statique $\mathbf{I_e}$ (image digitalisée) et le signal dynamique $\mathbf{D_e}$ (points échantillonnés). Une moitié de la banque est consacrée aux signaux de *référence* et la seconde moitié, aux signaux de *test*. Une représentation schématisée de la banque de données est montrée à la Figure 6.1. L'Annexe G montre des signatures sous les deux représentations utilisées, à raison d'une signature par scripteur.

Afin de minimiser les effets de la fatigue, les signatures produites par chaque scripteur ont été réalisées en quatre blocs de cinq signatures, avec une période de repos d'environ 5 minutes entre chaque bloc. Les signatures ont été produites sur des feuilles de papier blanc, fixées sur la tablette à digitaliser, à raison de cinq spécimens par feuille.

Le scripteur était assis confortablement et jouissait de toutes les libertés pour produire les spécimens de sa signature. Aucune contrainte n'était imposée quant à sa position, la hauteur de la chaise ou la position de la tablette à digitaliser.

La taille des signatures a été limitée par un rectangle de 3 cm × 12 cm. Cette limite était suggérée plutôt qu'imposée au scripteur: une feuille gabarit, sur laquelle étaient imprimés cinq rectangles de la dimension spécifiée, a été placée sous celle servant à recueillir les images des signatures.

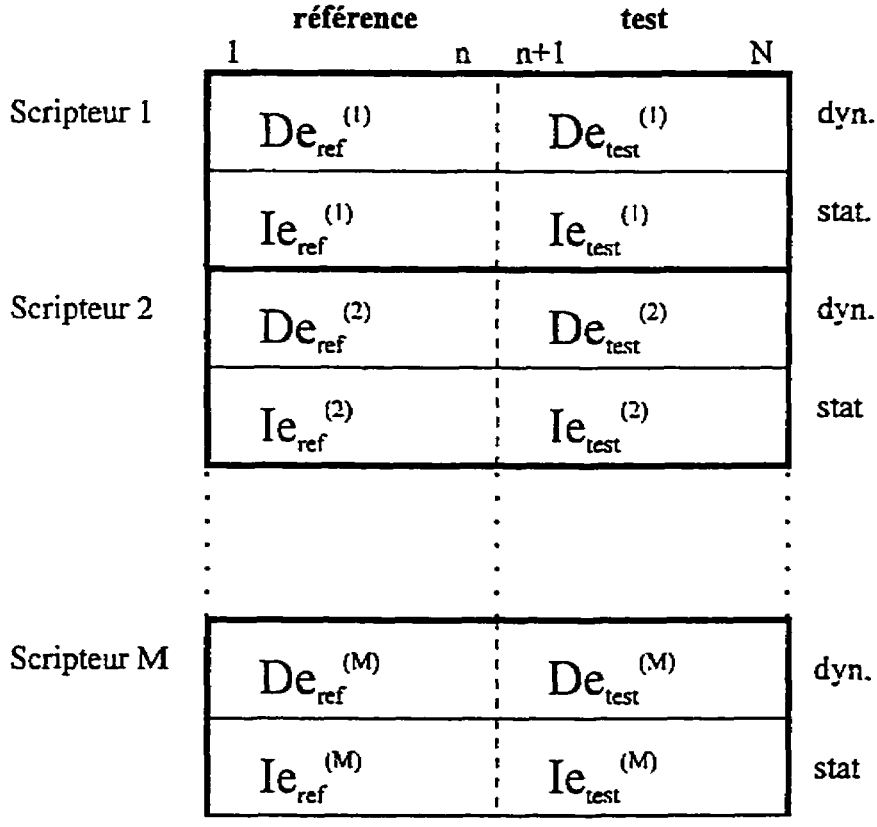


Figure 6.1 Représentation schématique de la banque de données.

Pour chaque signature manuscrite, deux fichiers ont été créés, soit un pour l'image et un pour la séquence correspondante. Le fichier image est le résultat de la digitalisation de l'image produite sur la feuille de papier. Cette opération est effectuée à l'aide d'un digitaliseur *Hewlett-Packard*, à une résolution de 300 *ppp*. Pour ce qui est du fichier séquence, il est constitué des points échantillonnés par la tablette. Cet échantillonnage s'est fait à l'aide d'une tablette *Wacom UD-1212*, à une fréquence de 100 *Hz* et avec une résolution de 1000 *ppp*. L'outil utilisé pour produire les signatures fut un stylo instrumenté *Wacom Up-401* muni d'une pointe à bille à encre noir.

6.2 Métriques définies pour l'évaluation des performances

Dans le but de quantifier les performances du système proposé et de permettre une analyse plus détaillée des résultats du processus d'inférence de la séquence, certaines mesures, ou métriques, sont proposées. L'information nécessaire à l'application des mesures est constituée par :

1. le graphe de mise en correspondance G_{MEC};
2. la séquence inférée S_{inf};
3. le graphe de représentation G_{rep};
4. la séquence originale D_{int}.

Les métriques proposées permettent de comparer les résultats du processus de squelettisation (G_{MEC} et S_{inf}) avec les informations de référence utilisées pour la mise en oeuvre de ce même processus (G_{rep} et D_{int}).

La première mesure, ΔL, correspond à *la proportion de séquence non considérée*. Cette mesure est dite globale puisqu'elle dépend de la longueur totale de la séquence de référence D_{int} et de la portion de cette séquence pour laquelle il n'y a pas eu de mise en correspondance. Une portion de séquence a été mise en correspondance s'il existe dans le graphe de mise en correspondance G_{MEC} un arc entre les noeuds délimitant cette portion de tracé (ce qui suppose aussi l'existence de ce noeud dans G_{MEC}).

Ainsi, une fonction booléenne, $arc(Z_1,Z_2)$ peut être définie de façon à prendre la valeur *VRAI* seulement lorsque les noeuds Z_1 et Z_2 existent dans G_{MEC} et sont reliés par un arc. Donc, tous les segments pour lesquels cette fonction prend la valeur *FAUX* constituent la portion de séquence de référence pour laquelle il n'y a pas eu de mise en correspondance.

La valeur de ΔL correspond donc à la longueur totale des segments de référence, $l_{squel}(Zd_i,Zd_{i+1})$, compris entre chaque paire de noeuds voisins (Zr_i,Zr_{i+1}) pour lesquels la fonction $arc(Zd_i,Zd_{i+1})$ est fausse, divisée par la longueur totale de la séquence de référence.

$$\Delta L = \frac{\sum_{i=1}^{N-1}\left[\, l_{squel}(Zd_i, Zd_{i+1}) \;\middle|\; \overline{arc(Zd_i, Zd_{i+1})} \,\right]}{l_{squel}(Zd_1, Zd_N)} \qquad (6\text{-}1)$$

$$0.0 \leq \Delta L \leq 1.0$$

La seconde mesure, Δl_i, concerne *la différence de longueur relative* des segments. Il s'agit d'une mesure à caractère local puisqu'elle se calcule pour chaque segment de la séquence de référence pour lesquels il y a eu mise en correspondance. Pour chaque segment (i) compris entre les noeuds Zs_i et Zs_{i+1}, une différence de longueur relative Δl_i est calculée par rapport à son équivalent compris entre les noeuds Zd_i et Zd_{i+1}. Cette mesure correspond à la différence de longueur en pixel entre les deux segments, normalisée par la longueur totale de la séquence de référence.

$$\Delta l_i = \frac{\left| l_{squel}(Zs_i, Zs_{i+1}) - l_{squel}(Zd_i, Zd_{i+1}) \right|}{l_{squel}(Zd_1, Zd_N)} \qquad (6\text{-}2)$$

Si les deux segments représentent la même information, ils doivent avoir la même longueur. De petites différences de taille peuvent cependant s'expliquer par des erreurs d'interprétation menant à un positionnement spatial différent pour des zones représentant la même information. Dans ce cas, les différences de longueur sont de l'ordre des différences de positions des zones concernées.

Une *différence de longueur moyenne* Δl_{tot} peut être estimée à partir d'une simple somme pondérée des différentes valeurs de Δl_i mesurées pour une séquence donnée. Le facteur de pondération utilisé correspond à la taille du segment concerné par rapport à la séquence totale. Mathématiquement, cela correspond à l'équation qui suit:

$$\Delta l_{tot} = \frac{\sum_{i=1}^{N} \Delta l_i \times l_{squel}(Zd_i, Zd_{i+1})}{l_{squel}(Zd_1, Zd_N)} \qquad (6\text{-}3)$$

Ainsi combinées, les différences de longueurs relatives Δl_i caractérisent globalement le résultat de la mise en correspondance. Cette nouvelle mesure ne représente pas pour autant une mesure de type global. Le résultat peut donc servir de facteur de comparaison entre les résultats de deux mises en correspondance.

La dernière mesure, Δp_i, traduit la *différence de position relative* des segments. Cette différence de position est estimée à l'aide de la position des centres de masse de chacun des segments. Tout comme pour Δl_i, il s'agit d'une mesure locale.

La position de ces centres de masse (cm_X et cm_Y) est calculée en accordant un poids identique à chacun des pixels composant un segment. Comme il s'agit de comparer des positions exprimées dans deux espaces différents, il faut les normaliser afin de rendre la comparaison significative. Les coefficients de normalisation utilisés sont les dimensions

des rectangles délimitant chacun des signaux (d_{Xs} et d_{Ys} pour la séquence inférée, d_{Xd} et d_{Yd} pour la séquence de référence). La *différence de position relative* Δp_i entre deux segments se calcule donc comme suit:

$$\Delta p_i = \sqrt{\left(\frac{cm_X(Zs_i, Zs_{i+1})}{d_{Xs}} - \frac{cm_X(Zd_i, Zd_{i+1})}{d_{Xd}}\right)^2 + \left(\frac{cm_Y(Zs_i, Zs_{i+1})}{d_{Ys}} - \frac{cm_Y(Zd_i, Zd_{i+1})}{d_{Yd}}\right)^2} \quad (6\text{-}4)$$

Il est possible que deux segments ne représentant pas la même information aient des tailles identiques, mais la position de leur centre de masse alors diffère. Cette mesure permet donc d'ajouter une dimension supplémentaire à la mesure de la qualité de la mise en correspondance pour chacun des segments composant le graphe G_{MEC}.

Pour les mêmes raisons il est possible de définir une *différence de position moyenne* Δp_{tot}. Cette différence moyenne est estimée en utilisant la procédure de pondération illustrée par l'équation suivante:

$$\Delta p_{tot} = \frac{\sum_{i=1}^{N} \Delta p_i \times l_{squel}(Zd_i, Zd_{i+1})}{l_{squel}(Zd_1, Zd_N)} \quad (6\text{-}5)$$

La procédure utilisée pour la mise en correspondance limite l'appariement entre des segments représentant des portions de tracé délimitées par des couples de zones semblables. Cette restriction assure que la mise en correspondance se fait seulement entre

portions de tracé représentant une information qui, à défaut d'être identique, est de même nature. Les mesures de distance du type de celles décrites ici, particulièrement les mesures locales, ont donc comme rôle de quantifier l'exactitude ou l'authenticité de la mise en correspondance.

Les mesures définies précédemment s'apparentent toutes trois à des distances. Puisqu'il s'agit dans tous les cas de distances normalisées, il est possible de les combiner de différentes façons afin d'en tirer un maximum d'information, ou de maximiser leur pouvoir de discrimination, comme il sera montré plus loin dans ce texte.

6.3 Étapes d'évaluation des performances

L'évaluation des performances doit se faire en tenant compte des objectifs visés par le projet. Puisqu'il est ici question d'inférence de la séquence, les premières étapes de validation portent justement sur cet aspect du projet.

Étant donné que la séquence inférée dans une image donnée est pertinente seulement lorsque l'image est une signature authentique (par rapport à la référence utilisée), les deux premières expériences se font donc en comparant seulement des signatures authentiques entre elles. Il est alors question de l'évaluation de l'efficacité du système.

Dans la première expérience, chaque image de la partie *référence* de la banque de données est comparée avec son équivalent dynamique ($\mathbf{Ie_i^{(m)}}$ avec $\mathbf{De_i^{(m)}}$, $\mathbf{1 \leq i \leq n}$). Puisqu'il s'agit de la comparaison d'*originaux* (les deux représentations d'une même signature), les performances de la procédure de mise en correspondance sont considérées optimales. Cette expérience permet, en plus de l'identification des forces et faiblesses de la procédure de mise en correspondance, d'évaluer l'exactitude de la procédure de squelettisation.

Dans la seconde expérience, les images sont comparées avec des séquences de référence autres que celles à laquelle elles correspondent ($\mathbf{Ie_i^{(m)}}$ avec $\mathbf{De_j^{(m)}}$, $\mathbf{1 \leq i,j \leq n}$, $\mathbf{i \neq j}$). Comme pour la première expérience, l'information provient uniquement de la partie *référence* de la banque de données. Il est alors question de comparaison entre des *authentiques*. Cette seconde expérience permet de pousser plus loin l'évaluation des performances de la procédure de mise en correspondance, en vérifiant sa capacité d'inférence lorsque la référence utilisée ne correspond pas à l'image de test.

Si les résultats de la seconde expérience sont concluants, une troisième expérience est envisagée: il s'agit cette fois de comparer des signaux de références et des images provenant de scripteurs différents. Les images jouent alors un rôle de *faux aléatoires*. Le but de cette troisième expérience est justement de vérifier la *capacité de détection des faux aléatoires* que présente le système. Les différents seuils qui permettent de décider s'il s'agit ou non d'un faux aléatoire sont déterminés en utilisant la partie *référence* de la

banque de données, et les performances en classification sont ensuite évaluées avec la partie *test* de cette même banque.

6.4 Évaluation de l'efficacité

6.4.1 Protocole expérimental

L'efficacité est évaluée par deux expériences différentes. Dans chacune des deux expériences, seule la portion *référence* de la banque de données est utilisée. Dans la première expérience, chaque signature de l'ensemble $\mathbf{Ie}_{ref}^{(m)}$ est comparée avec la séquence originale correspondante, pour un total de 200 comparaisons, alors que dans la seconde expérience, trois signatures de l'ensemble $\mathbf{Ie}_{ref}^{(m)}$ sont comparées avec trois éléments de $\mathbf{De}_{ref}^{(m)}$ en prenant bien soin d'éviter de comparer des originaux entre eux, pour un total de 180 comparaisons.

C'est à partir de ces informations que sera validée la méthode proposée. Les tableaux qui suivent résument les procédures à suivre pour effectuer les expériences.

Tableau 6.1 Protocole d'évaluation de l'efficacité: comparaison de signaux originaux.

1:	**POUR** tous les scripteurs (m)
2:	**POUR** toutes les signatures $\mathbf{Ie}_i \in \mathbf{Ie}_{ref}^{(m)}$ $(1 \leq i \leq 10)$
3:	Générer G_{rep} correspondant à $\mathbf{De}_i$
4:	Générer G_{rel}' correspondant à $\mathbf{Ie}_i$
5:	Créer G_{MEC} (mise en correspondance de G_{rep} et G_{rel}')
6:	Inférer la séquence S_{inf}
7:	**FIN POUR**
8:	**FIN POUR**

Tableau 6.2 Protocole d'évaluation de l'efficacité: comparaison de signaux authentiques.

1.	**POUR** tous les scripteurs (m)
2:	Choisir aléatoirement 3 signatures $\mathbf{Ie}_i \in \mathbf{Ie}_{ref}^{(m)}$ $(1 \leq i \leq 10)$
3:	**POUR** les 3 signatures $\mathbf{Ie}_i$
4:	Choisir aléatoirement 3 signatures $\mathbf{De}_j \in \mathbf{De}_{ref}^{(m)}$ tel que $j \neq i$ $(1 \leq j \leq 10)$
5:	**POUR** les signatures $\mathbf{De}_j$
6:	Générer G_{rep} correspondant à $\mathbf{De}_j$
7:	Générer G_{rel}' correspondant à $\mathbf{Ie}_i$
8:	Créer G_{MEC} (mise en correspondance de G_{rep} et G_{rel}')
9:	Inférer la séquence S_{inf}
10:	**FIN POUR**
11:	**FIN POUR**
12:	**FIN POUR**

6.4.2 Critères de validation

Les deux expériences décrites ici ayant été effectuées avec des signatures appartenant à la classe des authentiques seulement, la validation concerne la capacité d'inférence de la séquence qu'offre le système proposé.

Le premier critère utilisé pour la validation est la *proportion de séquence non considérée* ΔL. Cette mesure de distance sert de premier estimateur de la performance du système: plus la distance est petite, plus la mise en correspondance est efficace. Idéalement, la valeur de ΔL devrait être nulle pour les comparaisons de la première expérience, puisque les signaux comparés représentent la même information.

Dans la seconde expérience, la mesure de distance ΔL va permettre d'établir les limites du système. Puisque les signatures utilisées sont toutes des authentiques et que, contrairement à la première expérience, les images ne sont pas comparées avec les séquences originales, la mesure de ΔL permet d'évaluer la performance du système dans une situation réelle. L'expérience permet de déterminer, entre autres, le pourcentage maximum de séquence qui peut être ignoré lors d'une comparaison entre une image et une dynamique authentiques. Les valeurs espérées pour cette mesure sont évidemment plus élevées que celles de la première expérience.

La simple analyse de la valeur de ΔL n'est cependant pas suffisante pour la validation de la méthode proposée. Il est en effet possible que des portions de séquence mises en correspondance ne correspondent pas vraiment à la même information. Afin de vérifier si tel est le cas, les mesures de différence de longueur Δl_i et de différence de position Δp_i sont utilisées. Si la portion de séquence de référence et la portion correspondante de squelette inférée représentent la même information, alors les valeurs de Δl_i et Δp_i doivent tendre vers zéro. Ces deux distances devraient prendre des valeurs plus ou moins grandes dans les cas de mauvais appariements. Une analyse détaillée de ces valeurs devrait permettre de statuer sur la fiabilité de ces deux mesures ainsi que sur l'importance qui peut leur être accordée.

6.5 Évaluation de la capacité de détection des faux aléatoires

6.5.1 Protocole expérimental

Cette troisième expérience a pour but de vérifier si le système proposé peut être utilisé comme première étape de vérification pour l'élimination rapide des *faux aléatoires*. La première partie de cette expérience, nommée phase d'*apprentissage*, a comme objectif de vérifier la capacité discriminante des métriques proposées. Si tel est le cas, il sera alors possible de procéder à une seconde expérience, la phase de *généralisation*, ayant pour but la quantification des performances du système pour la détection des *faux aléatoires*.

Les deux phases de l'expérience se déroulent sensiblement de la même façon. Dans les deux cas, deux ensembles de signaux sont constitués pour chaque scripteur (i): un ensemble de *test* $E_{test}^{(i)}$ et un de *référence* $E_{ref}^{(i)}$. Les ensembles de *référence* sont, pour les deux phases de l'expérience, tous constitués de trois signaux dynamiques *authentiques* choisis aléatoirement dans la partie *référence* de la banque de données. Le terme *authentique* signifie que les signaux dynamiques sont choisis parmi ceux produits par le scripteur concerné. Les ensembles de *test*, eux, sont constitués d'images *authentiques* ainsi que d'images représentant des *faux aléatoires*. Ces dernières images sont choisies parmi celles produites par les scripteurs autres que celui concerné, à raison d'une signature par scripteur.

Pour les ensembles de la phase d'*apprentissage*, les images sont choisies dans la partie *référence* de la banque de données, tandis que pour la phase de *généralisation*, elles sont choisies dans la partie *test* de cette même banque. Étant donné que trois signaux dynamiques de la partie référence constituent l'ensemble de référence, il est inutile de retenir dans l'ensemble de test les images *originales* de ces signaux. Le nombre d'authentiques dans les ensembles de *test* en *apprentissage* est donc de 7, ce nombre étant de 10 pour les ensembles de *test* en *généralisation*. Le tableau qui suit résume la composition des différents ensembles pour les deux phases de l'expérience.

Tableau 6.3 Tableau synthèse résumant la composition des différents ensembles de signatures utilisés pour l'évaluation des performances en identification des faux aléatoires.

	apprentissage		généralisation	
$E_{ref}^{(i)}$	$3 \times De_n^{(i)}$	$1 \leq n \leq 10$	$3 \times De_n^{(i)}$	$1 \leq n \leq 10$
$E_{test}^{(i)}$	$7 \times Ie_m^{(i)}$	$1 \leq m \leq 10$, $m \neq n$	$10 \times Ie_m^{(i)}$	$11 \leq m \leq 20$, $\forall m$
	$1 \times Ie_p^{(j)}$	$1 \leq p \leq 10$, $1 \leq j \leq 20$, $\forall j$, $j \neq i$	$1 \times Ie_p^{(j)}$	$11 \leq p \leq 20$, $1 \leq j \leq 20$, $\forall j$, $j \neq i$

Chacune des images de l'ensemble de *test* $\mathbf{E}_{test}^{(i)}$ est comparée avec chacun des signaux dynamiques de l'ensemble de *référence* $\mathbf{E}_{ref}^{(i)}$. Le résultat de chacune des comparaisons permet d'estimer une mesure de distance ζ_{MEC}, traduisant la différence entre la séquence de référence et la séquence inférée. Des trois comparaisons associées à une image, seule celle donnant la distance minimum est considérée. C'est cette distance minimum qui servira de point de comparaison avec les autres images. La méthode de calcul de la distance ζ_{MEC} sera discutée dans la prochaine section.

Dans l'ensemble, les deux phases de l'expérience se déroulent sensiblement de la même façon; seul le traitement des résultats des comparaisons diffère. Les résultats de la phase d'*apprentissage* servent à déterminer si la mesure de distance est discriminante et, le cas échéant, à déterminer les différents seuils de classification, tandis que les résultats de la

phase de généralisation servent à déterminer les performances du système quant à l'identification des faux aléatoires. Les deux algorithmes utilisés pour la conduite des expériences sont résumés dans les deux tableaux suivants. Les numéros de lignes en caractères **gras** indiquent les étapes qui diffèrent d'un algorithme à l'autre.

Tableau 6.4 Algorithme pour l'évaluation des performances du système pour la discrimination des faux aléatoires: phase d'*apprentissage*.

1:	**POUR** tous les scripteurs (i)
2:	Construire l'ensemble $\mathbf{E}_{ref}^{(i)}$
3:	Construire l'ensemble $\mathbf{E}_{test}^{(i)}$
4:	**POUR** toutes les signatures $\mathbf{I}_e \in \mathbf{E}_{test}^{(i)}$
5:	Calculer la distance de similarité ζ_{MEC} entre $\mathbf{I}_e$ et chaque élément de $\mathbf{E}_{ref}^{(i)}$
6:	Retenir la distance ζ_{MEC} minimum
7:	**FIN POUR**
8:	Déterminer les seuils $\tau_I^{(i)}$ et $\tau_{tot}^{(i)}$
9:	**FIN POUR**

Tableau 6.5 Algorithme pour l'évaluation des performances du système pour la discrimination des faux aléatoires: phase de *généralisation*.

1:	**POUR** tous les scripteurs (i)
2:	Construire l'ensemble $\mathbf{E}_{ref}^{(i)}$
3:	Construire l'ensemble $\mathbf{E}_{test}^{(i)}$
4:	**POUR** toutes les signatures $\mathbf{I}_e \in \mathbf{E}_{test}^{(i)}$
5:	Calculer la distance de similarité ζ_{MEC} entre $\mathbf{I}_e$ et chaque élément de $\mathbf{E}_{ref}^{(i)}$
6:	Retenir la distance ζ_{MEC} minimum
7:	**POUR** chacun des seuils $\tau^{(i)}$ ($\tau_I^{(i)}$ et $\tau_{moy}^{(i)}$)
8:	**SI** $\zeta_{MEC} \geq \tau^{(i)} \wedge \mathbf{I}_e \in \omega_1$ **ALORS** Erreur type 1
9:	**SINON SI** $\zeta_{MEC} < \tau^{(i)} \wedge \mathbf{I}_e \in \omega_2$ **ALORS** Erreur type 2
10:	**SINON** Bonne classification
11:	**FIN POUR**
12:	**FIN POUR**
13:	Calculer les erreurs ε_1 , ε_2 et ε_{moy}
14:	**FIN POUR**

Pour ce qui est de la phase de *généralisation*, la classification se fait en comparant la *distance de mise en correspondance* ζ_{MEC} associée à une signature avec le seuil $\tau_{MEC}^{(i)}$ approprié. La règle de classification est la suivante :

$$\text{SI } \zeta_{MEC} < \tau_{MEC}^{(i)} \quad \text{ALORS} \quad \mathbf{I}_e \in \omega_1$$

$$\text{SINON} \quad \mathbf{I}_e \in \omega_2$$

Le nombre de signatures mal classées permet de calculer les erreurs ε_1 et ε_2. Une signature est mal classée si elle est considérée comme un *faux aléatoire* (classe ω_2), alors qu'elle est en réalité une *authentique* (erreur de type 1, ε_1), ou si elle est considérée comme une *authentique* (classe ω_1), alors qu'il s'agit d'un *faux aléatoire* (erreur de type 2, ε_2). L'erreur moyenne ε_{moy} se calcule comme la moyenne des erreurs ε_1 et ε_2.

6.5.2 Critères de validation

Tel que mentionné dans la section précédente, les résultats de la mise en correspondance entre une image $\mathbf{I_e}$ et une dynamique $\mathbf{D_e}$ sont comparés sur la base d'une mesure de distance ζ_{MEC}. Cette mesure de distance est élaborée à partir des différentes métriques présentées plus avant dans ce texte [section 6.2]. Les mesures de *proportion de séquence non considérée* ΔL, de *différence de longueur moyenne* Δl_{tot} et de *différence de position moyenne* Δp_{tot} sont combinées de la façon suivante:

$$\zeta_{MEC} = \beta\left[\alpha\,\Delta t_{tot} + (1-\alpha)\Delta p_{tot}\right] + (1-\beta)\Delta L \qquad (6\text{-}6)$$
$$0.0 \le \alpha, \beta \le 1.0$$

Il s'agit d'une somme pondérée des trois mesures de distance préalablement définies. La construction de l'équation est telle que chacun des facteurs de pondération joue un rôle distinct. Le facteur α permet de déterminer le poids relatif de chacune des mesures d'origine locale (Δl_{tot} et Δp_{tot}), tandis que le facteur β pondère l'importance de ces mesures

par rapport à la mesure strictement globale qu'est ΔL. Il faut cependant déterminer les valeurs que doivent prendre chacun de ces facteurs, et surtout comment procéder au choix de ces valeurs.

La qualité recherchée pour une mesure servant à la classification est sa capacité discriminante: permet-elle de bien distinguer les éléments appartenant à des classes différentes? La vérification des signatures étant essentiellement un problème à deux classes (ω_1: les authentiques, ω_2: les faux), une mesure de distance unidimensionnelle comme celle retenue ici est suffisante pour permettre cette distinction. Comme il s'agit d'une mesure de dissimilarité, les signatures de la classe ω_1 doivent présenter des valeurs de ζ_{MEC} plus petites que celles de la classe ω_2. Plus la différence entre les mesures de distance associées à ces deux classes est grande, plus la mesure est discriminante. Idéalement, si la mesure est suffisamment discriminante, les résultats forment deux groupes distincts comme le montre la Figure 6.2.

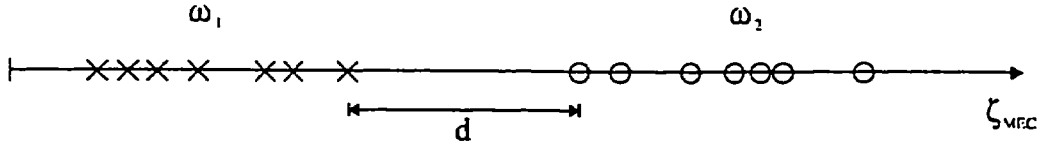


Figure 6.2 Illustration du concept de *séparabilité* des classes.

Les facteurs de pondération α et β utilisés dans la mesure de distance ζ_{MEC} doivent donc être choisis de manière à maximiser la *séparabilité* des classes. Les valeurs retenues sont celles qui maximisent la distance *d* séparant les deux classes [Figure 6.2]. Le processus d'optimisation consiste donc à faire varier les valeurs de α et β sur la plage 0 à 1 et à évaluer la distance *d* correspondante. Les valeurs α_{opti} et β_{opti} retenues correspondent au *d* maximum. Cette optimisation est effectuée pour chaque scripteur en utilisant les ensembles d'apprentissage correspondants. Les valeurs ainsi obtenues vont servir aux calculs de distance tant pour la phase d'*apprentissage* que pour la phase de *généralisation*. La figure qui suit [Figure 6.3] montre les résultats de la procédure d'optimisation pour les 20 scripteurs.

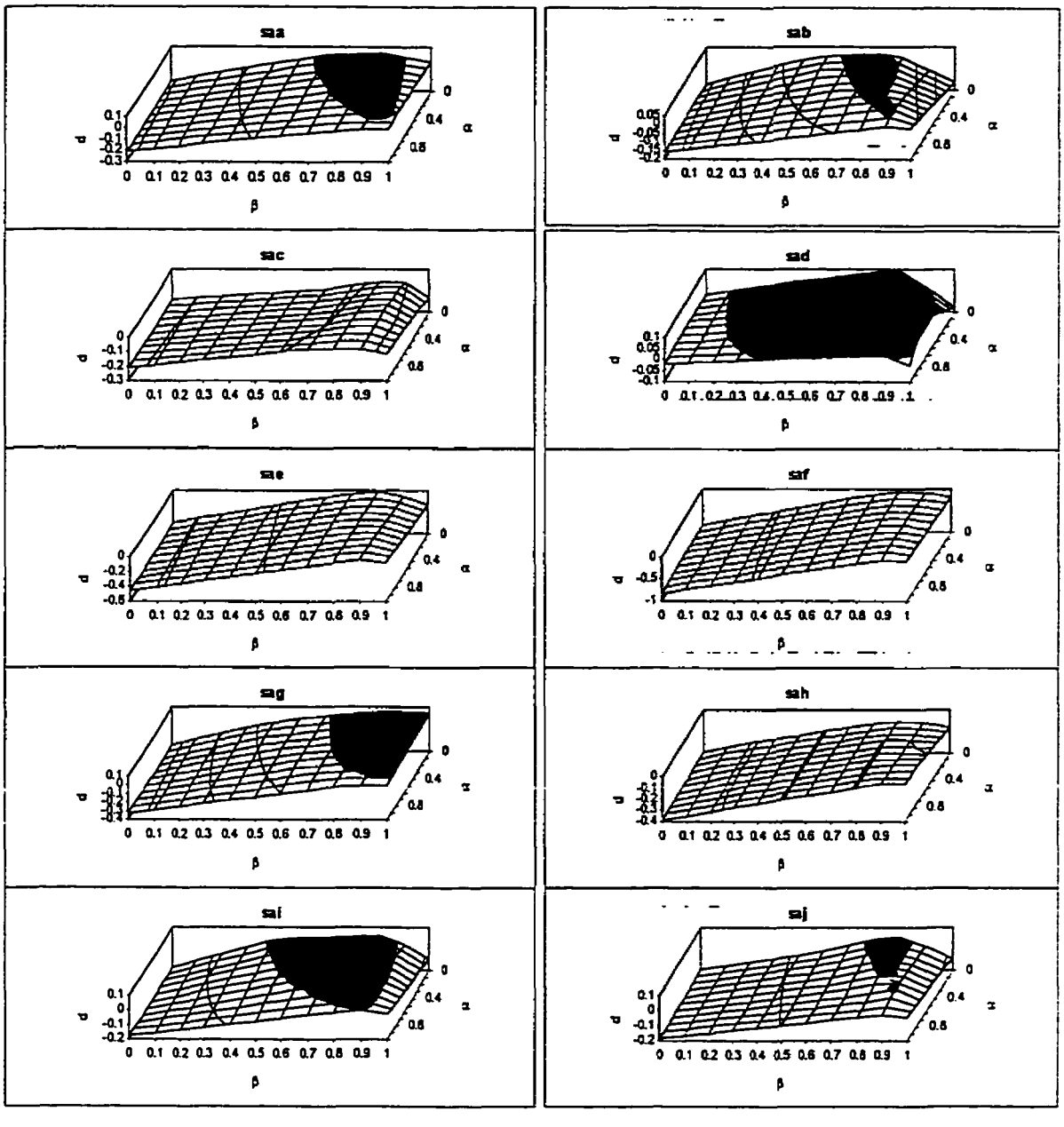


Figure 6.3 Graphiques illustrant, pour chacun des 20 scripteurs, la valeur de *d* en fonction des facteurs α et β. Les zones foncées, représentant des valeurs de *d* positives, indiquent les valeurs de α et β pour lesquelles les deux classes sont complètement *séparées*.

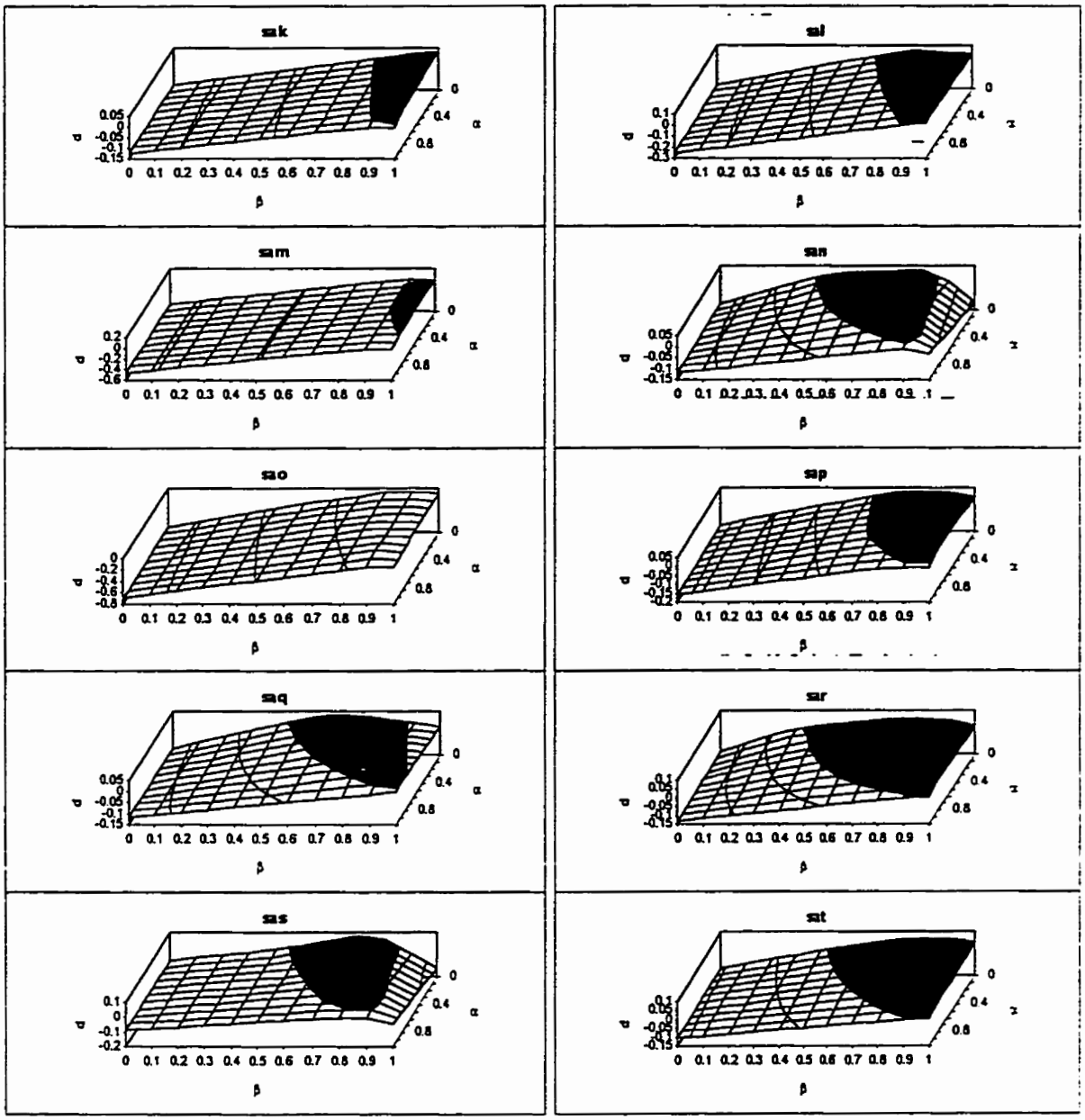


Figure 6.3 Suite.

Les résultats démontrent une certaine tendance dans le positionnement des valeurs optimales: le d maximum est obtenu pour des valeurs de $\beta_{opti} > 0.5$ et $\alpha_{opti} < 0.5$. Le positionnement de β_{opti} au-dessus de 0.5 signifie que les mesures d'origine locale Δl_{tot} et Δp_{tot} ont un pouvoir discriminant plus grand que la mesure globale ΔL. Et, puisque α_{opti} est plus près de zéro que de l'unité, c'est la mesure de *différence de positions moyenne* Δp_{tot} qui est la plus discriminante.

Ainsi optimisée, la mesure de distance ζ_{MEC} permet une séparation complète des classes ω_1 et ω_2 et ce, dans 75% des cas. Ce résultat tend à confirmer la capacité discriminante de la mesure proposée. Puisque tel est le cas, il est possible de procéder à la phase de *généralisation*. Mais, avant tout, des seuils de classification doivent être déterminés, en se basant sur les résultats de la phase d'*apprentissage* et plus particulièrement sur l'erreurs en fonction du seuil. Ces résultats, pour chacun des scripteurs, sont présentés dans la prochaine figure.

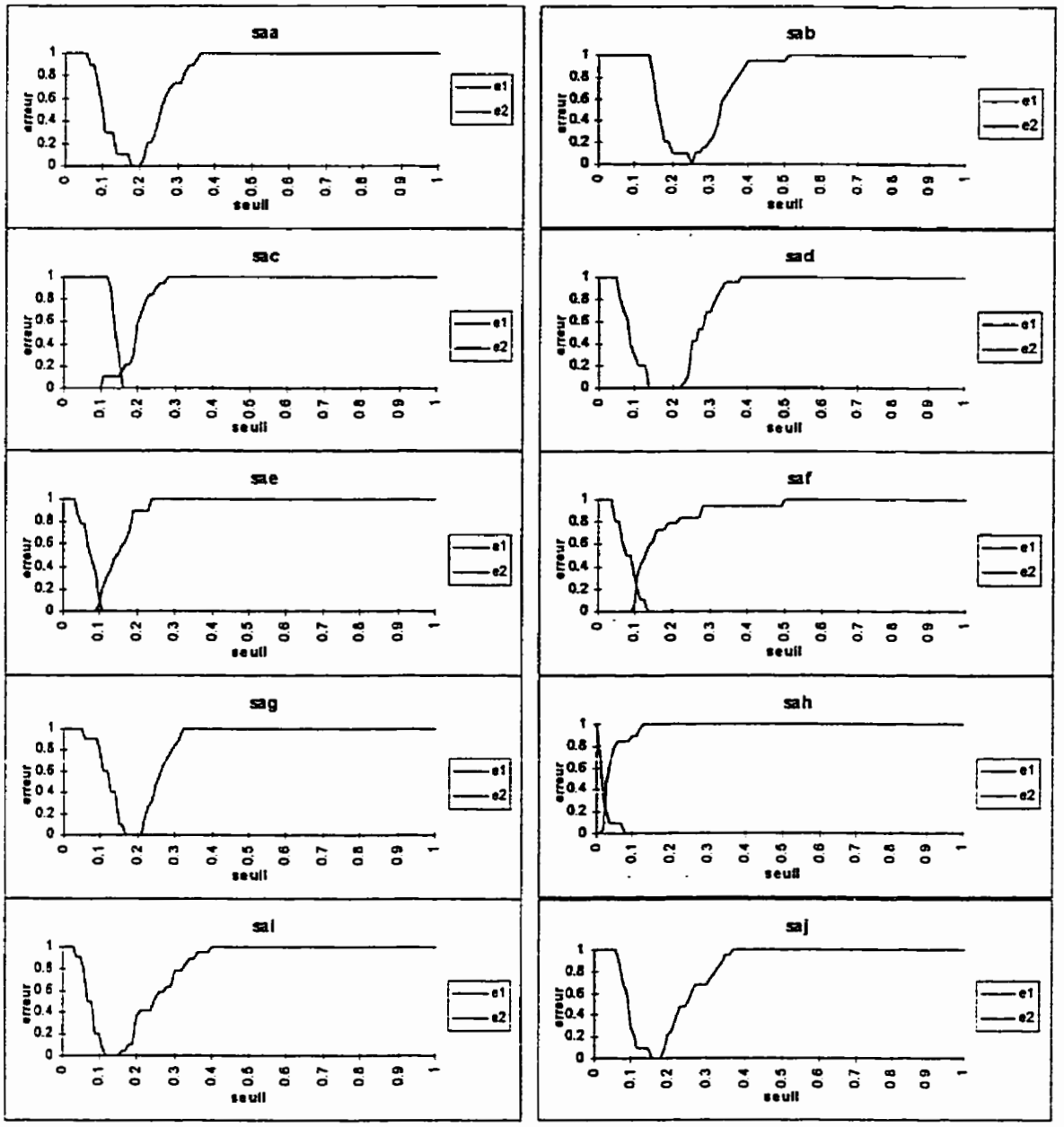


Figure 6.4 Graphiques illustrant, pour chacun des scripteurs, les taux d'erreur ε_1 et ε_2 en fonction de la position du seuil. Ces résultats de la phase d'apprentissage permettent d'identifier des seuils de classification personnalisés pour chaque scripteur.

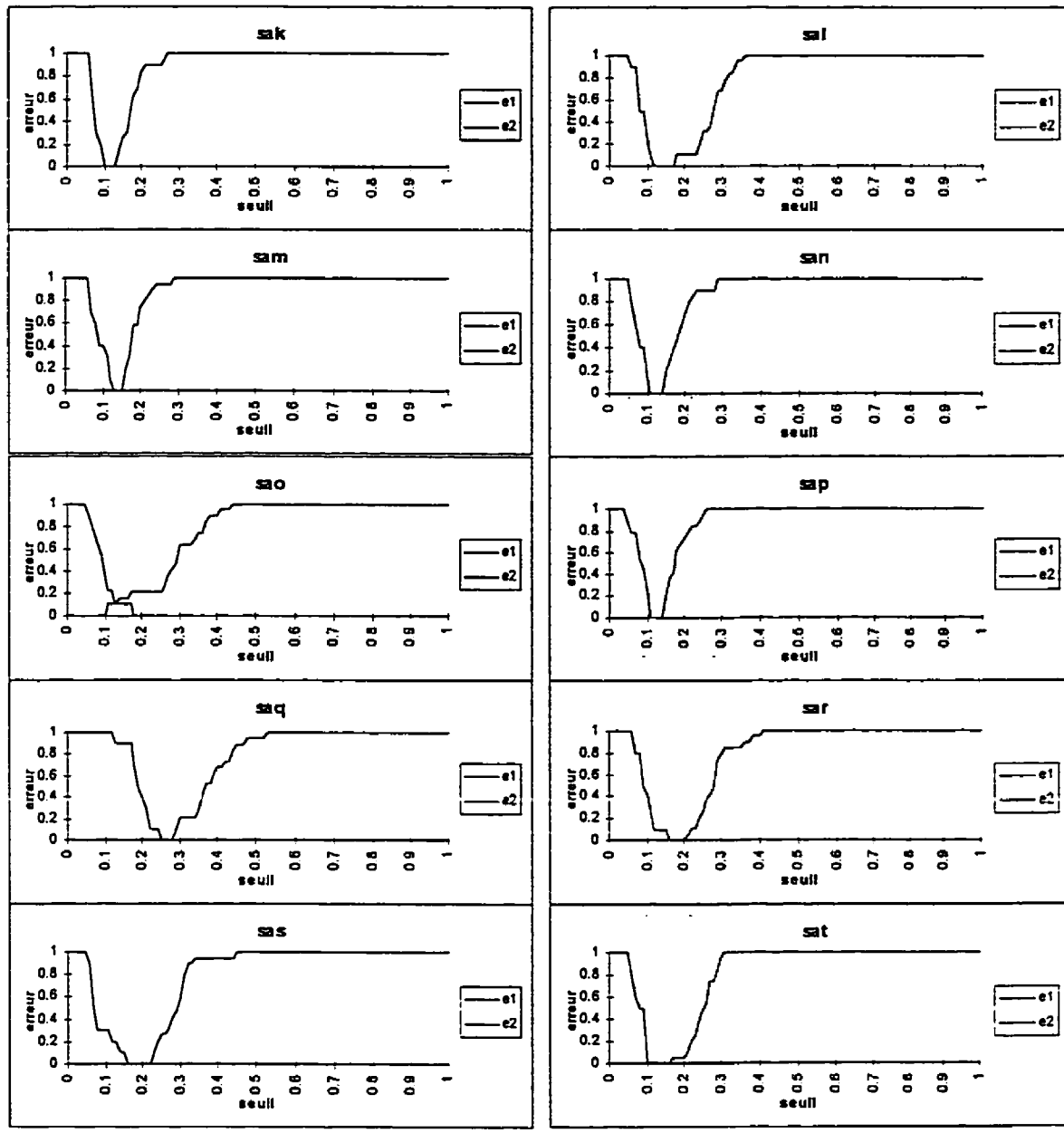


Figure 6.4 Suite.

Le positionnement des seuils se fait de manière à minimiser une des trois erreurs de classification (ε_1, ε_2 ou ε_{moy}) selon les objectifs visés. La Figure 6.5 illustre la façon de déterminer les seuils en se basant sur le graphe des erreurs en fonction du seuil de classification. Le seuil τ_1 minimisant ε_1 est positionné à la frontière supérieure de ω_1, représentée sur le graphique par la plus petite valeur pour laquelle ε_1 est nulle. De façon similaire, le seuil τ_2 minimisant ε_2 est positionné à la frontière inférieure cette fois de ω_2, représentée sur le graphique par la plus grande valeur pour laquelle ε_2 est nulle. Pour ce qui est du seuil τ_{moy} minimisant ε_{moy}, son mode d'attribution dépend du positionnement des classes. Si les classes se chevauchent [Figure 6.5a], τ_{moy} est positionné là où ε_{moy} est minimum. Par contre, si les classes sont complètement séparées [Figure 6.5b], créant ainsi un intervalle pour lequel ε_{moy} est nul, alors τ_{moy} est positionné au milieu de cet intervalle.

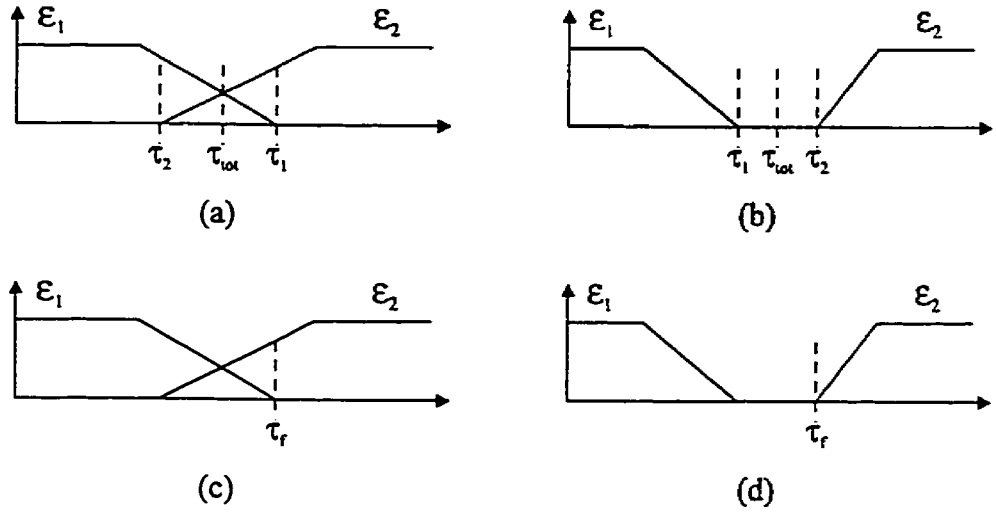


Figure 6.5 Positionnement des différents seuils de classification lorsqu'il y a chevauchement des classes (a) et lorsqu'elles sont séparées (b). Positionnement proposé pour le seuil de classification dans les deux même situations (c et d).

Tel que mentionné, le choix du seuil doit se faire en considérant les objectifs visés. Ici, l'objectif est d'évaluer la possibilité d'utiliser le système proposé, originalement dédié à l'inférence de la séquence, comme premier étage d'un filtre permettant de rejeter les faux aléatoires. Comme ce système ne constitue qu'un module de prétraitement, donc non dédié à la prise de décisions finales, le rejet d'un *authentique* à ce stade constitue une erreur irrécupérable. Tandis que l'erreur contraire, l'acceptation d'un *faux aléatoire*, pourra éventuellement être corrigée plus après dans la chaîne de traitement. Une relecture de l'affirmation précédente impose le choix d'un seuil τ_f minimisant l'erreur de type 1, c'est-à-dire un seuil qui évite le rejet d'*authentiques*, même si cela impose d'accepter des *faux aléatoires*. C'est notamment le cas lorsque les classes se chevauchent [Figure 6.5c]. Mais lorsque les classes sont clairement séparées, est-il nécessaire de positionner le seuil

tel que montré à la Figure 6.5b? Une approche plus sûre consisterait à positionner le seuil à la limite supérieure de l'intervalle entre les deux classes [Figure 6.5d]. De cette façon, la « zone neutre » que représente cet intervalle est associée à la classe des *authentiques* plutôt que la classe des *faux aléatoires*.

6.6 Analyse des résultats

L'évaluation des performances du système proposé se fait à partir des résultats des différentes expériences décrites précédemment. L'analyse de ces résultats consiste à comparer les squelettes résultant du traitement avec les signaux dynamiques utilisés comme référence pendant la procédure. À titre de référence, les différentes expériences menées sont résumées dans le Tableau 6.6.

Tableau 6.6 Synthèse des différentes expériences menées pour l'évaluation des performances du système proposé.

expérience	description	classe de signaux utilisés
1	comparaison de signaux originaux	ω_1
2	comparaison de signaux authentiques	ω_1
3a	prototypage, entraînement (incluant résultats de l'expérience 2)	ω_2
3b	généralisation	ω_1, ω_2

Dans un premier temps, il convient d'analyser les résultats sous l'angle de l'efficacité de la méthode proposée. Les seules mesures quantitatives disponibles pour porter un jugement sur l'efficacité sont les métriques préalablement définies [section 6.2].

La notion d'efficacité renvoie ici à la capacité qu'offre le système à bien inférer la séquence lorsque la comparaison est faite entre des signaux authentiques. Cela concerne donc directements les expériences 1 et 2. Pour bien mettre en relief ces résultats, il importe cependant de les comparer avec ceux de l'expérience 3a.

La première métrique à considérer est la *proportion de séquence non considérée* ΔL. Le graphique de la Figure 6.6 illustre, pour les 20 scripteurs identifiés de *saa* à *sat*, les valeurs moyennes obtenues pour chacune des trois premières expériences (1, 2 et 3a).

Ces données montrent une progression entre les résultats des trois expériences. Les meilleurs résultats sont obtenus à l'expérience 1 et les moins bons à l'expérience 3a, les résultats de l'expérience 2 se situant entre les deux. Cette première observation tend à confirmer la capacité discriminante de la mesure ΔL.

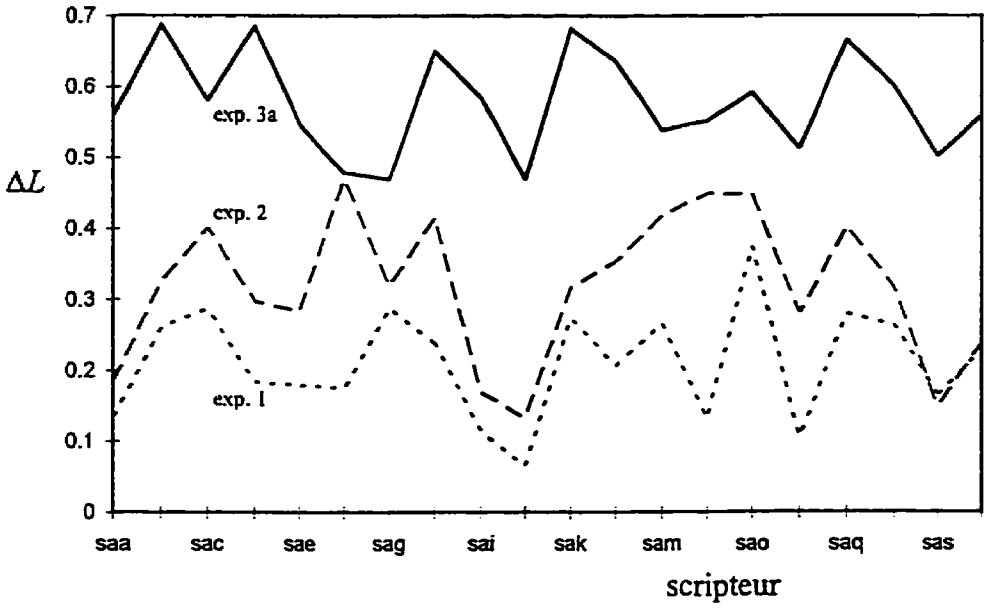


Figure 6.6 Graphe montrant la *proportion de séquence non considérée* ΔL pour les expériences 1, 2 et 3a.

Les deux autres métriques développées, Δl_i et Δp_i, doivent permettre de qualifier localement la séquence inférée en évaluant une distance entre les portions de séquences mises en correspondance. La valeur de ces métriques devraient donc traduire la qualité de la mise en correspondance et, par le fait même, la fiabilité de la procédure d'appariement.

Le prochain graphique [Figure 6.7] montre les valeurs de Δl_i et Δp_i calculées pour chacun des appariements faits lors d'une comparaison. L'intérêt de ce graphe se situe au niveau de la vérification de l'existence d'une relation entre la distance par rapport à l'origine et la qualité de la mise en correspondance.

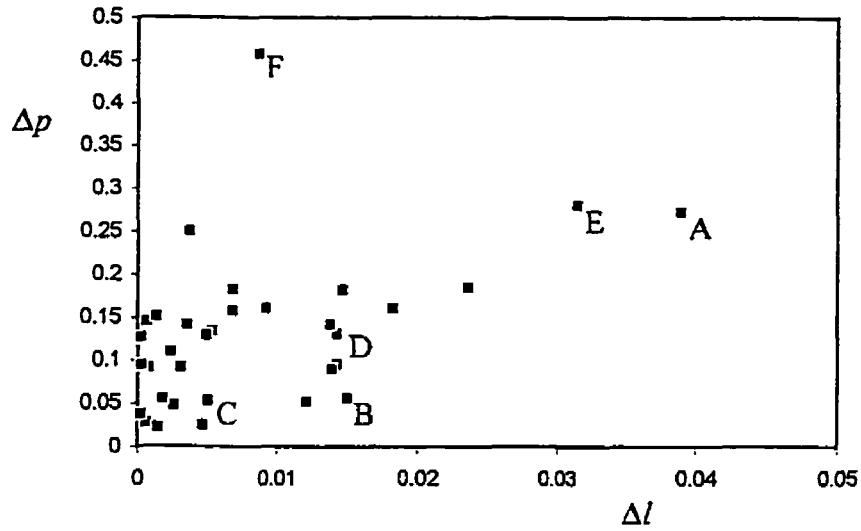


Figure 6.7 Graphique des valeurs de Δl_i et Δp_i pour une comparaison donnée. Chaque point de coordonnée $(\Delta l_i, \Delta p_i)$ corrrespond à un appariement i entre deux segments.

En regardant les couples de segments associés à certains de ces points, il est possible de constater l'existence d'un lien entre la distance et qualité de la mise en correspondance: plus le point est près de l'origine, plus la mise en correspondance semble adéquate. Pour illustrer cette observation, la Figure 6.8 montre les couples de segments correspondant aux points identifiés dans la Figure 6.7. Encore une fois, cela tend à confirmer la capacité discriminante des métriques proposées.

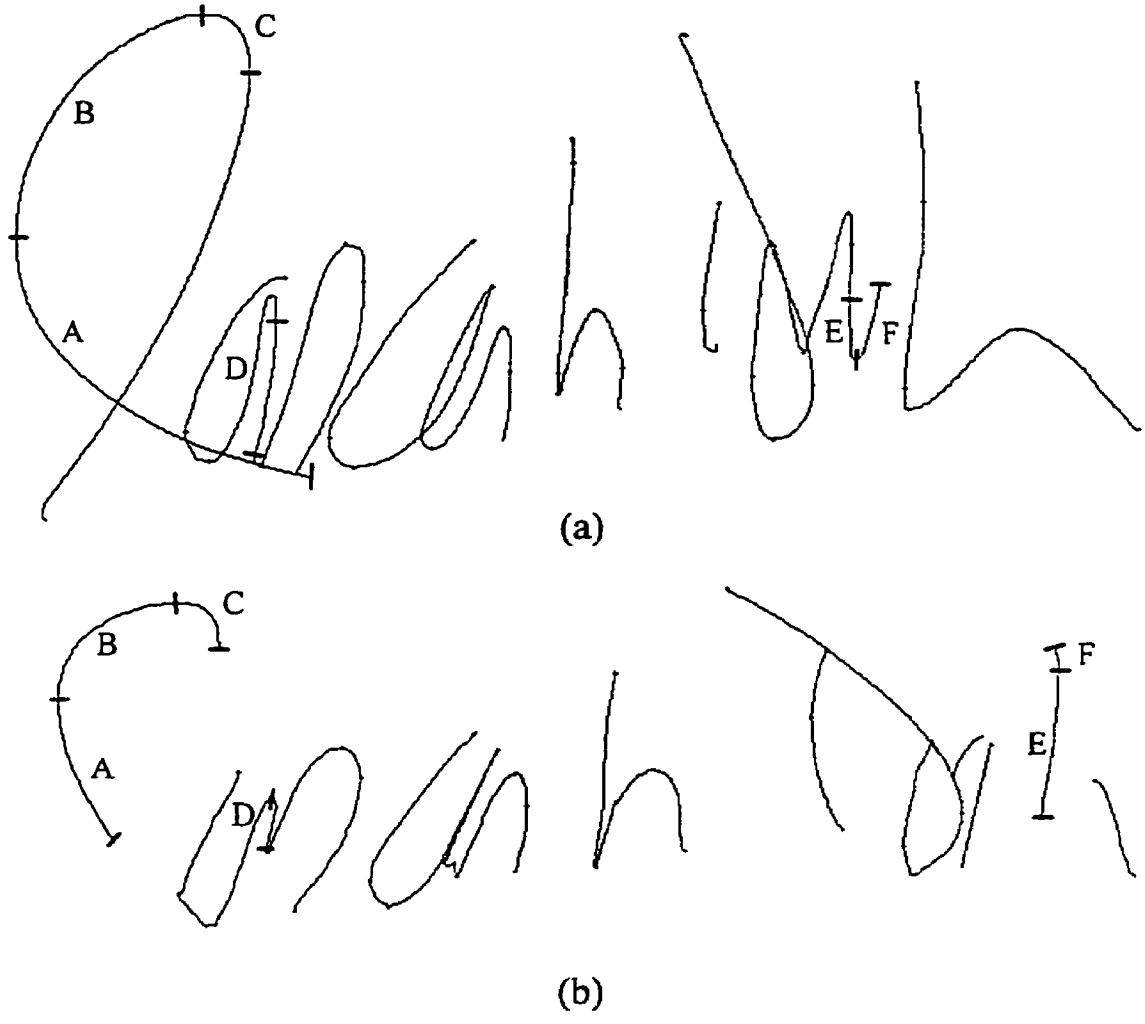


Figure 6.8 (a) Séquence de référence; (b) séquence inférée. Les lettres identifiant les couples de segments réfèrent aux points correspondants dans le graphe de la Figure 6.7.

Même si les métriques proposées semblent discriminantes, cela ne permet pas pour autant de statuer sur la capacité d'inférence du système. Cela permet tout au plus d'en relativiser les résultats. La figure qui suit a justement pour but d'illustrer les capacités du système. Il

s'agit de résultats tirés de l'expérience 2. Pour chacune des comparaisons retenues, la figure montre les résultats de différentes étapes du traitement qui sont, dans l'ordre:

1) l'image de luminance;
2) l'image binaire et les zones caractéristiques;
3) le graphe de relations;
4) la séquence de référence;
5) le graphe de représentation;
6) la séquence inférée (superposée à l'image binaire).

Certaines portions de ces images sont identifiées par des lettres. Ces lettres correspondent à des sources d'erreur pouvant causer des problèmes de traitement. Ces sources d'erreur sont les suivantes:

A: Binarisation.
B: Interprétation des contours / position des points caractéristiques.
C: Édition du graphe de relations.
D: Mise en correspondance / Cheminement dans le graphe de relations.
E: Positionnement du squelette.

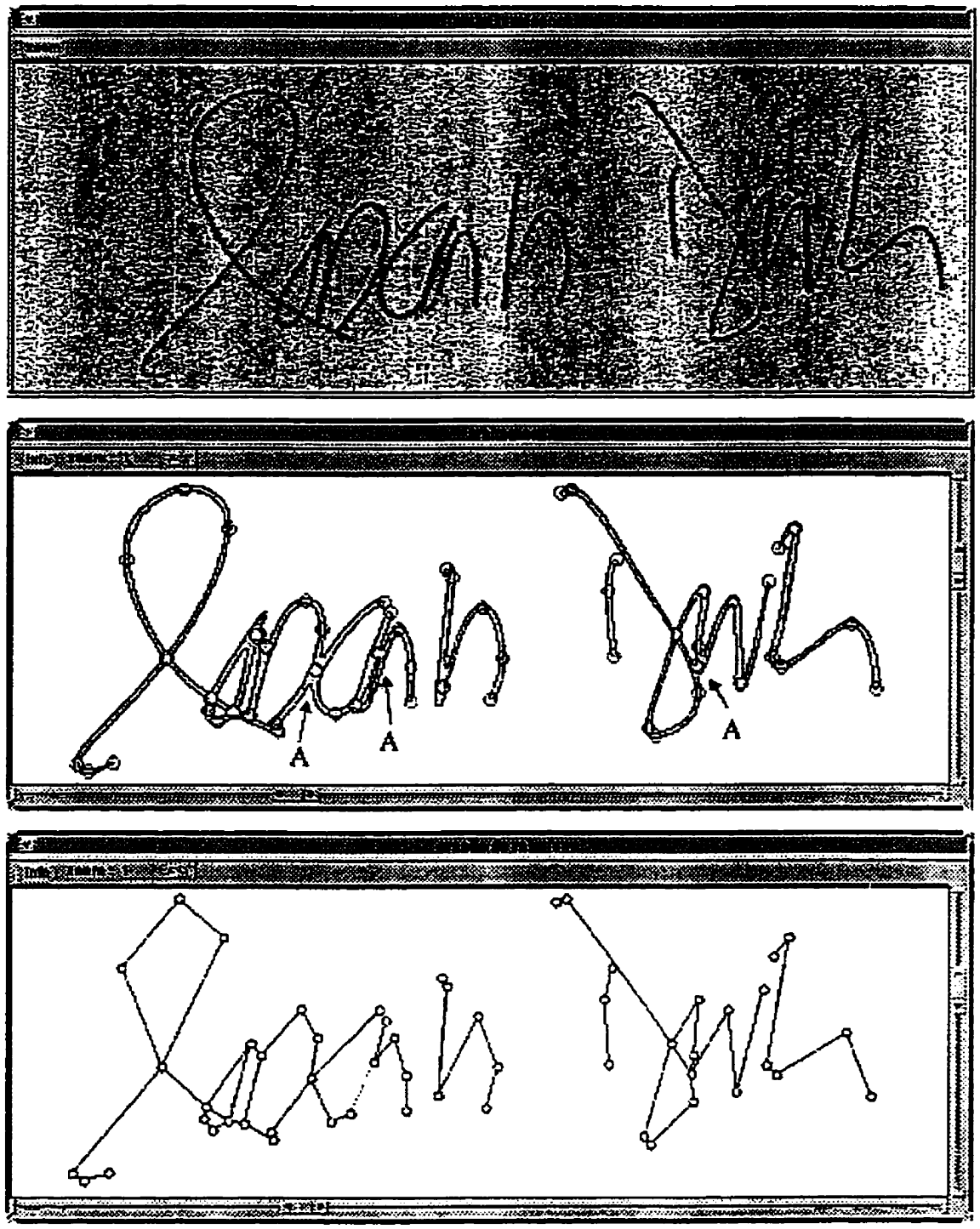


Figure 6.9 Exemples de résultats de traitement tirés de l'expérience 2.

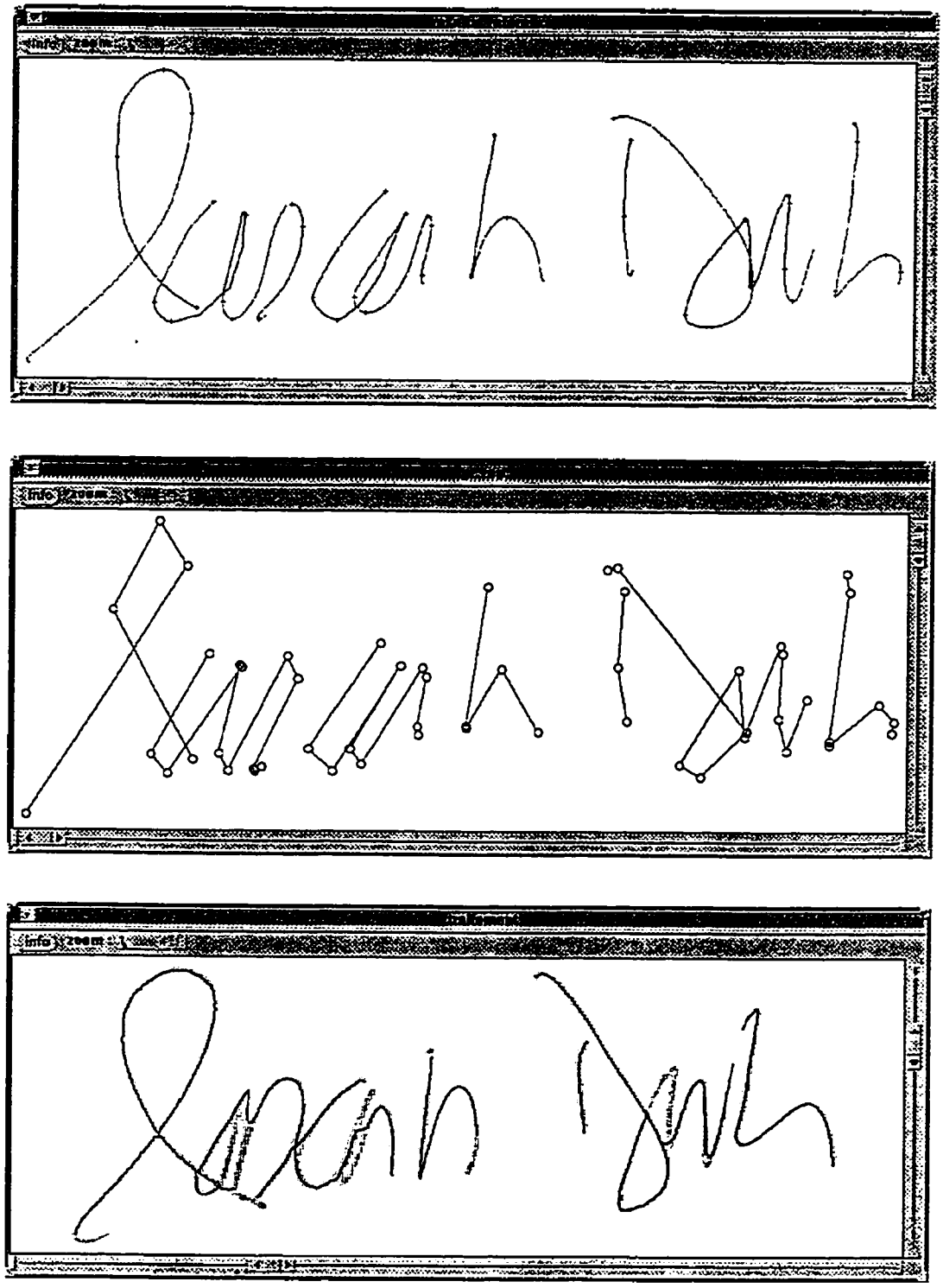


Figure 6.9 Suite.

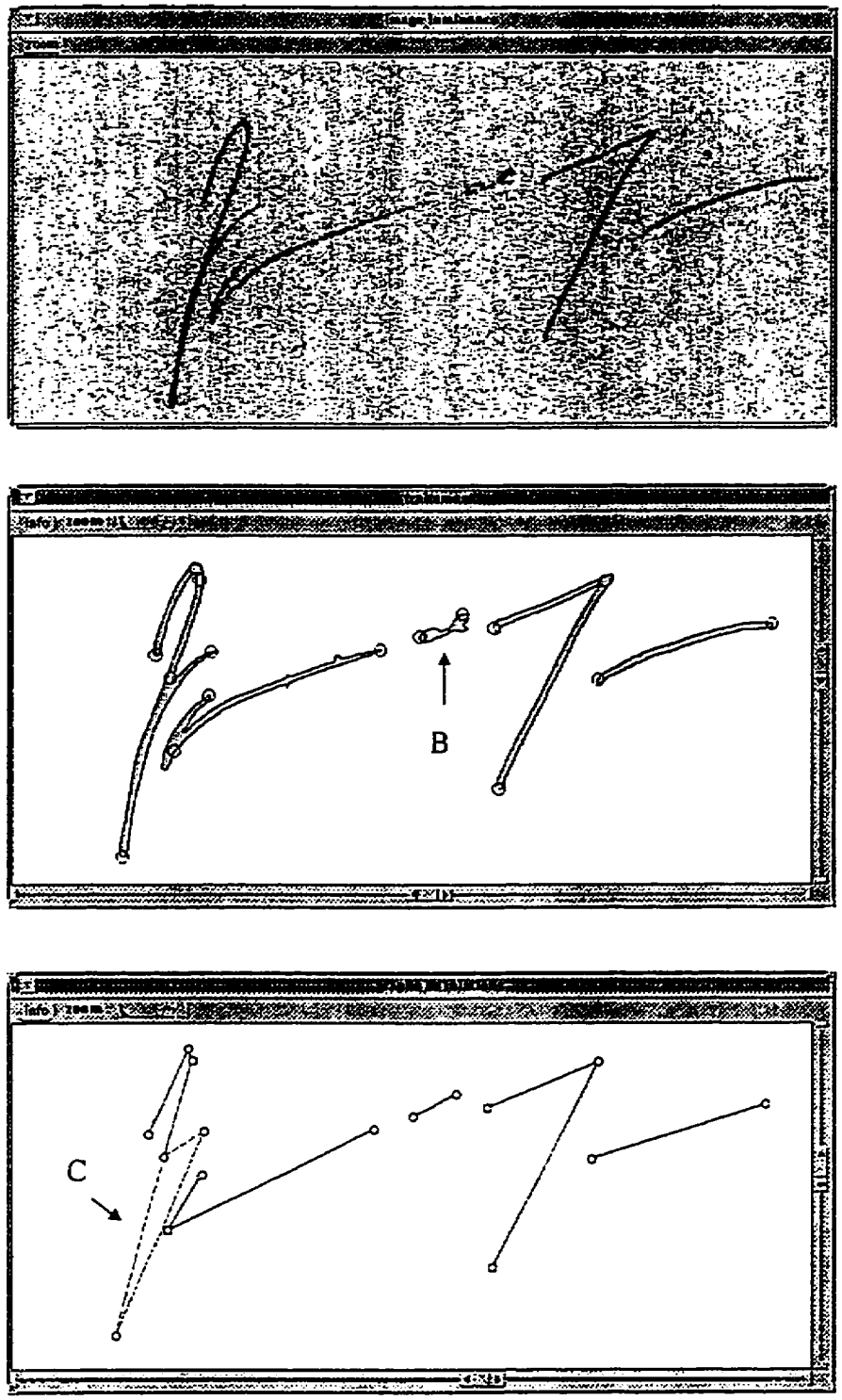


Figure 6.9 Suite.

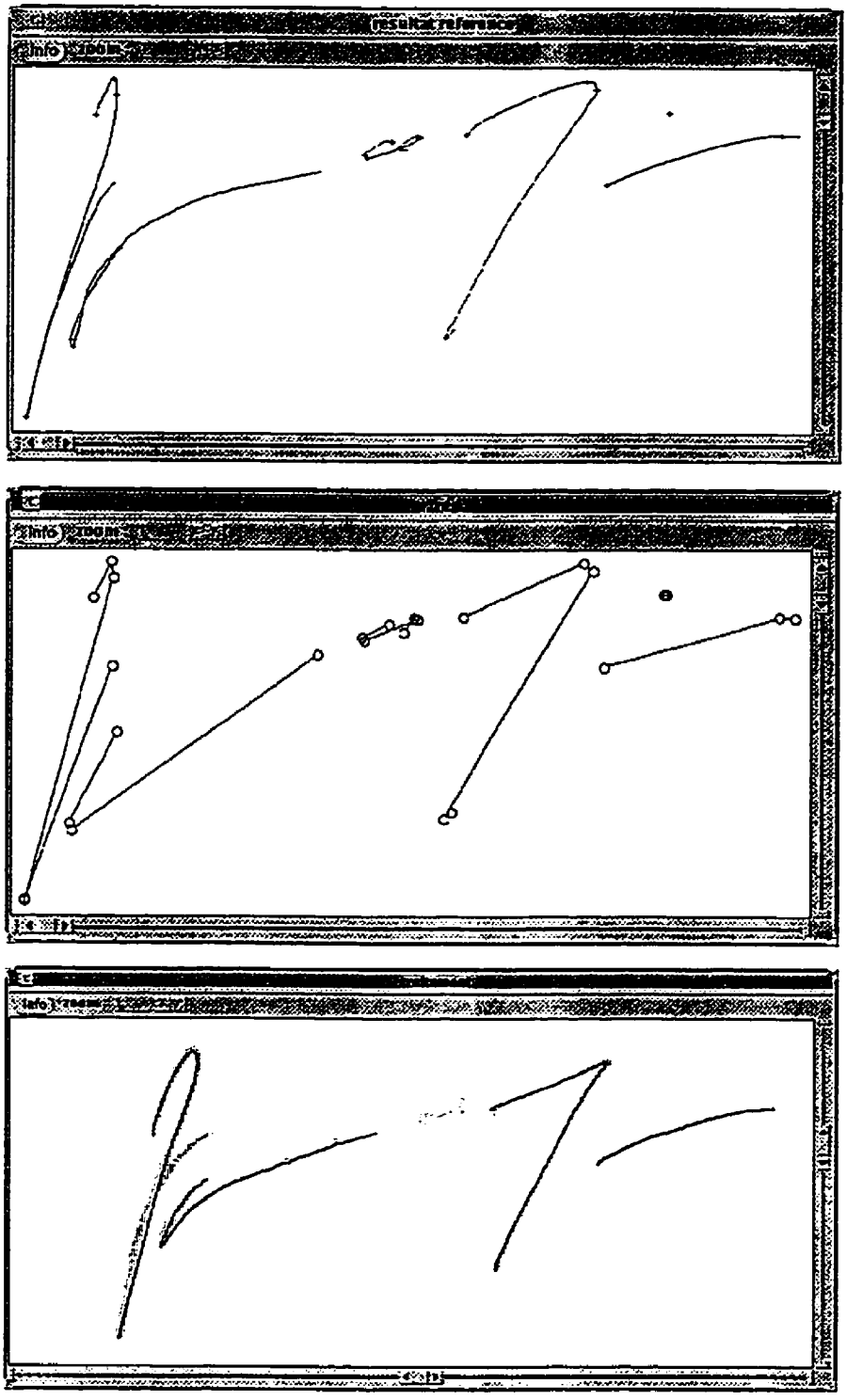


Figure 6.9 Suite.

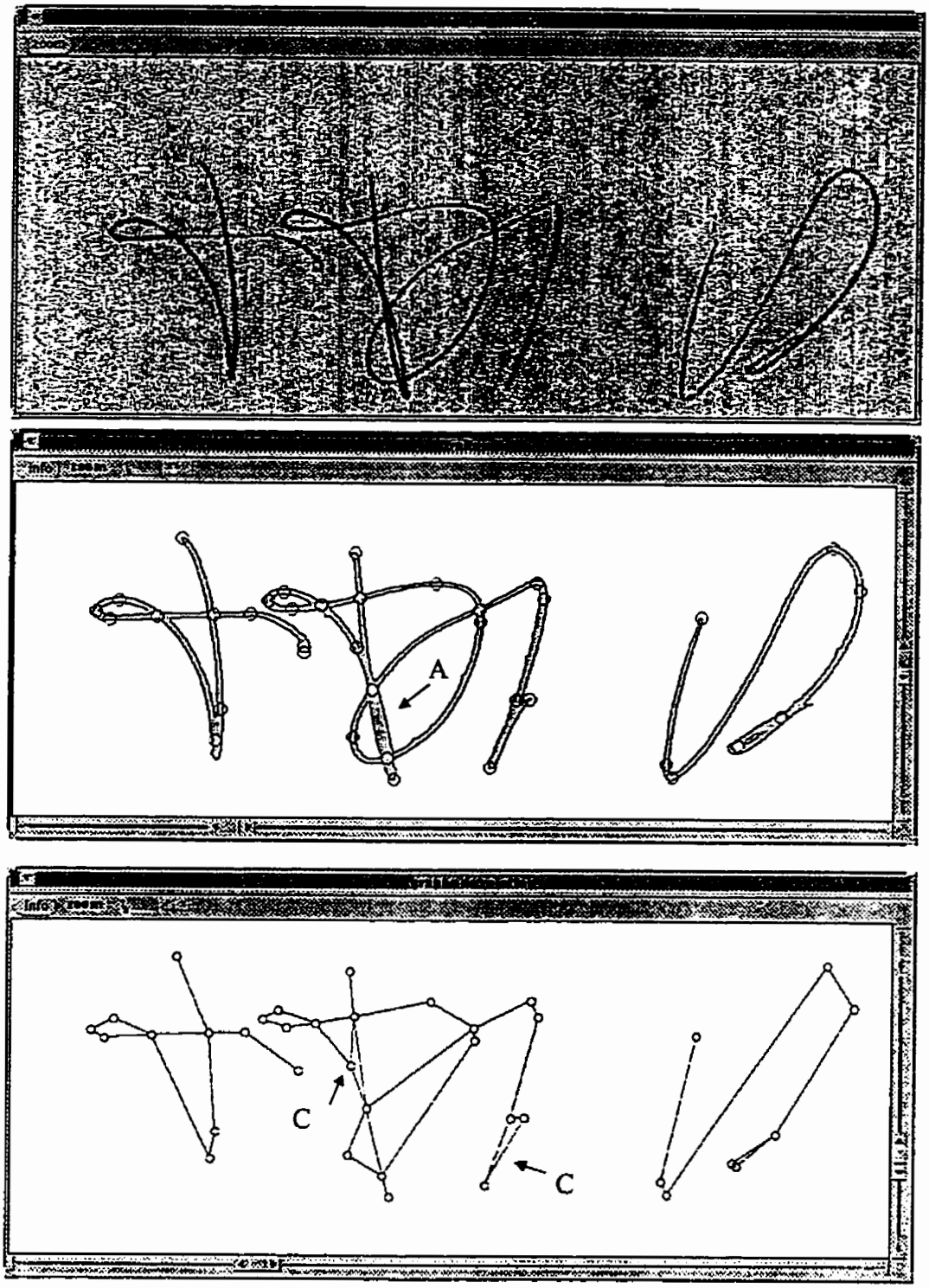


Figure 6.9 Suite.

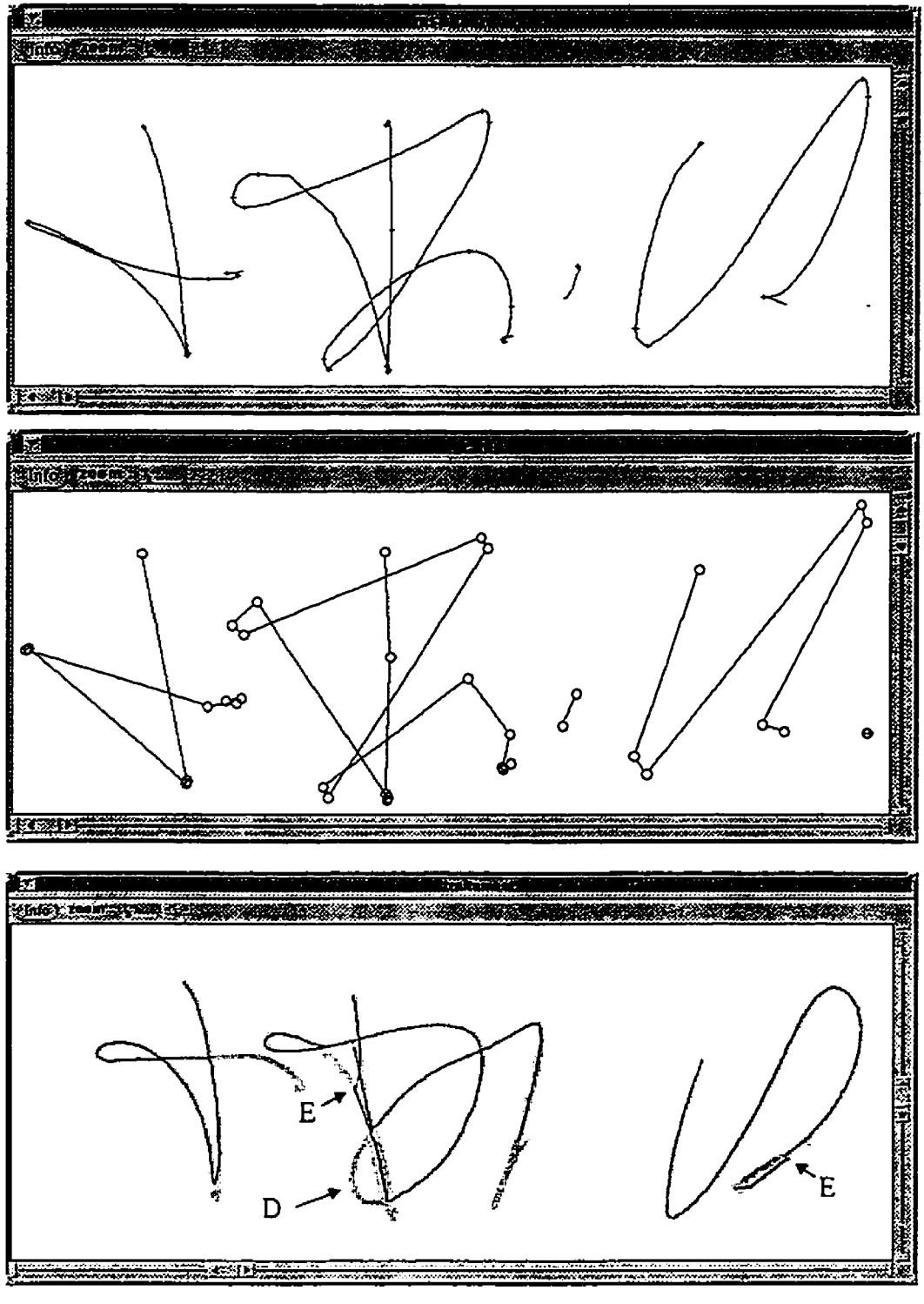


Figure 6.9 Suite.

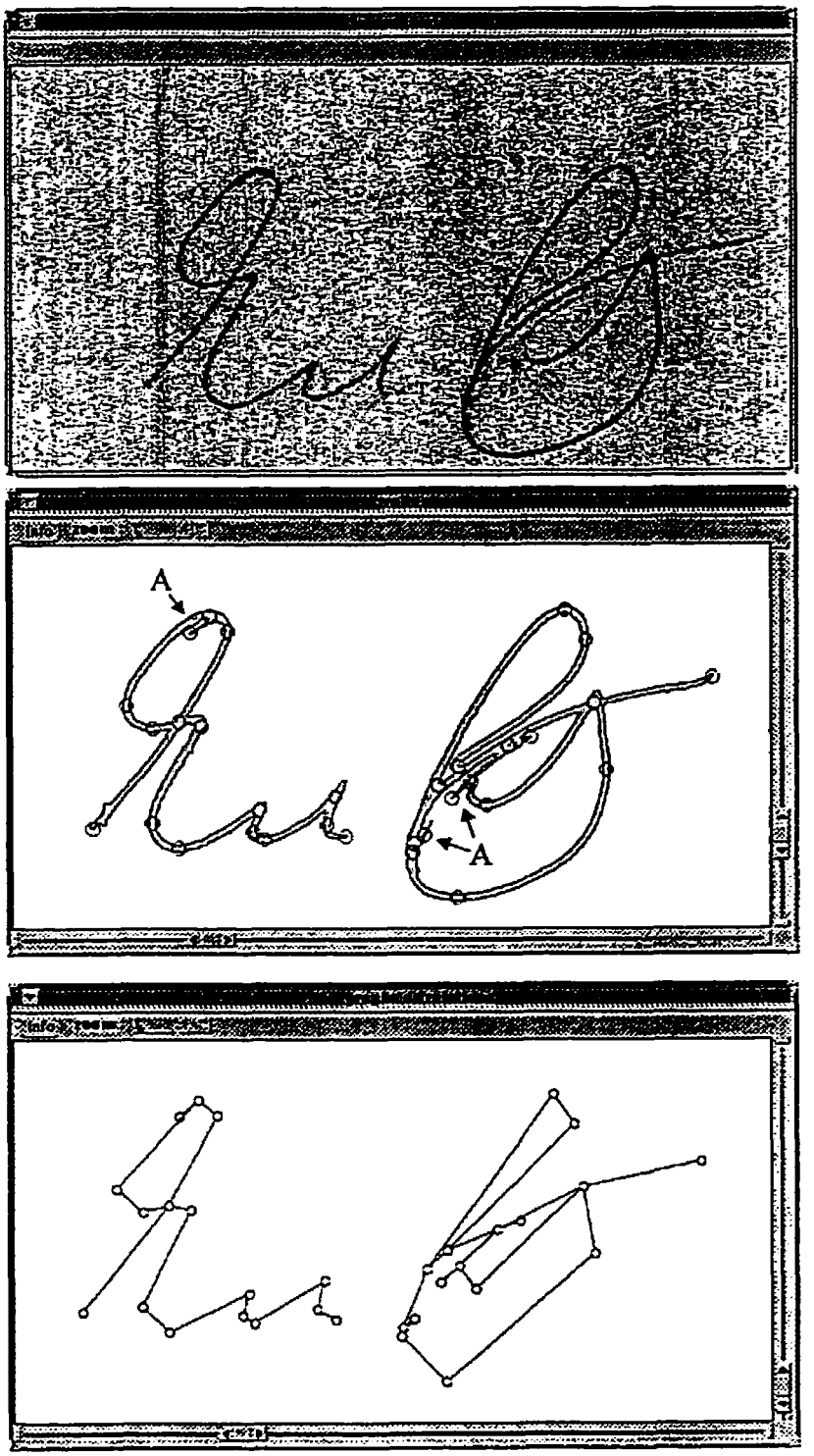


Figure 6.9 Suite.

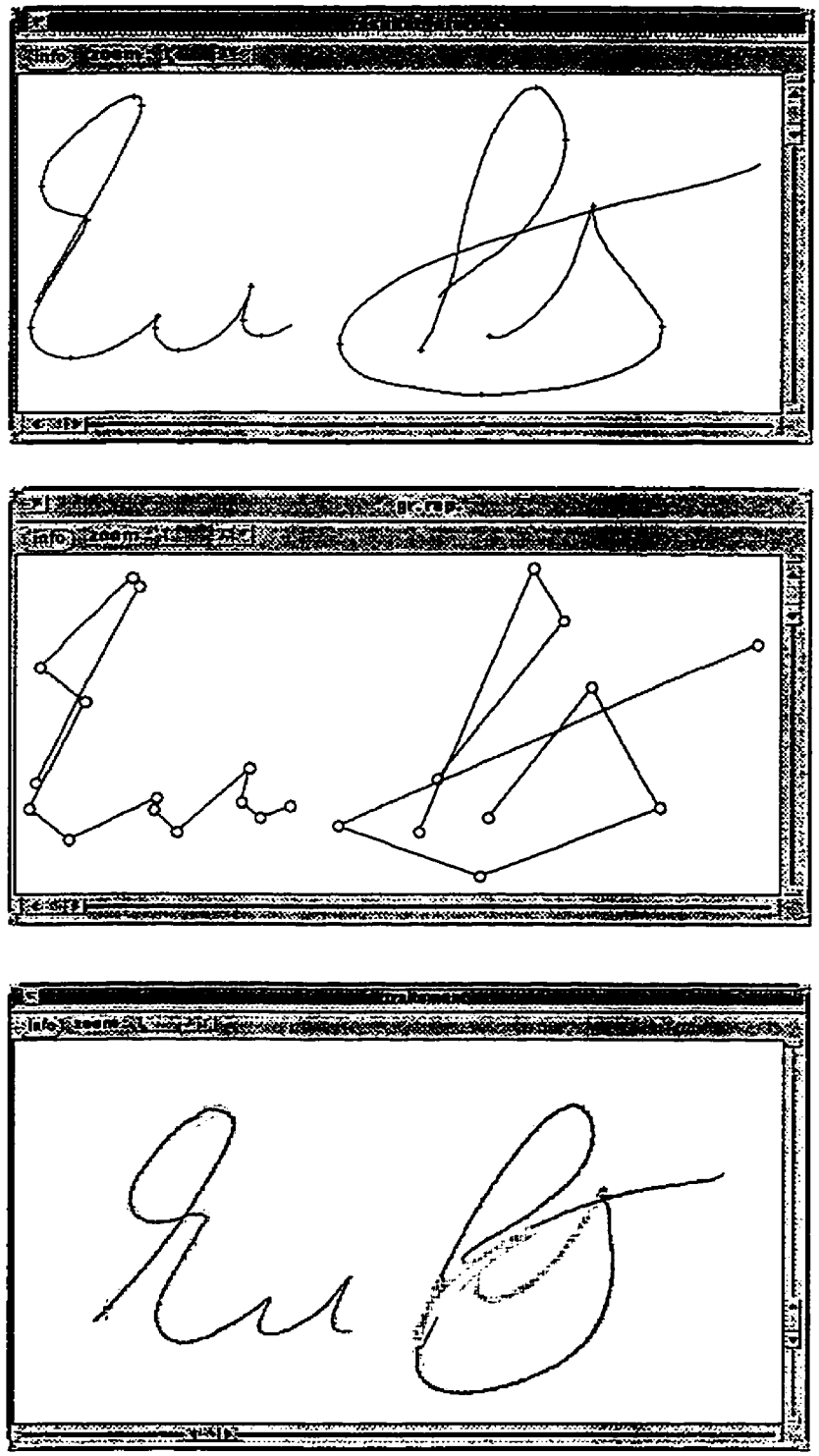


Figure 6.9 Suite.

Les exemples précédents sont suffisamment intéressants pour justifier la conduite de l'expérience 3b qui doit évaluer la capacité de détection des *faux aléatoires*. Les résultats de cette expérience sont regroupés dans le Tableau 6.7. Les taux d'erreur sont donnés pour différentes positions des seuils de classification. Ces seuils ont été déterminés sur la base des résultats des expériences 2 et 3a.

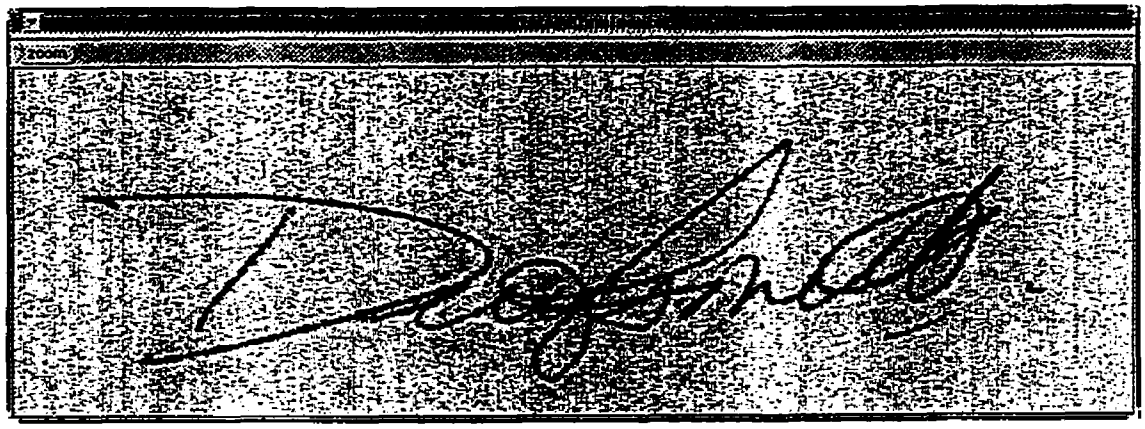
Parmi les différents seuils utilisés, ceux qui présentent un intérêt particulier sont ceux qui minimisent ε_1, l'erreur de type 1. Deux interprétations de cet objectif ont mené à la définition de deux seuils différents: τ_1 et τ_f. Il est intéressant de remarquer que le positionnement de τ_f permet de diminuer d'un facteur 4 l'erreur ε_1 sans même doubler l'erreur ε_2. En fait, ε_1 est nulle pour tous les scripteurs sauf un.

Il semble donc que le positionnement du seuil τ_f constitue un choix intéressant [section 6.5.2]. Même si le taux d'erreur de type 2 semble élevé (18.74%), il faut considérer que cela représente tout de même l'élimination de plus de 80% des *faux aléatoires*. Puisque le but premier du système proposé n'était pas la classification mais bien l'inférence de la séquence, cela peut être considéré comme un atout important. En éliminant ainsi la majorité des *faux aléatoires*, sans pour autant éliminer des *authentiques*, les éventuels traitements subséquents pourront être axés sur des caractéristiques plus particulières du signal.

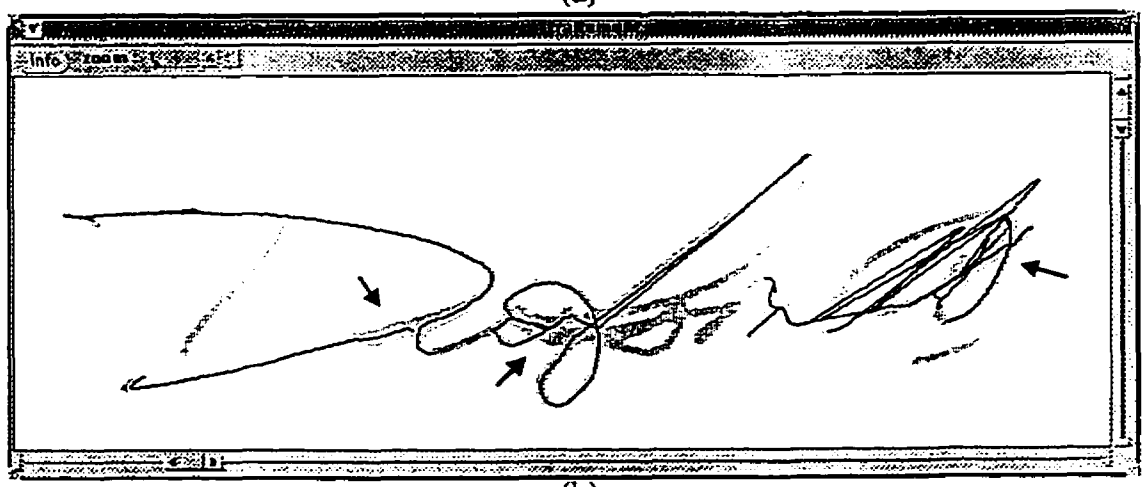
Tableau 6.7 Résultats de l'expérience en classification. Les taux d'erreur sont exprimés en pourcentage.

	τ_{moy}			τ_1			τ_2			τ_f		
script.	ε_1	ε_2	ε_{moy}	ε_1	ε_2	ε_{moy}	ε_1	ε_2	ε_{moy}	ε_1	ε_2	ε_{moy}
saa	0.00	7.14	3.57	0.00	0.00	0.00	0.00	7.14	3.57	0.00	7.14	3.57
sab	0.00	12.50	6.25	0.00	12.50	6.25	0.00	12.50	6.25	0.00	12.50	6.25
sac	0.00	11.11	5.56	0.00	11.11	5.56	22.22	0.00	11.11	0.00	11.11	5.56
sad	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.11	5.56	0.00	11.11	5.56
sae	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
saf	16.67	52.63	34.65	0.00	73.68	36.84	50.00	5.26	27.63	0.00	73.68	36.84
sag	0.00	5.26	2.63	0.00	0.00	0.00	0.00	21.05	10.53	0.00	21.05	10.53
sah	10.00	6.25	8.13	0.00	93.75	46.88	50.00	0.00	25.00	0.00	93.75	46.88
sai	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
saj	0.00	6.25	3.13	0.00	6.25	3.13	0.00	6.25	3.13	0.00	6.25	3.13
sak	0.00	14.29	7.14	11.11	0.00	5.56	0.00	14.29	7.14	0.00	14.29	7.14
sal	20.00	0.00	10.00	50.00	0.00	25.00	0.00	0.00	0.00	0.00	0.00	0.00
sam	0.00	15.79	7.89	0.00	5.26	2.63	0.00	31.58	15.79	0.00	31.58	15.79
san	0.00	10.53	5.26	11.11	0.00	5.56	0.00	21.05	10.53	0.00	21.05	10.53
sao	0.00	22.22	11.11	0.00	22.22	11.11	30.00	11.11	20.56	0.00	22.22	11.11
sap	0.00	7.69	3.85	0.00	7.69	3.85	0.00	15.38	7.69	0.00	15.38	7.69
saq	25.00	0.00	12.50	37.50	0.00	18.75	25.00	5.26	15.13	25.00	5.26	15.13
sar	0.00	5.56	2.78	0.00	0.00	0.00	0.00	11.11	5.56	0.00	11.11	5.56
sas	0.00	6.25	3.13	0.00	0.00	0.00	0.00	6.25	3.13	0.00	6.25	3.13
sat	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.11	5.56	0.00	11.11	5.56
moyenne	3.58	9.17	6.38	5.49	11.62	8.55	8.86	9.52	9.19	1.25	18.74	10.00

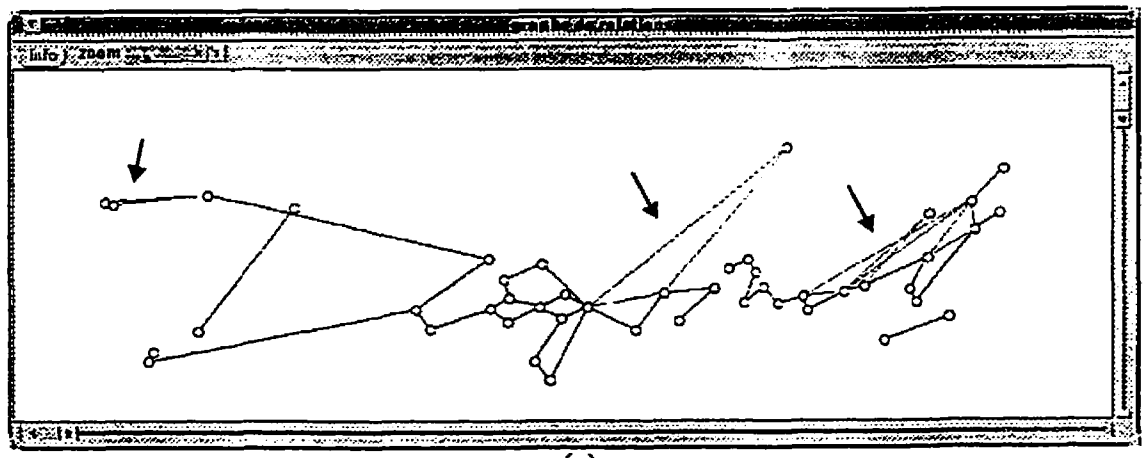
Pour l'expérience 3b, l'erreur de type 1 est non nulle pour un seul scripteur. L'observation d'une des signatures *authentiques* rejetées pour ce scripteur [Figure 6.10] montre les limites de la technique de binarisation utilisée. Les portions d'images pointées par des flèches dans l'*image binarisée* [Figure 6.10b] sont autant d'exemples de mauvaises segmentations visuelles. Bien évidemment ces erreurs se répercutent dans les étapes successives du traitement et ont comme effet ultime de fausser l'information contenue dans le *graphe de relations* [Figure 6.10c], causant ultimement des erreurs de squelettisation [Figure 6.10b].



(a)



(b)



(c)

Figure 6.10 Trois représentations d'une signature *authentique* rejetée lors de l'expérience en classification: (a) Image de luminance I_e , (b) séquence inférée S_{inf} superposée à l'image binarisée I_{bin} , (c) graphe de relations G_{rel} .

Il peut également y avoir non pas des erreurs d'interprétation, mais des différences d'interprétation: ce que l'image montre ne correspond pas à la dynamique originale. La Figure 6.11 illustre ce propos. Cette figure montre un détail d'une image binarisée [Figure 6.11a] et la séquence originale correspondante [Figure 6.11c]. Les lettres A, B et C identifient des portions de tracé qui, même si elles correspondent à la même information, sont interprétées différemment. La portion A par exemple présente un extremum en Y, tandis que l'équivalent dans la dynamique n'en a pas. C'est exactement le contraire pour les portions B et C.

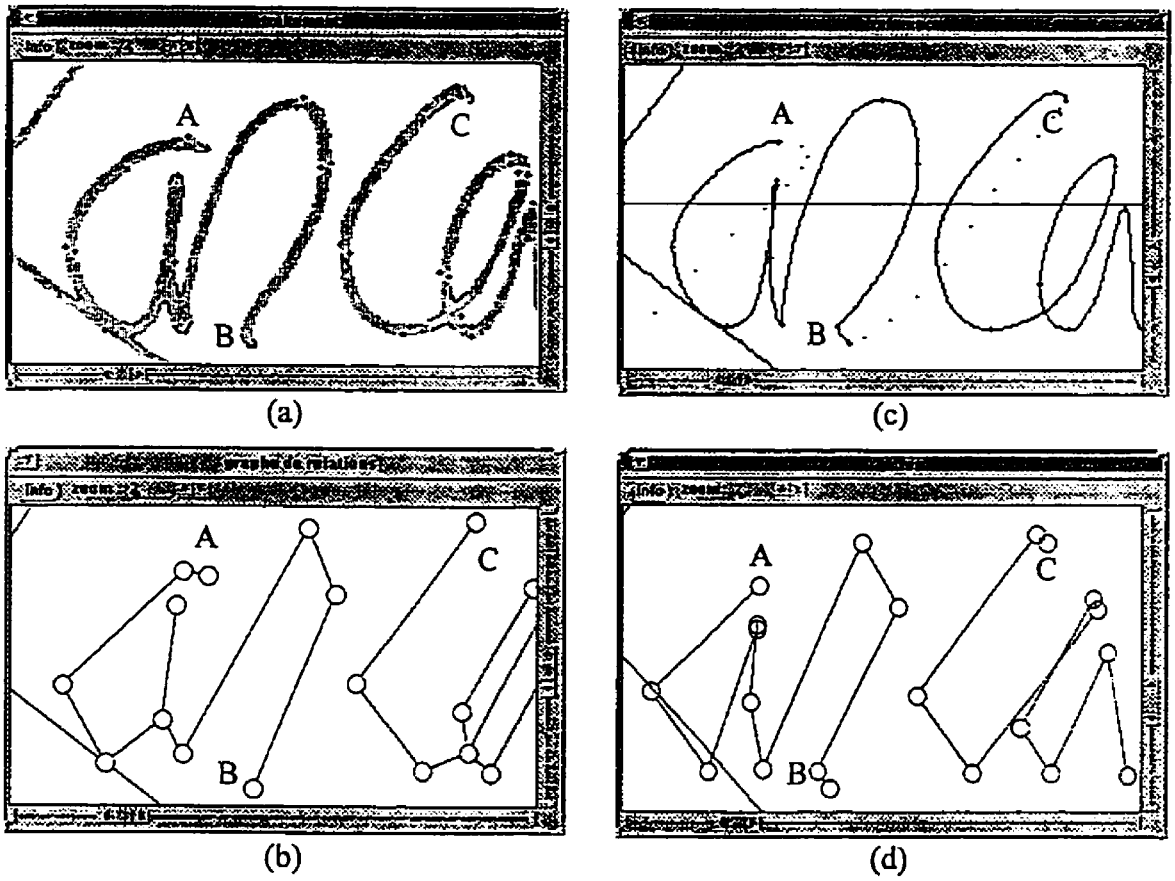


Figure 6.11 Quatre représentations tirées de la même information originale illustrant le problème de la différence d'interprétation: (a) image binarisé, (b) graphe de relations, (c) séquence de référence et (d) graphe de représentation.

Une première explication pour ces différences dépend de la nature même de l'outil d'écriture. Il est possible que la sensibilité du stylo instrumenté soit telle que les levées ou posées de crayon ne correspondent pas exactement à ce qui apparaît dans le tracé. Un décalage de quelques échantillons en début ou fin de composante peut faire la différence. Il peut s'agir aussi d'un mauvais fonctionnement de la pointe à bille qui déposerait une trace d'encre dont la forme pourrait laisser croire à la présence de points caractéristiques

supplémentaires n'apparaissant pas dans la dynamique. Une troisième explication impliquerait les méthodes d'identification des points caractéristiques. Il est possible que le choix des seuils ait comme effet d'occulter certains de ces points ou d'en identifier certains inutilement.

L'identification de sources d'erreurs potentielles dans les exemples précédents n'a pas comme but de discréditer la méthode proposée, mais doit plutôt servir à guider les développements futurs. Les résultats des différentes expériences ayant démontré la pertinence de l'approche, les efforts devraient être concentrés sur les étapes de traitements les plus critiques.

Il semble que la plupart des problèmes sont relatifs au traitement de l'information statique. Étant donné que la tâche à accomplir est justement de réinterpréter l'information contenue dans l'image, il était prévisible que les problèmes soient concentrés dans les modules y étant associés.

Puisque l'inférence de la séquence est le résultat d'un succession de traitements, lorsqu'une erreur se produit, elle peut être attribuée soit directement à l'étape de traitement concernée, soit à la répercussion d'une erreur survenue antérieurement dans la chaîne de traitement. Ce facteur doit être pris en compte pour juger des performances des différents modules de traitement. Pour les mêmes raisons, des modifications futures

apportées à une des étapes du traitement devront être évaluées en considérant l'effet qu'elles auront sur les performances des étapes ultérieures.

7. CONCLUSION

7.1 Rappel des objectifs et travail accompli

S'inscrivant dans le cadre de l'élaboration d'une nouvelle méthode de vérification de signatures *hors-ligne*, le présent travail avait comme objectif le développement d'une nouvelle méthode d'inférence de la séquence dans l'image d'une signature manuscrite. Tout cela dans le but d'appliquer des techniques de vérification faisant appel à de l'information spatio-temporelle, techniques jusqu'alors réservées aux systèmes *en-ligne*.

Puisqu'un système de vérification doit toujours comparer un signal de *test* avec un signal de *référence* contenu dans une *banque de données*, la décision fut prise d'utiliser l'information de cette banque pour guider la procédure d'inférence de la séquence. La *banque de données* utilisée est constituée de spécimens de signatures représentés sous forme *dynamique*, c'est-à-dire une suite de points échantillonnés à l'aide d'une tablette digitalisante.

L'inférence de la séquence dans l'image se fait donc en comparant l'image d'une signature (le signal de *test*) avec un signal dynamique de la *banque de données* (le signal de *référence*), et en tentant d'identifier dans l'image une séquence correspondant à celle représentée dans le signal dynamique.

Pour que la comparaison soit possible, les deux signaux doivent faire l'objet d'un traitement particulier ayant comme objectif de créer deux *graphes* représentant chacun un des signaux. De par la nature même des signaux, les deux graphes diffèrent. Le graphe de représentation G_{rep}, représentant la dynamique, est un graphe séquencé, tandis que le graphe de relations G_{rel}, représentant l'image, ne l'est pas. En plus, ce dernier comporte des éléments qui lui sont propres comme les noeuds de *croisement* et de *boucle dégénérée*.

Avant de procéder à la mise en correspondance, le *graphe de relations* G_{rel} doit être transformé afin de ne contenir que des types de noeud pouvant exister dans le second graphe. Une fois cette tâche accomplie, la mise en correspondance entre les deux graphes est tentée.

Le séquence inférée dans l'image est identifiée à partir des résultats de cette mise en correspondance. Puisque le *graphe de représentation* G_{rep}, associé au signal dynamique est lui-même séquencé, il suffit de reproduire cette séquence parmi les noeuds du *graphe de relations* G_{rel} mis en correspondance.

Ce cheminement entre les noeuds du graphe G_{rel} représentant l'image permet de guider la procédure de squelettisation qui constitue la dernière étape de traitement du module d'inférence de la séquence.

7.2 Contribution originale

Prise globalement, cette méthode se distingue des autres, entre autres, par le fait qu'elle traite des signaux de nature différente. Les signaux de *référence* constituant la banque de données sont des signaux dits dynamiques, tandis que les signaux de *test* sont des images de luminance. Tous les autres systèmes de vérification de signatures hors-ligne n'utilisent que des images de luminance même pour les signaux de *référence*. Même si cela paraît paradoxal d'utiliser des signaux dynamiques dans une système hors-ligne, rappelons que les signaux de référence doivent être acquis dans un environnement contrôlé afin de s'assurer de leur authenticité. Cette contrainte impose donc les mêmes impératifs quelle que soit la nature des signaux acquis.

Non seulement la nature des signaux diffère-t-elle, mais en plus l'information contenue dans la *banque de données* est utilisée comme information *a priori* pour guider la procédure d'inférence. Cela constitue une autre caractéristique distinctive et innovatrice par rapport aux techniques existantes.

Puisque la nature des signaux diffère, les procédures menant à la constitution des différents graphes diffèrent également. Le traitement de l'information dynamique étant relativement simple, rien de particulièrement innovateur n'y a été fait. Par contre, dans le traitement de l'information statique, quelques solutions originales ont été proposées pour résoudre certains problèmes.

Parmi ces solutions mentionnons d'abord le développement d'un *indice de visibilité*. Cet indice a comme rôle de quantifier la position relative entre deux points de contours en fonction de la distance euclidienne et des distances angulaires (exprimées par rapport aux vecteurs perpendiculaires au contour) séparant les points. Les différentes mesures de cet indice sont impliquées dans la formation des zones caractéristiques.

Une autre solution originale, qui s'applique à plusieures étapes du traitement, est la définition des différents seuils par rapport à un facteur commun. La presque totalité des seuils nécessaires au fonctionnement du système sont en effet déterminés comme des multiples de ***LMT***, la *largeur moyenne du tracé*. Cela a pour effet de rendre le choix de ces seuils indépendant du contexte. Si l'outil d'écriture change ou que la résolution de l'instrument de digitalisation est modifiée, les résultats de traitement ne seront pas affectés. Hormis les seuils ne concernant pas directement le traitement l'image, les seuls

qui ne sont pas définis comme des multiples de ***LMT*** sont ceux nécessaires aux traitements précédant l'évaluation de la *largeur moyenne du tracé*.

La mise en correspondance des graphes se fait par la méthode de l'étiquetage par relaxation. Généralement, cette technique est utilisée pour la segmentation d'images ou pour la recherche de sous-graphes dans un graphe. Pour ces deux applications, une constance demeure: le traitement s'effectue dans des environnements aux caractéristiques connues et fixes. Par exemple, dans une image, le nombre de voisins d'un pixel est le même partout dans l'image (sauf pour les pixels en bordure de l'image) et, de ce fait, la pondération pour chacun d'eux est toujours la même. De même, lorsqu'il s'agit de la recherche d'un sous-graphe, la topologie de ce sous-graphe et du graphe dans lequel se fait la recherche sont connus d'avance et restent les mêmes tout au long de la recherche.

Dans le cas qui nous concerne, il en va autrement. La tâche à accomplir peut être comparée à la recherche d'un sous-graphe, mais ici la topologie du graphe n'est pas nécessairement la même tout au long de la recherche. Nous savons que le sous-graphe recherché est composé de noeuds ayant un maximum de deux voisins. Cette contrainte a obligé le regroupement des interprétations des noeuds *développés* dans le *graphe de relations* en groupe de voisins de façon à ce que chaque noeud de ce graphe ait aussi un maximum de deux voisins. Ainsi, lors de la mise à jour des poids associés à chacun des appariements possibles, il faut considérer un seul poids par groupe de voisins plutôt que de

considérer l'ensemble des voisins. Aussi, puisqu'un noeud ne peut avoir qu'un maximum de deux voisins, lorsqu'une mise en correspondance est effectuée avec un noeud ayant été développé, les autres membres du groupe de noeuds concernés deviennent obsolètes. Ce qui a pour effet de modifier la topologie du graphe.

7.3 Limites et contraintes

Bien qu'innovatrice la méthode proposée n'en est pas moins soumise à certaines limites et contraintes. Évidemment, les premières limites sont celles découlant de la définition même du projet. Comme le projet s'inscrit dans un cadre de vérification de signatures, son champ d'application s'y limite. Le fait d'utiliser comme information *a priori* la banque de données de référence rend impossible l'utilisation intégrale de la méthode proposée à d'autres fins que celle de la vérification de signatures.

Une des contraintes volontairement imposées par l'auteur fut d'utiliser des signaux captés dans un environnement homogène et contrôlé. Les signatures ont été produites sur du papier blanc avec un slylo à bille à encre noire. Ce faisant, il n'était pas nécessaire d'implanter un étage supplémentaire de prétraitement pour éliminer le fond des images.

Cependant, malgré l'absence de fond dans les images, il semble que la méthode de binarisation choisie ne soit pas à la hauteur des attentes, particulièrement lorsque la signature est composée de traits qui, sans se toucher, sont très rapprochés. Il y aurait

avantage à concevoir une méthode de binarisation spécialement conçue pour le traitement des signaux manuscrits.

Cependant ce sont les résultats expérimentaux qui établissent les vraies limites du système. Dans l'état actuel des choses, le système ne permet pas d'inférer parfaitement la séquence dans une image. Une relecture des résultats de l'expérience 2, qui correspond à une simulation d'une application réelle, montre que la portion de séquence inférée peut varier de 55 % à 85 %.

Il est possible que, même avec un système parfait, le pourcentage de séquence inférée ne soit pas égal à 100 % car il peut exister des différences entre la séquence de référence et la séquence à l'origine de l'image de test. Toutefois, les observations empiriques montrent que les résultats actuels peuvent être améliorés.

Il faut tout de même noter que dans le pire des cas (55 %), le pourcentage de séquence inférée pour une signature authentique est toujours supérieur à celui observé pour un faux aléatoire. Cette observation montre une certaine consistance dans les résultats de l'inférence.

7.4 Nouvelles voies de recherche

Les résultats obtenus dans le cadre du présent projet ne constituent pas une fin en soi. Ils doivent plutôt servir de base à l'élaboration de projets futurs. Dans cette optique, deux possibilités se distinguent.

La première de ces possibilités consisterait à approfondir certains des aspects du traitement de façon à améliorer les performances du système. Les efforts devraient être concentrés sur la partie traitement de l'information statique, notamment la binarisation de l'image et la formation des zones caractéristiques, ainsi que sur la partie mise en correspondance.

Il ne faudrait pas non plus exclure la possibilité d'apporter des modifications majeures au système. Dans l'état actuel, il n'y a aucune rétroaction dans le traitement; une décision prise à un niveau ne peut être corrigée ou modifiée par une information obtenue subséquemment dans le traitement. Il serait peut-être intéressant de modifier le traitement de façon à permettre la rétroaction de l'information pour ainsi corriger certaines lacunes.

Dans le même ordre d'idée, des modifications pourraient également être apportées à la structure de l'information. Par exemple, une information de plus haut niveau pourrait faciliter la procédure de mise en correspondance. Par information de plus haut niveau,

nous entendons de l'information sur la présence de structures plus ou moins complexes dans l'image, telles que des boucles ou de courts traits.

L'autre possibilité consisterait à passer à une nouvelle étape dans le processus de la vérification de signatures. Cette nouvelle étape concernerait la prise de mesures permettant de statuer sur l'origine de l'image traitée, à savoir s'il s'agit d'une signature authentique ou non.

Une première technique envisageable pour réaliser cette tâche consisterait à comparer entre elles les portions de séquences appariées, en utilisant la technique de la *programmation dynamique*. Cette technique permettrait de calculer une distance entre les portions de séquences correspondantes. Cette distance serait la somme des poids associés à chacune des opérations d'édition nécessaires pour rendre les deux portions de séquences semblables.

La distance mesurée par *programmation dynamique* serait utilisée pour la classification, remplaçant ainsi les métriques proposées dans le présent travail. Cela permettrait également la validation du mécanisme de positionnement du squelette dans l'image, particulièrement pour les portions situées dans les *croisements* et les *boucles dégénérées*.

Afin de mieux exploiter l'information fournie par la séquence inférée, il serait également intéressant de développer un système d'extraction de profil de vitesse. Cette étape supplémentaire formerait le lien vers l'application des techniques de vérification *en-ligne*.

Bien évidemment il ne pourrait s'agir que de profils de vitesse *relative*, car la durée réelle de production d'un tracé ne peut être connue. La pertinence de ces profils résiderait dans des comparaisons *intra-séquence*, c'est-à-dire dans l'établissement de ratios entre différentes portions d'un même profil de vitesse. Parmi les techniques envisageables pour l'extraction de profils de vitesse, mentionnons, entre autres, la détection de portions d'éllipses ou le développement en série de Fourrier.

Peu importe les choix qui seront faits dans l'avenir, les travaux effectués dans le cadre du présent projet de maîtrise pourront servir de base à de futurs développements. Les performances actuelles étant encourageantes, toute modification constructive ne saurait que bonifier les efforts fournis pour arriver aux résultats présentés dans le présent mémoire.

BIBLIOGRAPHIE

[ABUH93] ABUHHAIBA, I.S.I. et AHMED, P., « Restoration of Temporal Information in Off-Line Arabic Handwriting », *Pattern Recognition*, vol. 26, no 7, p. 1009-1017, 1993.

[ABUH94] ABUHAIBA, I.S.I., HOLT, M.J.J. et DATTA, S., « Processing of Off-Line Handwritten Text: Polygonal Approximation and Enforcement of Temporal Information », *CVGIP: Graphical Models and Image Processing*, vol. 56, no 4, p. 324-355, 1994.

[ABUT89] ABUTALEB, A.S., « Automatic Thresholding of Gray-Level Pictures Using Two-Dimensional Entropy », *Computer Vision, Graphics, and Image Processing*, vol. 47, p 22-32, 1989.

[AMMA86] AMMAR, M., YOSHIDA, Y. et FUKUMURA, T., « A New Effective Approach for Off-Line Verification of Signatures by using Pressure Features », *Proceedings of the 8th International Conference on Pattern Recognition, Paris, France*, p. 566-569, 1986.

[AMMA88] AMMAR, M., YOSHIDA, Y. et FUKUMURA, T., « Description of Signature Images and its Application to their Classification », *Proceeding of the 9th Internationl Conference on Pattern Recognition, Rome, Italy*, p. 23-26, 1988.

[BABC88] BABCOCK, M. K. et FREYD, J. J., « Perception of dynamic information in static handwritten forms », *American journal of psychology*, vol. 101, no 1, p. 111-130, 1988.

[BOCC93] BOCCIGNONE, G., CHIANESE, A., CORDELLA, L.P. et MARCELLI, A., « Recovering Dynamic Information from Static Handwriting », *Pattern Recognition*, vol. 26, no 3, p. 409-418, 1993.

[BOLL79] BOLLES, R.C., « Robust feature matching through maximal cliques », *Imaging Applications for Automated Industrial Inspection & Assembly*, SPIE, vol. 182, p. 140-149, 1979.

[CHEN89] CHEN, Y.-S. et HSU, W.-H., « An Interpretative Model of Line Continuation in Human Visual Perception », *Pattern Recognition*, vol. 22, no 5, p. 619-639, 1989.

[CHER95] CHERIET, M., SAID, J.N. et SUEN, C.Y., « A Formal Model for Document Processing of Business Forms », *Proceedings of the 5th International Conference on Document Analysis and Recognition, Montréal, Canada*, p. 210-213, 1995.

[CHOU90] CHOUINARD, C., *Développement d'un nouvel algorithme de squelettisation de caractères manuscrits*, Mémoire de maîtrise. École Polytechnique de Montréal, 1990.

[CHOU92] CHOUINARD, C. et PLAMONDON, R., « Thinning and Segmenting Handwritten Characters by Line Following », *Machine Vision and Application*, vol. 5, p.185-197, 1992.

[CINQ96] CINQUE, L., YASUDA, D., SHAPIRO, L.G., TANIMOTO, S. et ALLEN, B., « An Improved Algorithm for Relational Distance Graph Matching », *Pattern Recognition*, vol. 29, no 2, p. 349-359, 1996.

[DOER91] DOERMANN, D.S. et ROSENFELD, A., *Recovery of Temporal Information from Static Images of Handwriting*. Rapport technique, Computer Vision Laboratory, Center for Automation Research, U. of Maryland, 29 p., 1991.

[DOER92] DOERMANN, D.S. et ROSENFELD, A., « Temporal Clues in Handwriting », *Proceedings of the 11th International Conference on Pattern Recognition, The Hague, Hollande*, p. 317-320, 1992.

[GERA89] GERALD, C.F. et WHEATLEY, P.O., *Applied Numerical Analysis, fourth edition*, Addition Wesley, 1989.

[HARA92] HARALICK, R.M. et SHAPIRO, L.G., *Computer and Robot Vision, volume 1*, Addison-Wesley, Reading, 1992.

[HUAN95] HUANG, W., RONG, G. et BIAN, Z., « Strokes Recovering from Handwriting Images », *Proceedings of the 5th International Conference on Document Analysis and Recognition, Montréal, Canada*, p. 861-864, 1995.

[HUMM83] HUMMEL, R.A. et ZUCKER, S.W., « On the Foundations of Relaxation Labelling Processes », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no 3, p. 267-286, 1983.

[KITT85] KITTLER, J. et ILLINGWORTH, J., « Relaxation labelling algorithms - a review », *Image and vision computing*, vol.3, no 4, p. 206-216, 1985.

[LAM 92] LAM, L., LEE, S.-W. et SUEN, C., « Thinning Methodologies - A Comprehensive Survey », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no 9, p. 869-885, 1992.

[LECL94] LECLERC, F. et PLAMONDON, R, « Automatic Signature Verification: The State of the Art - 1989-1993 », *Progress in Automatic Signature Verification, Series in Machine Perception and Artificial Intelligence*, vol. 13, World Scientific Publishing, 1994.

[LEE 89] LEE, S. et PAN, J.C., « Tracing and Representation of Human Line Drawings », *Proceedings of the International Conference on System, Man and Cybernetics*, p. 1055-1061, 1989.

[LEE 92] LEE, S. et PAN, J.C., « Offline Tracing and Representation of Signatures », *IEEE Transactions On Systems, Man, and Cybernetics*, vol. 22, no 4, p. 755-771, 1992.

[LEYM89] LEYMARIE, F. et LEVINE, M.D., « Shape features using curvature morphology », *Visual Communications and Image Processing IV*, SPIE vol. 1199, p. 390-401, 1989.

[MEUL91] MEULENBROEK, R.G.J. et THOMASSEN, A.J.W.M., « Stroke-direction preference in drawing and handwriting », *Human Movement Science*, vol. 10, p. 247-270, 1991.

[OGOR88a] O'GORMAN, L., « An Analysis of Feature Detectability from Curvature Estimation », *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, p. 235-240, 1988.

[OGOR88b] O'GORMAN, L., « Curvilinear Feature Detection from Curvature Estimation », *Proceeding of the 9th International Conference on Pattern Recognition*, p. 1116-1119, 1988.

[OTSU79] OTSU, N., « A Treshold Selection Method from Gray-Level Histograms », *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no 1, p.62-66, 1979.

[PARI89] PARIZEAU, M. et PLAMONDON, R., « What types of scripts can be used for Personal Identity Verification? », *Computer Recognition and Human Production of Handwriting*, World Scientific Publishing, p. 77-90, 1989.

[PLAM88] PLAMONDON, R. et PARIZEAU, M., « Signature Verification from Position, Velocity and Acceleration Signals: A Comparative Study », *Proceedings of the 9th International Conference on Pattern Recognition*, 1988.

[PLAM89] PLAMONDON, R. et LORETTE, G., « Automatic Signature Verification and Writer Identification - The State of the Art », *Pattern Recognition*, vol. 22, no 2, p. 107-131, 1989.

[PLAM92a] PLAMONDON, R., « A Model-Based Segmentation Framework for Computer Processing of Handwriting », *Proceedings of the 11th International Conference on Pattern Recognition, The Hague, Hollande*, p. 303-307, 1992.

[PLAM92b] PLAMONDON, R., YERGEAU, P. et BRAULT, J.J., « A multi-level signature verification system », *From Pixels to Features III: Frontiers in Handwriting Recognition*, Elsevier Science Publishers, p. 363-370, 1992.

[PRES92] PRESS, W.H., TEUKOLSKY, S.A., VETTERLING, W.T. et FLANNERY, B.P., *Numerical recipes in C the art of Scientific Computing*, second edition, Cambridge university press, 1992.

[PRIV95] PRIVITERA, C.M. et PLAMONDON, R., A System for Scanning and Segmenting Cursive Handwritten Words into Basic Strokes », *Proceedings of the Third International Conference on Document Analysis and Recognition*, p. 1047-1050, 1995.

[ROSE75] ROSENFELD, A., « A Characterization of Parallel Thinning Algorithms », *Information and Control*, vol. 29, p 286-291, 1975.

[ROSE77] ROSENFELD, A. et KAK, A.C., *Digital Picture Processing*, Academic Press, New-York, 1977.

[SABO94] SABOURIN, R., PLAMONDON, R. et BEAUMIER, L., « Structural Interpretation of Handwritten Signature Images », *Progress in Automatic Signature Verification, Series in Machine Perception and Artificial Intelligence*, vol. 13, World Scientific Publishing, 1994.

[SERR82] SERRA, J.-P., *Image Analysis and Mathematical Morphology*, Academic Press, 1982.

[SHAP90] SHAPIRO, L.G. et HARALICK, R.M., « Matching Relational Structures using Discrete Relaxation », *Series in Computer Science - Vol. 7, Syntactic*

and Structural Pattern Recognition, Theory and applications, World Scientific, p. 179-195, 1990.

[SIMO91] SIMON, J.-C. et ZERHOUMI, K., « Robust Description of a Line Image », *Proc. of the First International Conference on Document Analysis and Recognition*, vol. 1, p. 3-14, 1991.

[TRIE95] TRIER, Ø. D. et TAXT, T., « Evaluation of Binarisation Methods for Document Images », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no 3, p. 312-315, 1995.

[THOM79] THOMASSEN, A.J.W.M. et TEULINGS, H.-L. H. M., « The Development of Directional Preference in Writing Movements », *Visible Language*, Vol. 13, no 3, p.299-313, 1979.

[WALL88] WALLACE, A.M., « A Comparison of Approaches to High-Level Image Interpretation », *Pattern Recognition*, vol. 21, no 3, p. 241-259, 1988.

[YANO89] YANOWITZ, S.D. et BRUCKSTEIN, A.M., « A New Method for Image Segmentation », *Computer Vision, Graphics, and Image Processing*, vol. 46, p. 82-95, 1989.

ANNEXE A Méthode d'estimation de la largeur moyenne du tracé LMT

La largeur moyenne du tracé *LMT* est estimée pour servir d'unité de base pour la détermination de certains paramètres de traitement. Ceci a pour objet de rendre ces paramètres variables en fonction de la nature de l'information traitée.

L'information servant à l'estimation de *LMT* est constituée des points extremums identifiés sur les contours de la signature. Il faut, dans un premier temps, identifier tous les couples de points potentiellement appariables. Cette identification se fait de la même façon que pour la formation des zones extremums [section 2.3.3]. Pour qu'un couple de points extremums soit potentiellement appariable, il doit rencontrer les critères suivants :

1) Le couple doit être formé de deux extremums de types complémentaires.
2) La droite reliant les deux points du couple doit être complètement incluse dans le tracé.

Tous les couples de points qui rencontrent ces deux critères servent à l'estimation de *LMT*. Une mesure de distance est associée à chacun de ces couples. Cette mesure correspond à la distance euclidienne d entre les points, projetée sur l'axe correspondant au type des points concernés [Figure A.1].

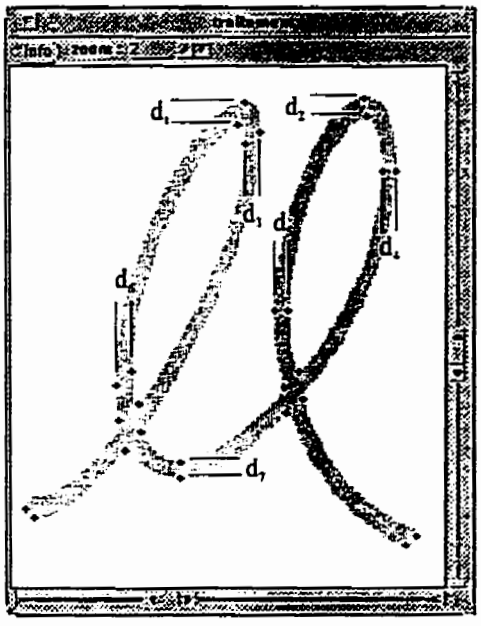


Figure A.1 Illustration de la mesure de distance *d* entre les points pour l'estimation de *LMT*. La distance est évaluée selon l'axe correspondant au type d'extremums formant le couple.

Autrement dit, pour les N couples de points extremums en X (en Y), la distance d_i correspond à la différence de coordonnées en X (en Y) entre les points du i^e couple. La valeur de *LMT* correspond donc à la moyenne de ces mesures de distance.

$$LMT = \frac{1}{N}\sum_{i=1}^{N} d_i \qquad \text{(A-1)}$$

La première estimation de *LMT* est faite avant le filtrage des points extremums [section 2.3.2.2]. Il est alors possible que certains des couples soient formés de points non valides. Cette première estimation, notée LMT_E, permet de fixer le seuil τ_{cont} et donc de filtrer P_{ext}.

Après filtrage, les zones extremums Z_{ext} sont formées et LMT est estimé de nouveau à partir des zones formées. C'est cette valeur qui servira de référence pour tout le reste du traitement.

ANNEXE B Choix des paramètres M et W pour le calcul de la courbure

La méthode retenue pour le calcul de la courbure est celle proposée par O'Gorman [OGOR88a]. La valeur de courbure associée à chaque point p d'une courbe est déterminée par la variation angulaire θ_c mesurée entre deux droites (P_1P_2 et S_1S_2) situées de part et d'autre du point p à l'étude [Figure B.1].

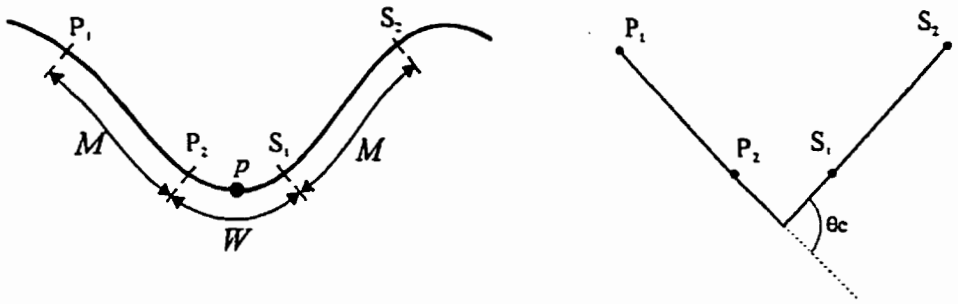


Figure B.1 Portion de contour et positionnement des paramètres nécessaires au calcul de la variation angulaire.

La position de ces deux droites est déterminée par les paramètres suivants :

M = l'intervalle entre les deux droites,

W = la longueur de ces droites.

La longueur totale L de la portion de contour nécessaire à l'évaluation de la variation angulaire se définit donc par:

$$L = 2W + M \qquad \text{(B-1)}$$

La valeur du paramètre M est fixée à 0. Pour la valeur du paramètre M, O'Gorman montre dans [OGOR88a] que des valeurs de M près mais différentes de zéro maximisent le rapport signal à bruit. Cependant, selon la configuration des contours traités, il est possible qu'en fixant M à une valeur différente de zéro, il y ait des valeurs de θ_c qui excèdent la plage de ±180° [Figure B.2]. Cette situation est évitée en fixant M=0, même si cela a comme effet de diminuer le rapport signal à bruit.

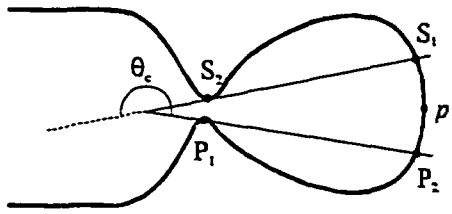


Figure B.2 Illustration d'une portion de contours problématique. La différence de trajectoire entre les droites P_1P_2 et S_1S_2 donne un angle de courbure θ_c pour le point p plus grand que 180°.

Le choix de valeur du paramètre W doit se faire en fonction du contexte dans lequel s'effectue la mesure. Ce paramètre est ajusté en tenant compte à la fois de la longueur du contour à l'étude et de la valeur de LMT, la *largeur moyenne du tracé* [Annexe A].

La taille totale du contour l_c permet de fixer une limite supérieure pour L. La valeur de la variation angulaire ne doit pas être évaluée en utilisant plus que la moitié de la longueur totale du contour, l_C. Donc, la limite supérieure sur L est:

$$L \leq \frac{l_C}{2} \qquad \text{(B-2)}$$

Cette limite exprimée en fonction de W (et en considérant la valeur de M=0) devient:

$$W \leq \frac{l_C}{4} \qquad \text{(B-3)}$$

La valeur de LMT permet de fixer la limite inférieure de la valeur de L. Cette limite assure que les valeurs de courbure atteignent un certain niveau dans les extrémités de tracé. La justification de cette limite inférieure est faite à partir de la représentation idéale d'une extrémité de composante [Figure B.3]. Une extrémité idéale de trait est caractérisée par un rayon de courbure $r = LMT/2$. La portion de contour correspondant au demi-cercle de l'extrémité est donc de longueur $c = \pi r$.

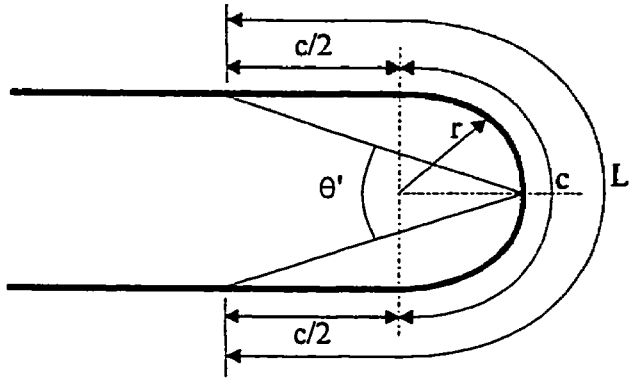


Figure B.3 Extrémité théorique de composante.

Afin de maximiser la valeur de la variation angulaire dans les extrémités de composante, la portion de contour utilisée pour le calcul doit excéder la portion courbe du contour dont la longueur est c. En fixant la limite inférieure à:

$$L \geq 2c \qquad \text{(B-4)}$$

L'angle maximum calculé pour une extrémité parfaite est d'environ 320°. En exprimant c en fonction de LMT, la limite inférieure de la valeur de L devient donc:

$$L \geq \pi LMT \qquad \text{(B-5)}$$

Cette limite, exprimée par rapport à W, est:

$$W \geq \frac{\pi LMT}{2} \qquad \text{(B-6)}$$

En exprimant la valeur de W comme un multiple n de LMT, ce multiple doit donc être:

$$n \geq \frac{\pi}{2} \qquad \text{(B-7)}$$

De façon à satisfaire l'équation précédente, la valeur de W est fixée à:

$$W = 2\,LMT \qquad \text{(B-8)}$$

La valeur de W est déterminée pour chaque contour de l'image à l'aide de l'équation suivante:

$$W = \min(2\,LMT, \max(1, \frac{l_c}{4})) \qquad \text{(B-9)}$$

Le paramètre W est donc égal à 2 LMT pour tous les contours de longueur supérieure à 8 LMT. Pour les contours plus petits, W est fixé à $\frac{l_c}{4}$ de façon à ce que la portion de contour L n'excède pas la moitié du contour. La plus petite valeur que peut prendre W est l'unité, étant donné le caractère discret de la liste de points.

ANNEXE C Choix du seuil amplitude $\tau_{\Delta\theta}$ pour la validation des points de discontinuités

Il s'agit ici de déterminer la valeur du seuil $\tau_{\Delta\theta}$ pour la validation des points de discontinuité. Après avoir trouvé toutes les régions de discontinuité par filtrage morphologique, il faut identifier celles qui sont significatives. Sont considérées comme significatives toutes les régions dont l'amplitude de la courbure dépassera un certain seuil $\tau_{\Delta\theta}$. Pour le choix de ce seuil, une portion de contour représentant un croisement idéal est considérée [Figure C.1]. Il s'agit de deux traits rectilignes se croisant selon un angle θ_1.

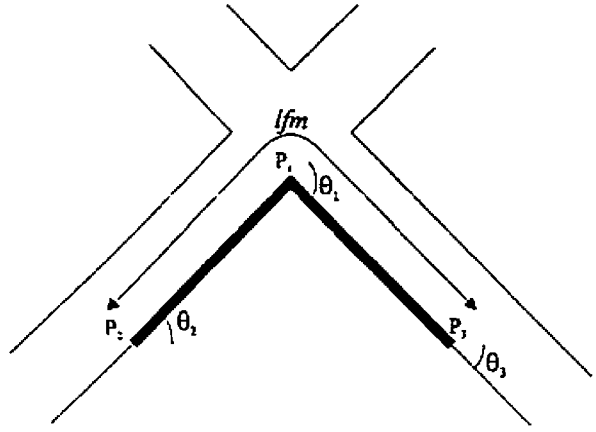


Figure C.1 Région de discontinuité dans un croisement idéal. Le trait foncé représente la région de discontinuité identifiée après le filtrage morphologique. Dans un tel cas, θ_2 et θ_3 sont égaux et le maximum de courbure est atteint au point P_1.

La largeur du filtre morphologique (*lfm*) est déterminée de la même façon que le paramètre *W* pour le calcul de la courbure [Annexe B].

Dans une situation idéale, la largeur de la région de discontinuité est égale à *lfm*. La variation angulaire au point P_1 est égale à θ_1. Les points P_2 et P_3 représentent les points limites de la région de discontinuité. Puisque *W=lfm*, le point P_3 constitue l'extrémité d'une des deux droites servant au calcul de variation angulaire au point P_2, même chose pour P_2 par rapport à P_3. Dans un cas idéal, les angles à ces deux points sont égaux et valent θ_2 ($\theta_3 = \theta_2$). Puisque ces angles représentent les plus grands angles de la région de discontinuité et θ_1, le plus petit, l'amplitude angulaire de la région de discontinuité s'exprime par:

$$\Delta\theta = |\theta_1 - \theta_2| \qquad \text{(C-1)}$$

En substituant dans cette équation l'expression de θ_2 en fonction de θ_1 qui, selon les propriétés du triangle, donne:

$$\theta_2 = \frac{\theta_1}{2} \qquad \text{(C-2)}$$

Nous obtenons l'expression de l'amplitude en fonction de l'angle θ_1:

$$\Delta\theta = \left|\frac{\theta_1}{2}\right| \qquad \text{(C-3)}$$

Donc, en théorie, l'amplitude $\Delta\theta$ de la variation angulaire dans une région de discontinuité est égale à la moitié de la valeur maximum de variation angulaire rencontrée dans cette région. La valeur du seuil $\tau_{\Delta\theta}$ doit donc être fixée en fonction du plus petit angle de croisement désirant être détecté.

Puisque le croisement de deux traits présente toujours un angle de croisement $\theta \geq 90°$, le choix d'un seuil $\tau_{\Delta\theta} = 45°$ permettrait à tout le moins de positionner des points de discontinuité sur les contours des parties aiguës du croisement. Cependant, le choix d'une valeur de seuil plus petite permettrait également de détecter les points de discontinuité dans les parties obtuses des croisements. Le caractère discret des images traitées combiné à une valeur de seuil $\tau_{\Delta\theta}$ trop petite pourrait mener à la confusion des régions de croisement et des portions de tracé courbes. L'analyse empirique semble favoriser une valeur de seuil égale à:

$$\tau_{\Delta\theta} = 15° \qquad \text{(C-4)}$$

ANNEXE D Méthode d'évaluation de l'indice de visibilité $v(p_i,p_j)$

L'indice de visibilité permet de quantifier la position relative entre deux points de contour. Cette quantification est à la base de la formation des ensembles de points caractéristiques qui forment les zones. Le terme *visibilité* provient de l'analogie faite avec deux individus qui se regardent. Dans un tel cas, la *visibilité* qu'ont les deux individus l'un de l'autre dépend à la fois de la *distance* qui les sépare et de la *direction* dans laquelle chacun regarde. L'indice de visibilité proposé tient compte de facteurs similaires.

La *distance* est donc le premier facteur considéré dans la mesure $v(p_i,p_j)$. Plus les points sont près, plus l'indice doit avoir une valeur élevée. Inversement, plus les points sont éloignés, plus la valeur de l'indice de visibilité doit être faible, jusqu'à atteindre une valeur nulle lorsque la distance dépasse une distance d_{max} [Annexe E]. Une fonction $fd_{i,j}$ permettant de pondérer la distance entre les points p_i et p_j, situés en (x_i,y_i) et (x_j,y_j) respectivement, est donc définie comme suit:

$$fd_{i,j} = \begin{cases} \dfrac{d_{\max} - d(p_i, p_j)}{d_{\max}} & si \quad d(p_i, p_j) \le d_{\max} \\ 0 & autrement \end{cases} \qquad \text{(D-1)}$$

où:

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2} \qquad \text{(D-2)}$$

Le second facteur impliqué dans la mesure de visibilité est la *direction*. En plus de la distance, la visibilité entre deux observateurs est influencée par la direction du regard et par les limites du champ de vision de chacun. Transposée aux points, la direction du « regard » correspond à un vecteur perpendiculaire au contour et inclus dans le tracé [Figure D.1a].

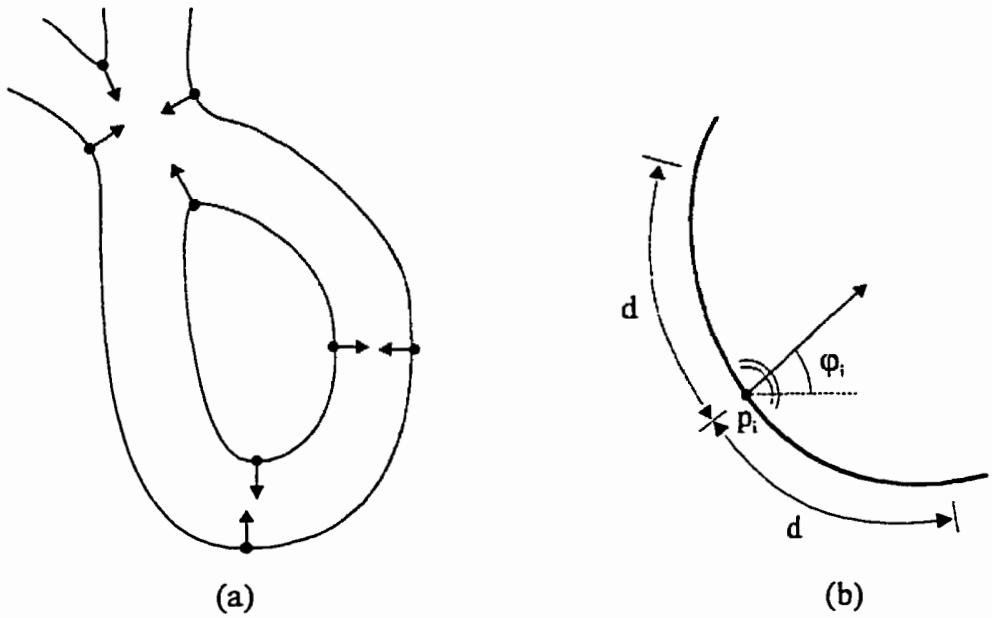


Figure D.1 (a) Exemple des vecteurs utilisés comme analogie à la notion de « direction du regard ». (b) Technique utilisée pour déterminer l'orientation du vecteur perpendiculaire.

L'orientation φ du vecteur est estimée en utilisant une technique différentielle inspirée de celle utilisée pour le calcul de la courbure. Deux droites sont positionnées de part et d'autre du point à l'étude. Chaque droite passe par le point p_i à l'étude et par un point situé à une distance $\pm d$ sur le contour [Figure D.1b]. L'orientation φ du vecteur perpendiculaire est déterminée par la bissectrice de l'angle formé par ces deux droites. Ce résultat est corrigé de π si le vecteur résultant n'est pas inclus dans le tracé.

Les vecteurs ainsi formés permettent l'évaluation de fa_n, la seconde fonction impliquée dans le calcul de l'indice de visibilité. Cette fonction s'inspire du concept du champ de vision: plus l'objet observé est près du point d'attention du regard, plus il est visible. La valeur de la fonction fa_n, pour un point donné n, est donc déterminée à partir de la distance angulaire θ_n entre l'objet observé et le vecteur associé au point n.

Puisque l'indice de visibilité se mesure entre deux points, la fonction fa_n doit être estimée pour les deux points concernés, p_i et p_j. Dans l'exemple de la Figure D.2, la valeur de fa_i est établie en fonction de l'angle θ_i formé par le vecteur perpendiculaire au point p_i et la droite reliant ce point au point p_j. De la même façon, la valeur de fa_j, pour le point p_j, dépendra de l'angle θ_j.

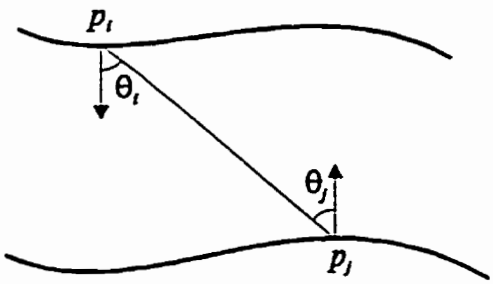


Figure D.2 Illustration des angles formés par les perpendiculaires et la droite reliant les points p_i et p_j. Ces angles servent à l'évaluation des fonctions fa_i et fa_j.

Plus les angles θ sont près de 0°, plus la visibilité est bonne. Et tout comme pour le champ de vision des observateurs, au-delà d'un certain angle, la visibilité devient nulle. La limite est ici fixée à $\pm\pi/2$. L'équation traduisant un tel comportement correspondant à la fonction fa_n est donnée ci-après:

$$fa_n = \begin{cases} \dfrac{\frac{\pi}{2} - |\theta_n|}{\frac{\pi}{2}} & si \ |\theta_n| < \frac{\pi}{2} \\ 0 & autrement \end{cases} \quad , n = i, j \qquad \text{(D-3)}$$

L'indice de visibilité $v(p_i,p_j)$ est une combinaison des fonctions d'orientation fa_i et fa_j, estimées pour les points p_i et p_j, et de la fonction de distance $fd_{i,j}$. Ces trois mesures sont combinées de la façon suivante:

$$v(p_i, p_j) = fd_{i,j} \times fa_i^{\,2} \times fa_j^{\,2} \qquad \text{(D-4)}$$

Les trois fonctions prenant des valeurs entre 0 et 1, le résultat est donc nécessairement un nombre compris entre 0 et 1. Puisque les trois fonctions sont multipliées, elles doivent toutes trois être différentes de zéro pour que la mesure de visibilité soit elle-même différente de zéro. De cette façon chaque fonction joue un rôle déterminant sur la valeur de $v(p_i,p_j)$. L'élévation à la puissance 2 dans le cas des fonctions d'orientation fa_i et fa_j a pour effet de les rendre plus discriminantes encore: la distance angulaire entre les vecteurs devient plus significative que la distance euclidienne qui sépare les points.

La Figure D.3 illustre le comportement de l'indice en fonction de la position relative des points. L'indice est calculé en considérant toutes les positions possibles du point p_j par rapport au point p_i et ce, pour des orientations φ_i et φ_j des perpendiculaires aux points p_i et p_j constantes.

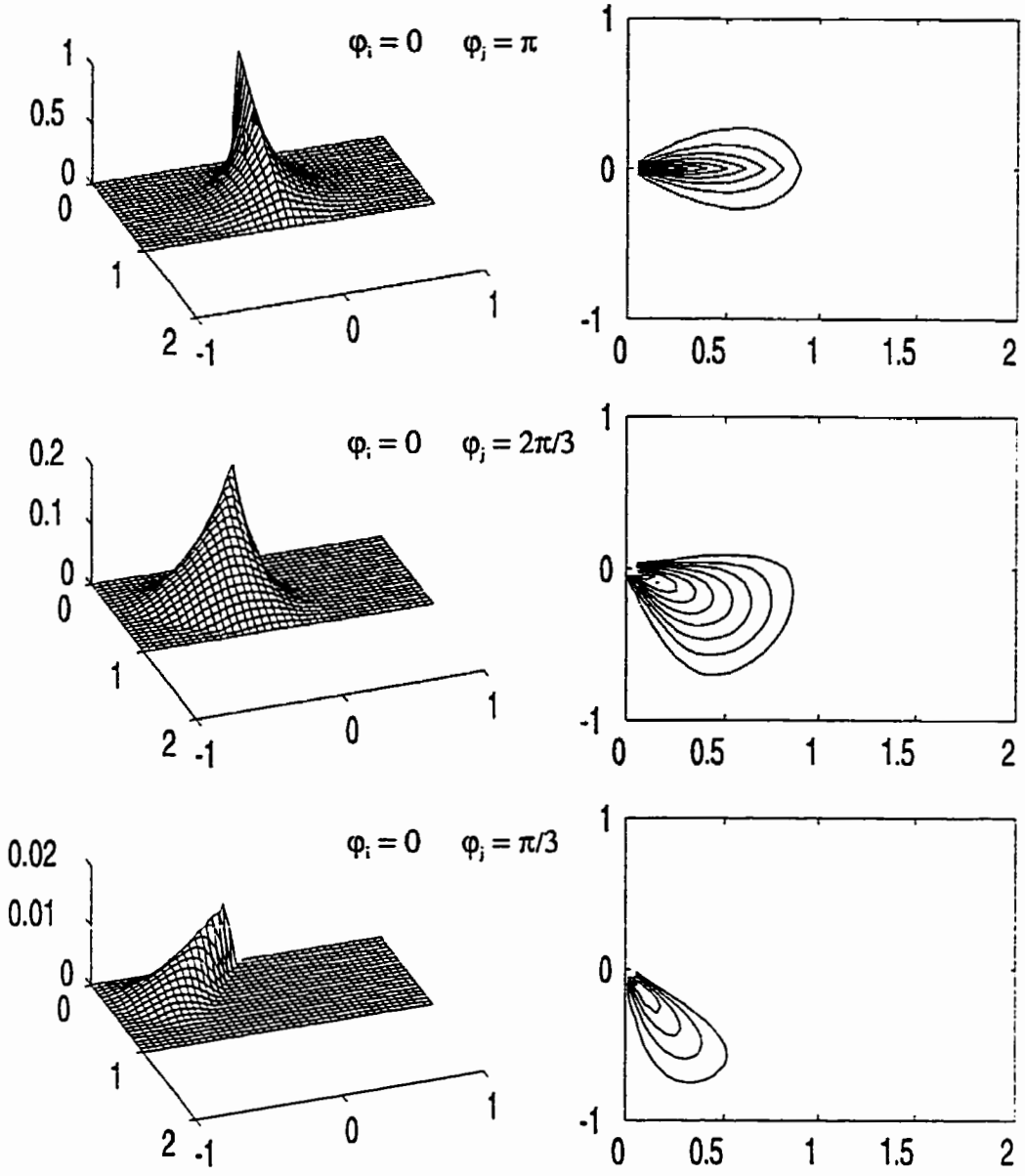


Figure D.3 Comportement de l'indice de visibilité en fonction de la position relative des points. (Tous ces graphiques sont exprimés en fonction de d_{max}).

ANNEXE E Choix du paramètre d_{max} pour l'évaluation de l'indice de visibilité $v(p_i,p_j)$

Dans l'évaluation de l'indice de visibilité $v(p_i,p_j)$, le paramètre d_{max} représente la distance au-delà de laquelle $v(p_i,p_j)$ est nulle. Ce paramètre prend différentes valeurs selon le type des points pour lesquels l'indice de visibilité v est estimé. Afin de rendre le choix de ces valeurs dépendant du contexte, d_{max} est toujours multiple de la mesure *LMT*.

Les points les plus distants qui sont appelés à être associés sont les points de discontinuité de la partie aiguë d'un croisement de traits. La valeur de d_{max} est estimée à partir de la représentation d'un croisement de trait [Figure E.1].

Les deux traits constituant ce tracé sont de largeurs identiques, égales à *LMT*, et se croisent selon un angle minimum α. La valeur de d représente la distance entre les points D1 et D2, les points de discontinuité les plus éloignés. Le choix de d_{max} doit être tel que la mesure de visibilité soit non nulle lorsque les points sont séparés par une distance d ($d_{max} > d$).

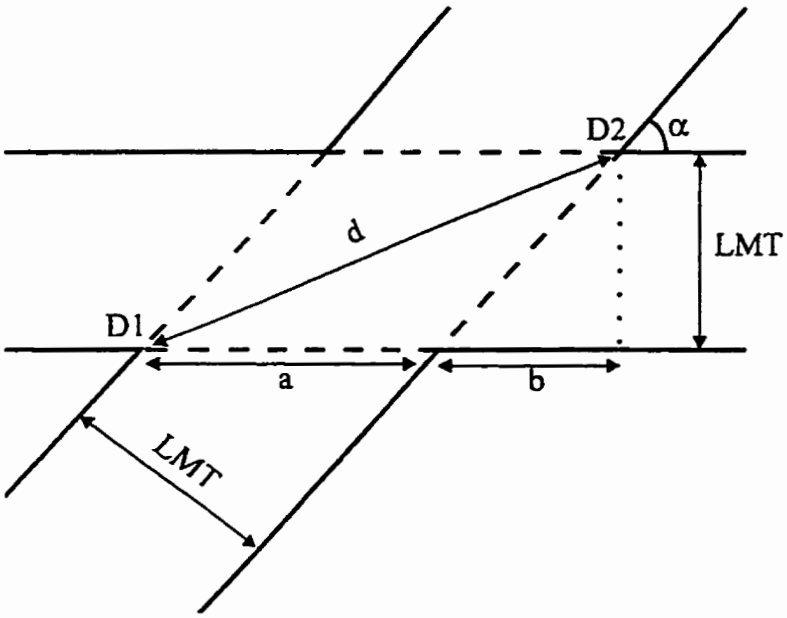


Figure E.1 Représentation idéale d'un croisement de traits avec un angle $\alpha < 90°$.

Nous cherchons à exprimer d en fonction de LMT et α. Sachant que d est égal à:

$$d = \sqrt{(a+b)^2 + LMT^2} \qquad \text{(E-1)}$$

où:

$$a = \frac{LMT}{\sin\alpha} \qquad \text{(E-2)}$$

$$b = a\cos\alpha \qquad \text{(E-3)}$$

l'équation (E-1) devient:

$$d = LMT\sqrt{\left(\frac{1+\cos\alpha}{\sin\alpha}\right)^2 + 1} \qquad \text{(E-4)}$$

En considérant une valeur minimum pour l'angle de croisement α de 15°, la valeur de d est donc:

$$d = 8\,LMT \qquad \text{(E-5)}$$

Cette distance correspond à la distance théorique séparant les points de discontinuité les plus éloignés dans un croisement de traits à 15°. En pratique, cette distance sera supérieure à 8 *LMT*, notamment à cause de la procédure de *comblement des petits trous* [section 2.2.2.3], comme le montre la Figure E.2, qui illustre le contour de la partie aiguë d'un croisement de traits à un angle de 15°.

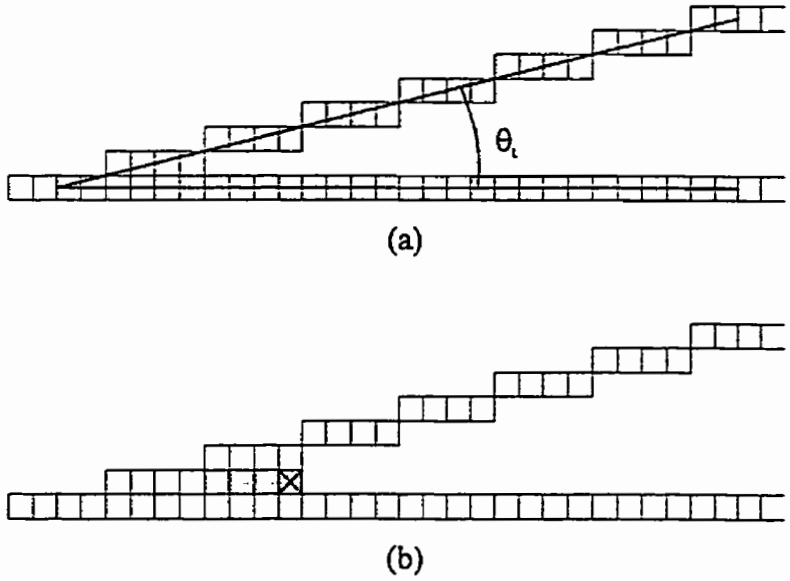


Figure E.2 Illustration de la variation entre la position du contour avant et après la procédure de *comblement des petits trous*. Le contour théorique est montré en (a). Les pixels ombrés en (b) sont ceux ajoutés au tracé par la procédure de *comblement des petits trous*. Le nouveau contour est représenté par les pixels tracés en noir.

Considérant l'augmentation de la distance d causée par la procédure de *comblement des petits trous* et considérant le fait qu'il faut que la mesure de visibilité ne soit pas nulle pour des points de discontinuité séparés par une telle distance, le choix de la distance d_{max} est fixé à:

$$d_{max} = 20\ LMT \qquad \text{(E-6)}$$

De cette façon, la mesure de visibilité $v(D1,D2)$ dans une situation telle que celle de la Figure E.1 serait d'environ 0.5.

ANNEXE F Implantation de la méthode d'interpolation par splines cubiques

La technique d'interpolation par *splines cubiques* [GERA89] est utilisée afin de générer une séquence continue de points qui représentent la *séquence interpolée* D_{int} à partir de la liste de *points échantillonnés* D_e produite par la tablette à digitaliser.

La méthode des *splines cubiques* consiste à déterminer, pour chaque paire de points de contrôle successifs, les coefficients d'une fonction du 3e degré permettant de relier les deux points de contrôle concernés. Une condition de continuité est imposée à chacun des points de contrôle. La méthode des *splines cubiques* permet donc d'interpoler une courbe qui passe par tous les points de contrôle et sans présenter de discontinuité à ces points.

Supposons une liste de N points de contrôle. La fonction de degré 3 pour le i^e intervalle, soit entre les points (x_i,y_i) et (x_{i+1},y_{i+1}), est de la forme:

$$y = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \qquad \text{(F-1)}$$

Puisque la fonction passe par les deux points de contrôle qui sont:

$$y_i = d_i \tag{F-2}$$

$$\begin{aligned} y_{i+1} &= a_i(x_{i+1} - x_i)^3 + b_i(x_{i+1} - x_i)^2 + c_i(x_{i+1} - x_i) + d_i \\ &= a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i \end{aligned} \tag{F-3}$$

où h_i symbolise la variation en x pour le i^e intervalle. La condition de continuité est obtenue en imposant l'existence de la première et seconde dérivée des fonctions de part et d'autre d'un point de contrôle. Les expressions de ces dérivées sont:

$$y' = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \tag{F-4}$$

$$y'' = 6a_i(x - x_i) + 2b_i \tag{F-5}$$

En exprimant la dérivée seconde au point (x_i, y_i) par S_i et celle au point (x_{i+1}, y_{i+1}) par S_{i+1} de l'équation précédente, nous obtenons:

$$S_i = 2b_i \tag{F-6}$$

$$S_{i+1} = 6a_i h_i + 2b_i \tag{F-7}$$

En isolant a_i et b_i, cela devient:

$$b_i = \frac{S_i}{2} \tag{F-8}$$

$$a_i = \frac{(S_{i+1} - S_i)}{6h_i} \tag{F-9}$$

En substituant les relations obtenues pour a_i, b_i et d_i dans l'équation (27), nous obtenons pour c_i la relation suivante:

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{(2S_i + S_{i+1})h_i}{6} \qquad \text{(F-10)}$$

Pour qu'il y ait continuité, il faut que les dérivées premières des fonctions qui se rejoignent au point de contrôle (x_i,y_i) soient les mêmes. L'expression de cette dérivée au point (x_i,y_i) pour la fonction dans l'intervalle i est:

$$y_i' = 3a_i(x_i - x_i)^2 + 2b_i(x_i - x_i) + c_i = c_i \qquad \text{(F-11)}$$

La dérivée à ce même point pour la fonction dans l'intervalle précédent est:

$$\begin{aligned} y_i' &= 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1} \\ &= 3a_{i-1}h_{i-1}^{\ 2} + 2b_{i-1}h_{i-1} + c_{i-1} \end{aligned} \qquad \text{(F-12)}$$

En égalant les deux équations précédentes et en remplaçant a,b,c, et d par leur expression en termes de S et h, nous obtenons:

$$h_{i-1}S_{i-1} + (2h_{i-1} + 2h_i)S_i + h_iS_{i+1} = 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right) \qquad \text{(F-13)}$$

Cette équation montre que la dérivée seconde au i^e point de contrôle (S_i) dépend des dérivées secondes aux points de contrôle précédent (S_{i-1}) et suivant (S_{i+1}). Considérant une

liste de $N+1$ points de contrôle, l'équation précédente permet de définir un système de N-1 équations pour $1 \leq i \leq N$-1. Les dérivées secondes S_0 et S_N doivent être définies arbitrairement. Poser S_0=0 et S_N=0 suppose que les cubiques deviennent linéaires près des extrémités. Cela correspond au comportement désiré pour les extrémités de trait.

Une fois le système d'équations résolu, il est alors possible de déterminer les paramètres de chacune des N fonctions d'ordre 3 associées à chacun des N intervalles. Ces paramètres sont définis par:

$$a_i = \frac{(S_{i+1} - S_i)}{6h_i}$$

$$b_i = \frac{S_i}{2}$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{(2S_i + S_{i+1})h_i}{6}$$

$$d_i = y_i$$

Maintenant qu'une fonction est définie entre chaque paire de points de contrôle successifs, il ne reste qu'à procéder à l'interpolation dans chacun des intervalles entre les points de contrôle. Puisque le signal dynamique ne correspond généralement pas à la définition d'une fonction du type $y=f(x)$, puisqu'il y a des mouvements de retour du tracé, le signal dynamique doit être décomposé en deux signaux, soit:

$$x = f_x(n)$$

$$y = f_y(n)$$

où l'axe n représente l'ordre d'acquisition des points échantillons. Une représentation graphique de cette décomposition du signal est donnée à la Figure F.1. Puisque l'acquisition se fait à une fréquence constante, l'intervalle entre chacun des points sur l'axe n est considéré constant et égal à l'unité. La coordonnée (x,y) d'un point interpolé est établie pour une valeur de n donnée en procédant à l'interpolation dans les deux fonctions définies pour l'intervalle correspondant, chacune fournissant un des membres du couple de la coordonnée (x,y).

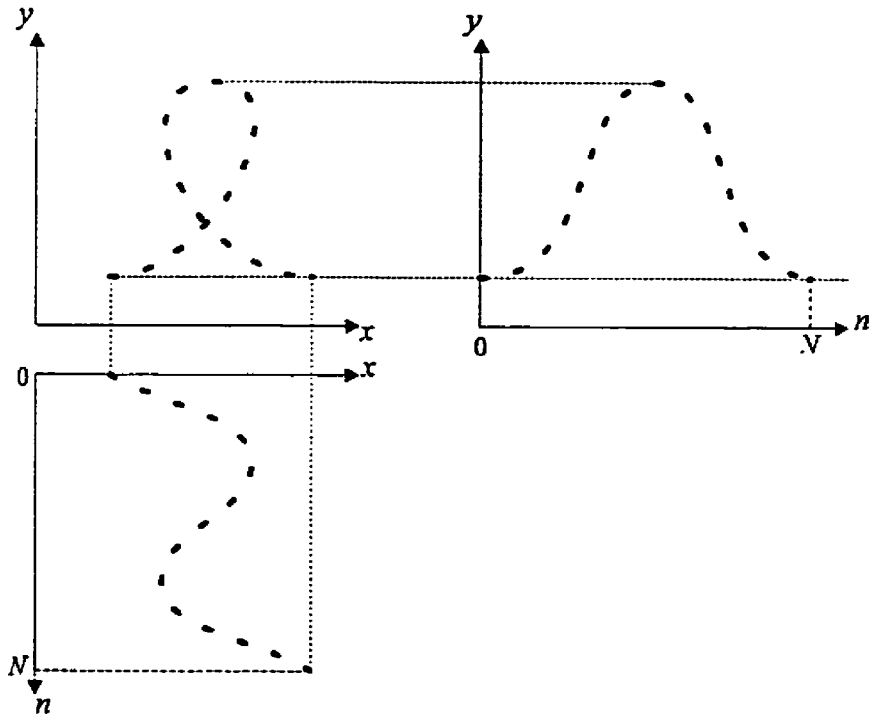


Figure F.1 Décomposition du signal dynamique en deux fonctions $x=f(n)$ et $y=f(n)$, afin de rendre possible l'interpolation par splines cubiques.

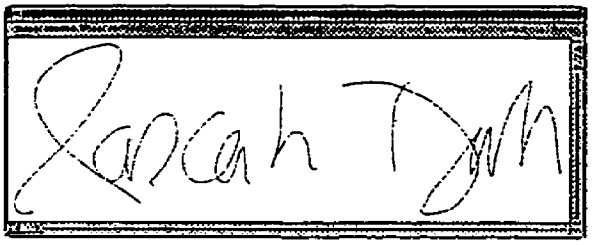
Le nombre de points à interpoler entre chaque paire de points de contrôle dépend de la distance séparant ces points: il faut insérer des points jusqu'à ce que tous les points aient deux voisins (considérant une connexité 8). Chaque nouveau point est inséré au milieu d'un intervalle à combler, le séparant ainsi en deux intervalles. La procédure est reprise tant que tous les intervalles ne sont pas comblés. Un intervalle entre deux points est considéré comblé si les deux points sont voisins. La procédure suivie pour l'interpolation est définie dans le tableau suivant.

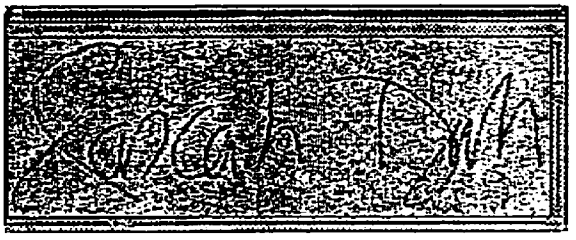
Tableau F.1 Procédure d'interpolation récursive par la méthode des splines cubiques.

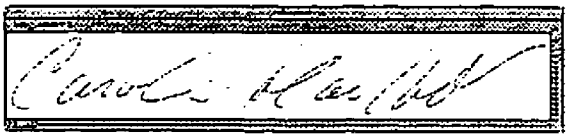
```
1:  FONCTION Interpolation(p_i,p_{i+1})
2:      SI les points p_i et p_{i+1} ne sont pas voisins ALORS
3:          Identifier position d'interpolation n_j = (n_i + n_{i+1}) / 2
4:          Interpoler coordonnée x_j = f_x(n_j)
5:          Interpoler coordonnée y_j = f_y(n_j)
6:          Insérer nouveau point p_j de coordonnée (x_j,y_j,n_j) entre les points p_i et p_{i-1}
7:          Interpolation(p_i,p_j)
8:      SINON
9:          Interpolation(p_{i+1},p_{i+2})
10:     FIN SI
11: FIN FONCTION
```

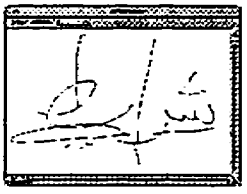
ANNEXE G Composition de la banque de données

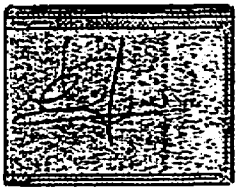
Les prochaines pages montrent des échantillons tirés de la banque de données utilisée. Pour chacun des 20 scripteurs, les deux représentations d'une même signature sont montrées, soit le signal dynamique (colonne de gauche) et l'image de luminance (colonne de droite).

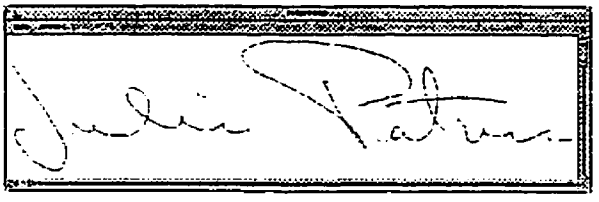


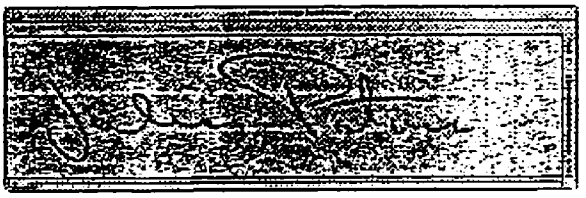


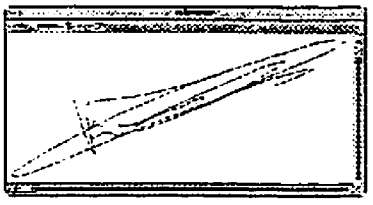


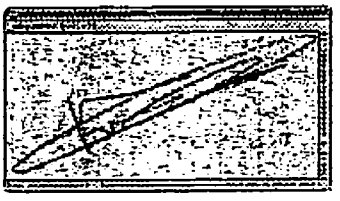


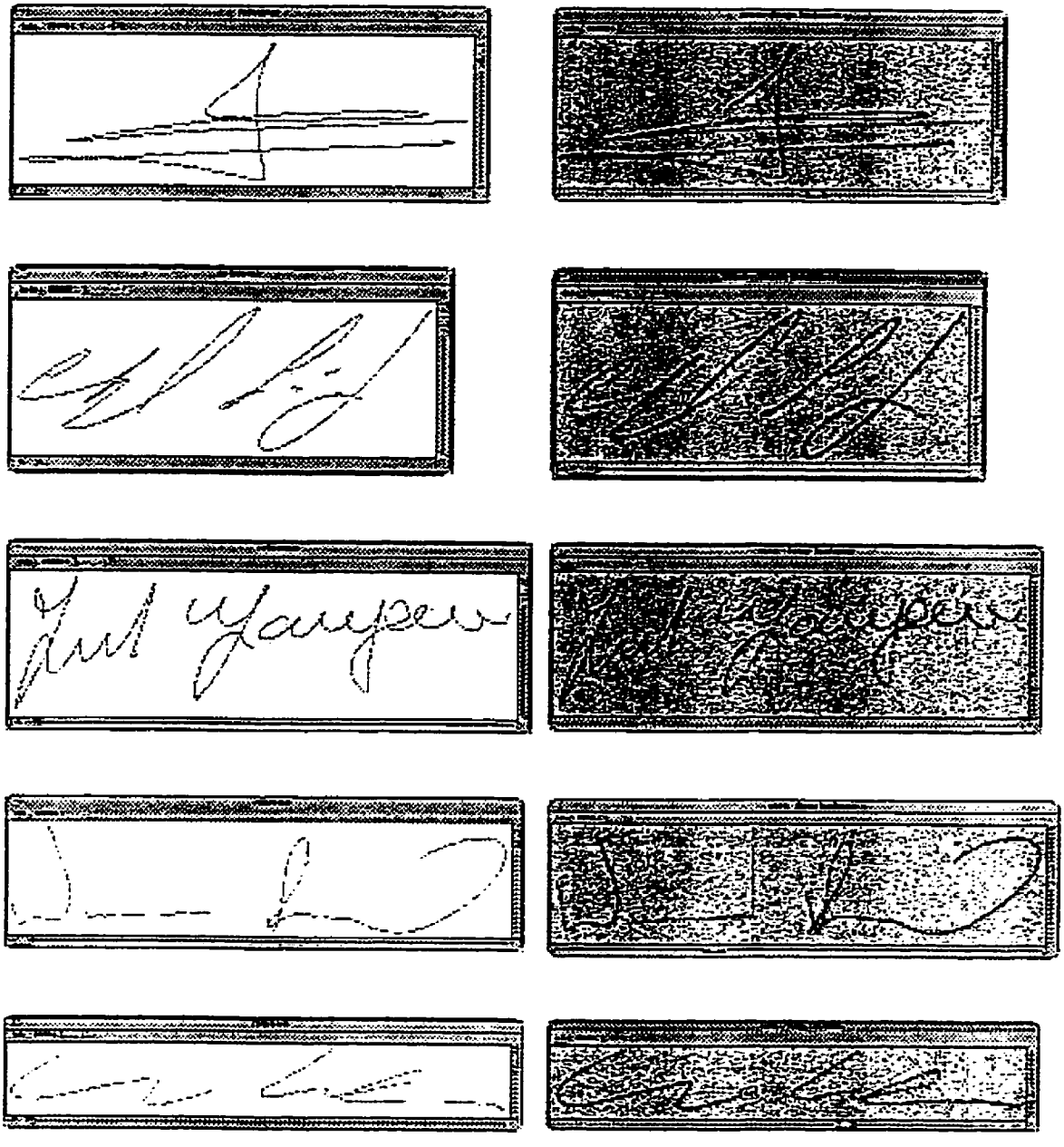


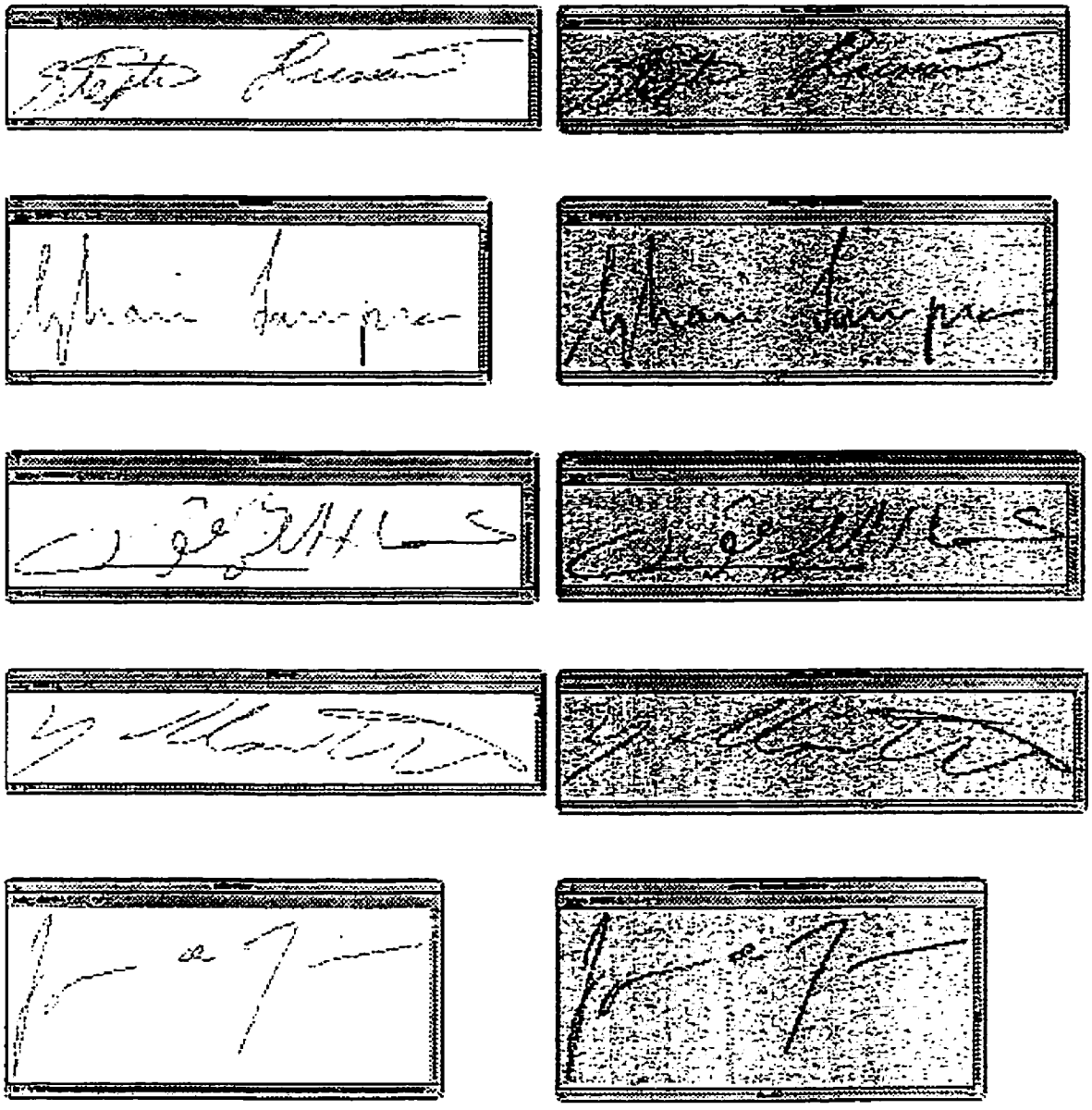


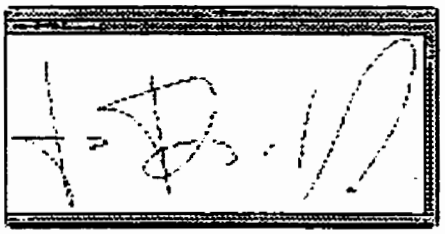


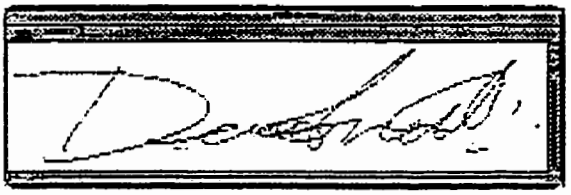


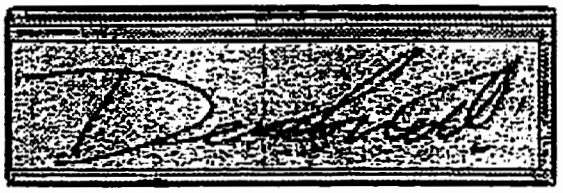


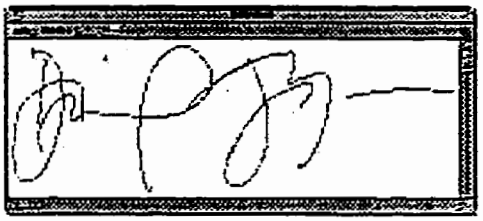


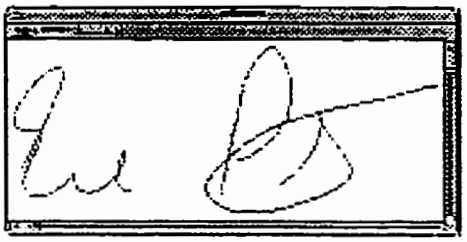


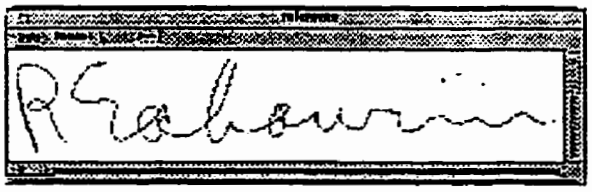












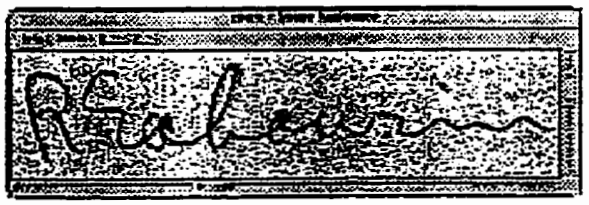
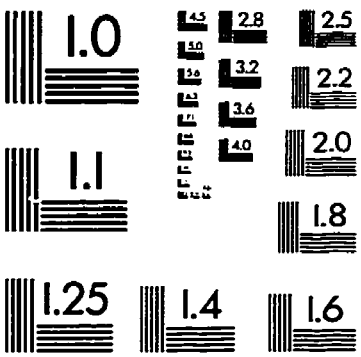
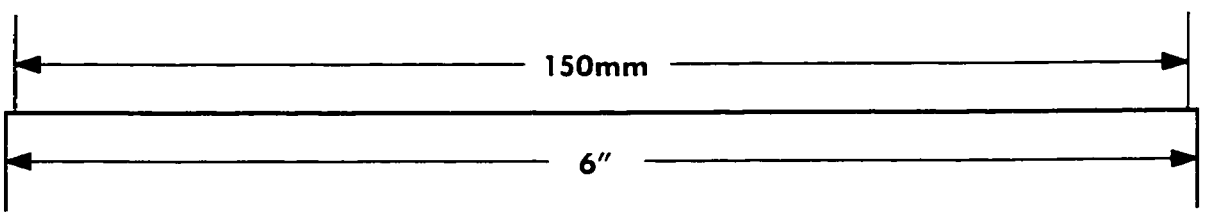


IMAGE EVALUATION TEST TARGET (QA-3)





APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved