

**ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC**

**MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE**

**COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE
M. ING.**

**PAR
JEAN-PHILIPPE VALOIS**

RECONNAISSANCE EN LIGNE DE CROQUIS DE SCHÉMAS ÉLECTRIQUES

MONTRÉAL, LE 14 JUIN 2001

© droits réservés de Jean-Philippe Valois

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

- **Mme Myriam Côté, directrice de mémoire**
Département de génie de la production automatisée à
l'École de technologie supérieure
- **M. Mohamed Cheriet, codirecteur**
Département de génie de la production automatisée à
l'École de technologie supérieure
- **M. Alain Croteau, président du jury**
Département de génie de la production automatisée à
l'École de technologie supérieure
- **M. Chakib Tadj, professeur**
Département de génie électrique à
l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 7 JUIN 2001

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

RECONNAISSANCE EN LIGNE DE CROQUIS DE DESSINS D'INGÉNIERIE

Jean-Philippe Valois

Sommaire

Cette recherche s'inscrit dans le cadre des nouvelles approches pour le développement de systèmes d'édition graphique. Notre objectif est de permettre à un utilisateur de réaliser des dessins d'ingénierie précis à l'aide d'un outil interactif qui permette la saisie d'informations sous la forme de croquis faits à la main. Ceci implique une approche qui s'appuie majoritairement sur les techniques en reconnaissance de graphiques, mais qui tient aussi compte des recherches récentes dans le domaine des interfaces basées sur le crayon numérique.

Nous décrivons dans ce travail la nature du défi posé par la reconnaissance en ligne de croquis de dessins d'ingénierie et, à la lumière des ouvrages réalisés dans des domaines de recherche connexes, nous montrerons comment une technique de reconnaissance hiérarchique basée sur une description structurale et sur les relations topologiques des formes à reconnaître apporte une solution à notre problématique.

Dans le cadre de cette recherche, nous avons développé un système afin de mettre en œuvre notre méthode de reconnaissance, lequel fut validé auprès d'un groupe d'utilisateurs témoins. Dans ce travail, nous présentons les résultats obtenus lors de ces tests ainsi qu'une discussion sur le sujet.

Enfin, nous montrerons comment ce système est à notre avis un bon premier pas dans la direction des outils d'édition graphique intelligents et nous présenterons quelques-unes des avenues de recherche à envisager pour l'approfondissement de la technique.

Mots clés : reconnaissance de graphiques et de croquis, ordinateurs à stylo, crayon numérique, formes à tracés multiples, édition interactive, informations contextuelles, reconnaissance en ligne, embellissement, reconnaissance de symboles et de gestes, reconnaissance de schémas et de diagrammes, reconnaissance incrémentale.

ON-LINE RECOGNITION OF SKETCHED ENGINEERING DRAWINGS

Jean-Philippe Valois

Abstract

This research is based on emerging approaches to graphics editing applications. Our objective in using these new approaches was to develop a flexible and interactive tool that would allow a user to translate hand-drawn sketches into precise engineering drawings. This project requires a methodology founded mostly on graphics recognition techniques, but which must also draw from recent developments in the field of pen-based computing.

In this work, we demonstrate how the on-line recognition of hand-drawn engineering sketches can be challenging. In the light of various research conducted in related lines of work, we show how a method of hierarchical recognition of the structural description and topological relations of shapes can be a good solution to our problematics.

During this research, we implement a simple prototype in order to demonstrate and put into practice our recognition method. With the help of a group of subjects, we validate our method.

Finally, we show that this system is a good first step towards the development of intelligent graphics editing tools. We end our discussion with proposals on ways to improve and extend the technique.

Keywords: graphics and sketch recognition, pen-based computing, multi-stroke recognition, on-line recognition, interactive editing, contextual information, beautification, gesture and symbol recognition, diagram recognition, incremental recognition.

AVANT-PROPOS ET REMERCIEMENTS

Ce mémoire a été réalisé dans le cadre du projet de télé-enseignement Synchronédia, présidé par M. Mohamed Cheriet, professeur à l'ÉTS, et auquel participe également Mme Côté. Il a en outre été rendu possible grâce à des bourses du FODAR (Fonds de Développement Académique du Réseau) et du M.E.Q. (Ministère de l'Éducation du Québec).

Je remercie sincèrement ma directrice de mémoire, Mme Myriam Côté et mon co-directeur, M. Mohamed Cheriet, sans l'initiative desquels ce projet n'aurait pas été possible. Je tiens à leur exprimer toute ma reconnaissance pour leur dévouement, la confiance qu'ils m'ont accordée, leur rigueur et la qualité des commentaires et suggestions dont ils m'ont fait part.

Je remercie Jason I. Hong du GUIR (*Group for User Interface Research*) de l'université de Berkeley pour son assistance avec la librairie graphique SATIN (pour crayons numériques) développée par le GUIR.

Je remercie chaleureusement tous les membres du LIVIA (Laboratoire en imagerie, vision et intelligence artificielle) d'avoir aimablement accepté de prendre part à la séance de tests de validation du système, mais davantage encore pour leur enthousiasme et leur bonne humeur quotidienne. Ce fut un plaisir de les côtoyer.

De nombreux mercis à Alessandro Koerich et Christine Nadal pour leurs compétences en matière de systèmes informatiques.

Je voudrais exprimer ma reconnaissance à Katherine Ackerman et mon frère, Alain Valois, pour avoir corrigé et revu mes textes en langue anglaise.

Merci également à Nicole Mongeau, de l'ÉTS, pour ses encouragements soutenus.

Enfin, je tiens à remercier M. Nicholas Krouglicof, professeur au *Union College* (anciennement professeur à l'ÉTS) et M. Jacques de Guise, professeur à l'ÉTS, de m'avoir guidé dans mes premiers pas en tant qu'étudiant à la maîtrise.

TABLE DES MATIÈRES

	Page
SOMMAIRE	i
ABSTRACT	ii
AVANT-PROPOS ET REMERCIEMENTS.....	iii
TABLE DES MATIÈRES.....	v
LISTE DES TABLEAUX.....	viii
LISTE DES FIGURES	ix
LISTE COMPLÈTE DES ABRÉVIATIONS ET SIGLES	xii
INTRODUCTION.....	1
SITUATION DU PROBLÈME.....	3
OBJECTIFS DU MÉMOIRE.....	7
ORGANISATION DES CHAPITRES.....	7
CHAPITRE 1 : REVUE DE LA LITTÉRATURE.....	10
1.1 INTRODUCTION.....	10
1.2 MÉTHODES EXISTANTES	11
1.2.1 Gestion de l'aspect interactif du système et le procédé intuitif de l'utilisateur .	12
1.2.2 Traitement d'un signal inexact et imprécis.....	14
1.2.3 Séparation des symboles et des lignes de connexion	16
1.2.4 Représentation des connaissances	17
1.2.5 Techniques de reconnaissance.....	20
1.2.5.1 Techniques de reconnaissance structurales.....	23
1.2.5.2 Techniques de reconnaissance statistiques.....	25
1.2.5.3 Choix d'une technique.....	25
1.2.6 Techniques périphériques (rétroaction, incertitude).....	26
1.3 JUSTIFICATION DE NOTRE APPROCHE	29
1.4 BIBLIOGRAPHIE THÉMATIQUE DE L'ÉTAT DE L'ART.....	31
CHAPITRE 2 : MÉTHODE ET ARCHITECTURE PROPOSÉES	33
2.1 INTRODUCTION.....	33
2.2 SYSTÈME DANS SON ENSEMBLE	34

2.3	MODULE DE RECONNAISSANCE	35
2.3.1	Structure des composantes	38
2.3.2	Principe de « notification »	39
2.3.3	Moteur de reconnaissance	40
2.3.4	Calcul du degré de similitude	42
2.3.5	Tâches de traitement	44
2.4	CONCLUSION	45
CHAPITRE 3 : RELATIONS TOPOLOGIQUES		46
3.1	INTRODUCTION.....	46
3.2	RELATIONS TOPOLOGIQUES	46
3.2.1	Connexion.....	46
3.2.2	Distance de connexion.....	47
3.2.3	Degré de contact.....	48
3.2.4	Degré de similitude	49
3.2.5	Degré de concentricité	50
3.2.6	Degré de parallélisme	51
3.2.7	Degré de colinéarité.....	51
3.2.8	Degré de codirection.....	52
3.2.9	Degré de perpendicularité.....	52
3.2.10	Degré de superposition.....	53
3.2.11	Facteur gauche / droite	54
3.2.12	Facteur avant / arrière.....	55
3.2.13	Facteur intérieur / extérieur	56
3.2.14	Ordre temporel.....	56
3.2.15	Distance.....	57
3.3	CONCLUSION	57
CHAPITRE 4 : EXTRACTION DES PRIMITIVES		58
4.1	INTRODUCTION.....	58
4.2	CHOIX D'UNE MÉTHODE DE SEGMENTATION POUR L'EXTRACTION DES PRIMITIVES	59
4.3	DESCRIPTION DE LA MÉTHODE DE SEGMENTATION.....	60
4.3.1	Identification des maximums et minimums locaux.....	60
4.3.2	Identification des points de transition	62
4.4	EXTRACTION DES PRIMITIVES	65
4.4.1	Extraction préliminaire des lignes et arcs	65
4.4.2	Définition des lignes et des arcs.....	69
4.4.2.1	Définition de la primitive <i>ligne</i>	70
4.4.2.2	Définition de la primitive <i>arc</i>	71
4.4.3	Validation et recherche des autres primitives	72
4.4.4	Qualité de la segmentation et raffinement des paramètres	73
4.5	CONCLUSION	75
CHAPITRE 5 : PRÉ-TRAITEMENT		76
5.1	INTRODUCTION.....	76
5.2	SECTIONNEMENT DES EXTRÉMITÉS.....	76

5.3	ENLÈVEMENT DES POINTS COÏNCIDENTS	76
5.4	ÉGALISATION	77
5.5	ADOUCCISSEMENT	78
5.6	INTERPOLATION SPLINE	78
CHAPITRE 6 : EMBELLISSEMENT		80
6.1	INTRODUCTION.....	80
6.2	PROBLÉMATIQUE.....	81
6.3	L'EMBELLISSEMENT	81
6.4	RÉSULTATS D'EMBELLISSEMENT.....	82
CHAPITRE 7 : RÉSULTATS EXPÉRIMENTAUX ET DISCUSSION.....		83
7.1	INTRODUCTION.....	83
7.2	PROTOCOLE DE VALIDATION ET RÉSULTATS DE RECONNAISSANCE.....	83
7.2.1	Paramètres du système	84
7.2.2	Reconnaissance de la bobine	85
7.2.3	Reconnaissance du condensateur	87
7.2.4	Reconnaissance de la diode	89
7.2.5	Reconnaissance de la source de tension.....	91
7.3	DISCUSSION.....	92
7.4	CONCLUSION SUR LA RECONNAISSANCE DES SYMBOLES.....	95
CONCLUSION		97
	PRINCIPALES CONTRIBUTIONS	97
	PERSPECTIVES.....	98
ANNEXES		
	1 : Heuristiques de reconnaissance.....	101
	2 : Description structurelle des composantes.....	116
	3 : Résultats d'embellissement	122
	4 : Références électroniques	124
	5 : Références thématiques.....	126
BIBLIOGRAPHIE.....		128

LISTE DES TABLEAUX

	Page
Tableau I	Récapitulation des états de l'art pertinents par domaine d'application... 31
Tableau II	Récapitulation des articles pertinents en reconnaissance de graphiques par domaine d'application 32
Tableau III	États 66
Tableau IV	Événements entrants 66
Tableau V	Événements sortants 67
Tableau VI	Prédicats..... 67
Tableau VII	Variables d'état 67
Tableau VIII	Actions spécifiques 68
Tableau IX	Table état / événement 69
Tableau X	Détails des actions à prendre en fonction de l'état / événement..... 69
Tableau XI	Taux de reconnaissance du symbole <i>bobine</i> selon l'utilisateur..... 85
Tableau XII	Taux de reconnaissance du symbole <i>condensateur</i> selon l'utilisateur..... 87
Tableau XIII	Taux de reconnaissance du symbole <i>diode</i> selon l'utilisateur..... 89
Tableau XIV	Taux de reconnaissance du symbole <i>source de tension</i> selon l'utilisateur 91

LISTE DES FIGURES

		Page
Figure 1	Dessin d'ingénierie imprimé ou numérisé	5
Figure 2	Croquis d'un schéma électrique saisi en ligne	5
Figure 3	Texte manuscrit numérisé	5
Figure 4	Symbole saisi en ligne	5
Figure 5	Tracé embelli d'une bobine dessinée à l'oblique	6
Figure 6	Résumé du procédé complet	9
Figure 7	Système dans son ensemble	34
Figure 8	Primitive « ligne »	36
Figure 9	Composante spécifique « bout en T »	36
Figure 10	Symbole « source de tension »	36
Figure 11	Module de reconnaissance hiérarchique	37
Figure 12	Niveaux de reconnaissance	38
Figure 13	Structure du symbole <i>condensateur variable</i>	39
Figure 14	Structure du symbole <i>source de tension</i>	39
Figure 15	Composante latente	40
Figure 16	Reconnaissance de la composante spécifique flèche	41
Figure 17	Boîtes englobantes des composantes <i>signe plus</i> et <i>cercle</i>	42
Figure 18	Transformées typiques pour le calcul des facteurs de confiance	43
Figure 19	Transformée de confiance pour un moyenne de 0,15 et un écart type de 0,25	44
Figure 20	Voisinage « significatif » pour deux composantes différentes	45
Figure 21	Tracé continu	47
Figure 22	Primitives connectées, relation de connexion vraie	47
Figure 23	Tracés distincts	47
Figure 24	Relation de connexion fausse	47
Figure 25	Calcul de la relation distance de connexion	48
Figure 26	Relation distance de connexion élevée	48
Figure 27	Relation distance de connexion faible	48
Figure 28	Relation de contact entre deux composantes	49
Figure 29	Dimension des boîtes englobantes	49
Figure 30	Intersection des boîtes englobantes de C1 et C2	49
Figure 31	Degré de similitude = 1,0	50
Figure 32	Degré de similitude = 0,5	50
Figure 33	Calcul du degré de concentricité	50
Figure 34	Degré de concentricité en a) 0,0, en b) 0,5 et en c) 1,0	50
Figure 35	Mesure du parallélisme	51
Figure 36	Valeurs de parallélisme. a) 1,0, b) 0,71 et c) 0,0	51
Figure 37	Facteurs pour le calcul du degré de colinéarité	52
Figure 38	Vecteurs de direction pour le calcul du degré de codirection	52
Figure 39	Calcul du degré de superposition	53
Figure 40	Valeurs de superposition. a) 0,0, b) 0,5 et c) 1,0	53
Figure 41	Calcul de la relation gauche / droite	54

Figure 42	Valeurs possibles de la relation gauche/droite. a) GAUCHE, b) AUCUN et c) DROITE	54
Figure 43	Calcul de la relation avant / arrière	55
Figure 44	Valeurs de la relation avant / arrière. a) ARRIÈRE, b) AVANT	55
Figure 45	Valeurs de la relation intérieur / extérieur. a) INTÉRIEUR, b) EXTÉRIEUR	56
Figure 46	Relation ordre temporel.....	56
Figure 47	Relation temporelle : C1 est apparue <i>avant</i> C2	56
Figure 48	Calcul de la relation « distance »	57
Figure 49	Valeurs de la relation <i>distance</i> . a) 0,0, b) 0,5, c) 1,0.....	57
Figure 50	Procédé de segmentation du tracé.....	58
Figure 51	Code de Freeman pour connectivité d'images raster	59
Figure 52	Caractéristiques du tracé dynamique	59
Figure 53	<i>k</i> -vecteurs pour le calcul de la courbure	60
Figure 54	Identification d'un <i>maximum local</i>	62
Figure 55	Identification d'un <i>minimum local</i>	62
Figure 56	Tracé en « coin »	63
Figure 57	Profil de courbure du tracé en « coin ».....	63
Figure 58	Tracé droit avec minimums locaux	63
Figure 59	Profil de courbure du tracé droit	64
Figure 60	Tracé en forme de cycloïde	64
Figure 61	Profil de courbure du tracé en forme de cycloïde	64
Figure 62	Diagramme état / transition pour la recherche des primitives	68
Figure 63	Points de début et de fin des segments.....	70
Figure 64	Exemple de facteurs de certitude : a) erreur grande, donc certitude faible b) erreur faible, donc certitude élevée	70
Figure 65	Technique pour déterminer les caractéristiques d'un arc.....	71
Figure 66	Segmentation avec facteur de voisinage <i>m</i> variable.....	74
Figure 67	pré-traitement : tronquer les extrémités	76
Figure 68	Pré-traitement : enlèvement des points coïncidents	77
Figure 69	Pré-traitement : opération d'égalisation	78
Figure 70	Pré-traitement : adoucissement.....	78
Figure 71	B-spline cubique et ses points de contrôle	79
Figure 72	Pré-traitement : interpolation spline	79
Figure 73	Procédé de reconnaissance (sans embellissement).....	80
Figure 74	Procédé d'embellissement	81
Figure 75	Structure du symbole source de tension	82
Figure 76	Embellissement d'une flèche et d'un « bout en T ».....	82
Figure 77	Embellissement d'un cercle.....	82
Figure 78	Reconnaissance du symbole bobine	83
Figure 79	Symboles de référence. a) bobine, b) condensateur, c) diode, d) source de tension	84
Figure 80	Symbole bobine erroné	86
Figure 81	Symbole bobine reconnu.....	86
Figure 82	Symbole bobine erroné	86
Figure 83	Symbole bobine erroné	86
Figure 84	Symboles condensateur erronés.....	88
Figure 85	Composante flèche non reconnue.....	90

Figure 86	Composante flèche mal reconnue.....	90
Figure 87	Symbole diode non-reconnu	90
Figure 88	Symbole <i>source de tension</i> non-reconnu	92
Figure 89	Symbole diode erroné	93
Figure 90	Symbole condensateur erroné	93
Figure 91	Primitive cercle difficile à tracer	94
Figure 92	Symbole diode erroné en raison d'un mauvais calcul de relations.....	95
Figure 93	Ligne.....	117
Figure 94	Arc	117
Figure 95	Cercle	117
Figure 96	Bobine.....	117
Figure 97	Résistance	118
Figure 98	Bout en T	118
Figure 99	Bout en C.....	118
Figure 100	Bout en S.....	118
Figure 101	Bout en flèche.....	118
Figure 102	Signe plus	119
Figure 103	Résistance	119
Figure 104	Résistance variable	119
Figure 105	Inductance	119
Figure 106	Inductance variable	119
Figure 107	Condensateur	120
Figure 108	Condensateur variable	120
Figure 109	Condensateur 2	120
Figure 110	Condensateur 2 variable	120
Figure 111	Condensateur	120
Figure 112	Condensateur variable	120
Figure 113	Condensateur 2	121
Figure 114	Condensateur 2 variable	121
Figure 115	Source de tension	121
Figure 116	Source de courant.....	121
Figure 117	Source alternative	121
Figure 118	Transistor.....	121

LISTE COMPLÈTE DES ABRÉVIATIONS ET SIGLES

image « raster »	image composée de pixels, par opposition à une image de type « vecteur »
image « <i>bitmap</i> »	image de type <i>raster</i> non-compressée où chaque élément d'information est composé d'un nombre donné de bits (ex. 24 bits pour une quantité quasi infinie de couleurs)
DAO	Dessin assisté par ordinateur
CAO	Conception assistée par ordinateur
FAO	Fabrication assistée par ordinateur
reconnaissance ascendante	technique de reconnaissance par laquelle les données sont analysées du bas vers le haut
reconnaissance descendante	technique de reconnaissance par laquelle les données sont analysées du haut vers le bas
<i>split and merge</i>	technique de segmentation par laquelle le tracé est découpé en plusieurs parties selon un ensemble de critères et où celles-ci sont fusionnées selon un tout autre ensemble de critères
OCR	Optical Character Recognition (reconnaissance optique de caractères)

INTRODUCTION

Le croquis est une forme d'expression naturelle par laquelle l'humain peut facilement et rapidement communiquer des idées par le biais du texte, de schémas et d'images ; selon Harley et Woodward (1987), l'utilisation de diagrammes planaires et de croquis précède l'invention de l'écriture de plus de 30 siècles. Très tôt, l'ordinateur a rendu possible la saisie de ces données sous diverses formes. Les premiers systèmes, pour la plupart spécialisés dans le domaine de la CAO, permirent à l'utilisateur de créer des dessins, d'abord par l'entremise d'interfaces textuelles et plus tard, par le moyen d'interfaces graphiques évoluées.

Les systèmes modernes facilitent le processus créatif de l'utilisateur en permettant une interaction avec l'ordinateur à l'aide de la souris et de crayons tactiles. Les formes dessinées et les opérations réalisées sur celles-ci sont choisies à partir du clavier ou de divers menus et palettes offrant une gamme de choix prédéterminés. Aujourd'hui, nous retrouvons trois types principaux de systèmes d'édition graphique : 1) les systèmes permettant de *peindre des pixels* individuels, 2) les systèmes servant à la création de *graphiques structurés*, tels les logiciels de CAO et 3) les systèmes à base de *croquis*, qui permettent de dessiner librement des tracés curvilignes représentés sous la forme d'encre électronique.

Nombreux sont les systèmes de nouvelle génération qui préconisent une approche intelligente afin d'améliorer la convivialité. Celles-ci prennent entre autres la forme de commandes d'édition variées et de modes d'interaction spéciaux qui servent à assister l'utilisateur. Par exemple, certains systèmes en CAO permettent la création de croquis à l'aide de techniques de contraintes qui détectent et suggèrent à l'utilisateur un choix de relations topologiques, notamment la perpendicularité, la tangence, etc. Les logiciels les plus avancés en cette matière permettent les commandes vocales ou encore l'utilisation de gestes pour mener des opérations simples comme le copier/coller. Dans leur ensemble, ces modalités nouvelles sont appelées « interfaces calligraphiques » par Fonseca et Jorge (2000a).

En bout de ligne cependant, malgré le fait qu'elle se dise *graphique*, l'interface usager moderne est caractérisée par des modalités d'*interaction discrète* telles que la sélection dans des menus, le fait de pointer des régions de l'écran ou l'action d'appuyer sur des boutons. Cette approche est une entrave au processus créatif de l'utilisateur et finalement, le paradigme du dessin, si puissant et pratique pour l'être humain, est en vaste partie sous-exploité (Fonseca et Jorge, 2000a). Par surcroît, comme le notent Plamondon et Srihari (2000), malgré toutes ces innovations technologiques, un crayon et un bout de papier sont dans maintes situations encore à ce jour préférables à un clavier et une souris ; dans une classe d'école, l'étudiant note les textes, équations et diagrammes avec un crayon. Ce constat est à la base d'une approche nouvelle en édition graphique où les systèmes classiques font place à ce qui est communément appelé le « *pen computing* », ou « l'informatique à base de crayon ».

En raison de la popularité de ces nouveaux systèmes, notamment les assistants personnels numériques (ou *PDA, Personal Digital Assistants*), les techniques de reconnaissance de gestes et de formes géométriques faites à la main sont dans la mire des chercheurs. Certaines avenues de recherche ont pour but ultime de mimer et prolonger la métaphore du papier et du crayon par un traitement automatique de l'encre électronique (Plamondon et Srihari, 2000). D'autres encore recherchent une interaction plus naturelle qui intègre aux systèmes d'édition la reconnaissance de gestes et de formes simples, comme c'est le cas dans les travaux de Julia et Faure (1995) et Jorge et Fonseca (1999). Quant à eux, Saund et Moran (1994) croient en l'importance d'interagir selon des critères perceptifs qui s'appuient sur la théorie de la Gestalt, où les dessins prennent forme au fur et à mesure que de nouveaux tracés sont ajoutés au croquis.

Par contre, ces systèmes sont essentiellement limités à deux domaines : 1) la reconnaissance de dessins et diagrammes au niveau du design, où les formes et schémas sont sujets à des critères perceptifs et subjectifs, et 2) la reconnaissance de tracés et formes géométriques simples qui sont détachées les unes des autres. En somme, s'il existe en édition graphique une vaste quantité de travaux dédiés au traitement de graphiques et de diagrammes hors-ligne, nous n'avons encore vu aucun

système qui puisse assister l'utilisateur de manière *interactive* à la création de croquis et schémas *spécialisés*.

À la lumière des nouvelles avancées dans le domaine des interfaces intelligentes basées sur le crayon et en nous appuyant sur les multiples ouvrages consacrés depuis ces dernières années à la reconnaissance de graphiques, nous présenterons dans ce travail une méthode permettant de reconnaître en ligne des croquis de dessins d'ingénierie. Le système développé permettra en outre à l'utilisateur de réaliser des schémas rapidement et naturellement, sans que l'utilisation d'une interface complexe soit nécessaire.

Situation du problème

L'interprétation interactive et l'embellissement de croquis de schémas électriques faits librement à la main est un domaine qui soulève plusieurs défis techniques au chapitre de la reconnaissance de formes, notamment :

- le traitement en ligne d'un signal, inexact et imprécis,
- la reconnaissance de schémas bidimensionnels et interconnectés,
- la gestion de l'aspect interactif du système et du procédé intuitif de l'utilisateur,
- l'embellissement de schémas,
- la gestion de la rétroaction et de l'incertitude.

Selon le type des données lues en entrée, on peut identifier deux types de reconnaissances : en ligne et hors-ligne. De ce fait, le choix des primitives sera généralement fonction de la tâche à accomplir, mais il sera également influencé par la qualité des données à reconnaître : par exemple, la reconnaissance de schémas manuscrits requiert un choix de primitives différent de celui préféré pour la reconnaissance de l'imprimé.

Le domaine de la reconnaissance *hors-ligne* implique que les données sont lues après avoir été créées ; c'est le cas de l'interprétation de dessins d'ingénierie numérisés. Par opposition, les données *en ligne* sont celles qui peuvent être saisies au moment même

où elles sont fournies par l'utilisateur ; elles proviennent généralement d'une tablette numérique ou d'une souris.

Les signaux saisis *hors-ligne* contiennent deux types d'informations : une information spatiale, dont l'unité est le pixel, composé d'une paire de coordonnées (x, y) et une intensité, souvent traduite en 256 niveaux de gris. Les données *en ligne*, quant à elles, sont accompagnées d'une dimension temporelle ; vu l'échantillonnage constant de l'unité de saisie, nous pouvons savoir à quel moment l'élément d'information en question a été généré, résultant en un triplet (x, y, t) . On peut aussi connaître la pression exercée par le crayon ; cette donnée est analogue à l'information d'intensité disponible dans le *hors ligne*.

La qualité des données à reconnaître influencera également le choix des primitives. Les schémas imprimés seront souvent faits de données dont la définition et l'échelle sont constantes (ex. au sein d'un même dessin en reconnaissance de schémas électroniques, le symbole graphique représentant la porte logique « ou » aura toujours la même définition). Ces derniers seront aussi généralement moins bruités et mieux définis, donc dans l'ensemble plus stables. Inversement, les données manuscrites seront inexactes et davantage bruitées et leur échelle, définition et orientation plus variables.

Contrairement aux systèmes qui s'attardent aux dessins d'ingénierie précis (conçus pour le DAO), nous tentons de faire la reconnaissance de croquis, voire de schémas et diagrammes qui, de par leur nature, sont approximatifs (voir figure 1 et figure 2). En effet, Jenkins et Martin (1993) notent qu'en raison de la nature inexacte des croquis, divers usagers ont diverses habiletés quand vient le temps de dessiner et que pour cette raison, le système doit pouvoir s'ajuster (ex., à l'aide de paramètres variables). Aussi, en conséquence de cette nature inexacte, les croquis ne sont pas forcément dessinés à l'échelle (Ullman, Wood et Craig, 1990). La nature imprécise des croquis est cependant aussi un avantage car dans la mesure où la reproduction exacte n'est pas une exigence, nous pouvons nous permettre d'apporter au croquis embelli des modifications qui, selon toute probabilité, n'incommoderont pas l'utilisateur.

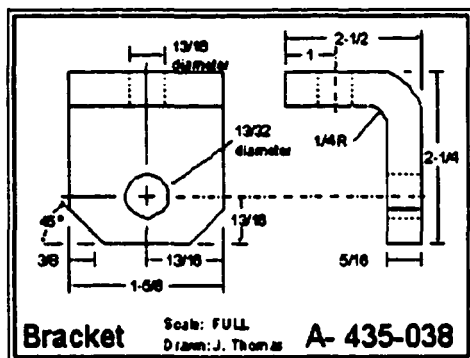


Figure 1 - Dessin d'ingénierie imprimé ou numérisé

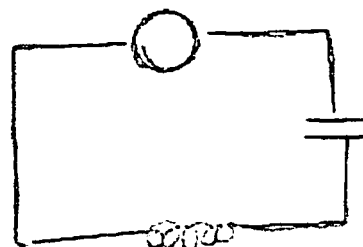


Figure 2 - Croquis d'un schéma électrique saisi en ligne

Par ailleurs, comme le notent Tappert, Suen et Wakahara (1990), la reconnaissance d'équations, de schémas et de dessins de ligne est complexifiée par la nécessité d'une analyse syntaxique bidimensionnelle. En effet, contrairement à la reconnaissance classique de l'écriture manuscrite dont la syntaxe s'exprime avant tout sur une dimension, nous devons distinguer des caractéristiques et des relations qui s'étalent sur deux dimensions (voir figure 3 et figure 4). Blostein (1996) poursuit dans la même veine en avançant que la compréhension d'un diagramme implique deux activités : d'une part la reconnaissance et d'autre part, l'analyse de la disposition des symboles, tâche qui relève de la séparation de ces derniers d'avec les lignes de connexion. Nous verrons dans le chapitre suivant comment divers auteurs abordent cette difficulté.

Yaourt

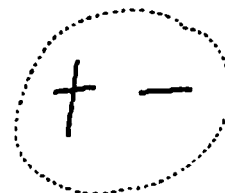


Figure 3 - Texte manuscrit numérisé

Figure 4 - Symbole saisi en ligne

La plupart des systèmes existants imposent des contraintes à l'utilisateur : certains exigent que les symboles et lignes de connexion soient dessinés en deux étapes différentes (Lin, 1982) ou accompagnés de marques ayant pour but de faciliter la phase de

reconnaissance. D'autres forcent l'utilisateur à dessiner les formes d'une manière bien précise, ou encore avec un nombre de traits donné. Pourtant, selon Ullman et al. (1990), il est impératif de permettre à l'utilisateur de dessiner de façon telle que son processus créatif ne soit pas gêné. Nous chercherons donc à libérer l'utilisateur de toute contrainte lors de la création de dessins (de sorte qu'idéalement, les formes puissent être reconnues indépendamment de l'échelle, de la rotation¹ et de la translation) et parallèlement, tenterons de tirer profit de la nature interactive du système afin de lui permettre d'intervenir en cas d'erreur.

Le but ultime de notre système est d'abord de reconnaître les symboles et ensuite de les embellir. L'embellissement consiste donc à transformer les tracés originaux, une fois les symboles identifiés, en une représentation plus exacte de ceux-ci. Cette tâche est problématique car dans la mesure où l'utilisateur est libre de dessiner sans contraintes, nous devons pouvoir embellir les symboles indépendamment de l'angle et de l'échelle (voir figure 5).

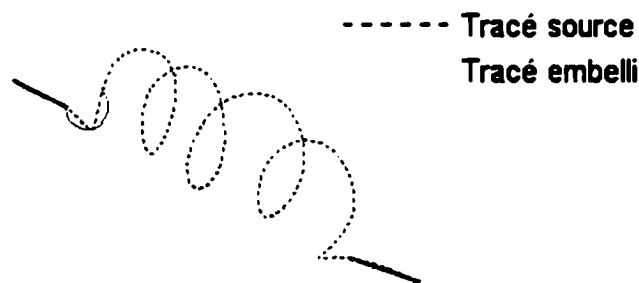


Figure 5 - Tracé embelli d'une bobine dessinée à l'oblique

Enfin, nous nous intéressons aux techniques périphériques ; certains auteurs les nomment « méta-techniques » car elles ont pour objectif de prendre un regard global sur la méthode de reconnaissance. La difficulté consiste à prendre en considération les informations contextuelles afin de réduire les erreurs et l'incertitude dans la segmentation et la reconnaissance de symboles, mais aussi de considérer les

¹ Notons que l'orientation des composantes sera conservée au moment de l'embellissement de telle sorte que la sémantique du schéma ne soit pas modifiée.

méthodes de correction par rétroaction (Blostein, 1996, Tombre, 1996 et Chhabra, 1997).

Nous avons défini dans cette section la nature du problème posé par la reconnaissance en ligne de croquis de dessins d'ingénierie. Dans la section suivante, nous présentons les objectifs de notre recherche.

Objectifs du mémoire

Le but de ce mémoire est de développer une méthode de reconnaissance en ligne de croquis de dessins d'ingénierie. Dans la mesure où la reconnaissance de diagrammes est un domaine complexe, tel que nous l'avons vu à la section précédente, nous nous limiterons dans cet ouvrage à la reconnaissance des symboles déjà isolés.

Ce système est développé dans le cadre du projet Synchronédia² pour le télé-enseignement et la télé-manipulation. Il servira à assister un professeur lors de présentations sur un « tableau blanc » (*whiteboard*) dans le but de permettre la saisie de schémas complets. Ces derniers seront éventuellement transformés automatiquement en notes de cours électroniques qui pourront être réutilisées et manipulées sous diverses formes par les étudiants. C'est dans le but ultime de solutionner cette problématique que nous entreprenons les travaux de recherche décrits dans cet ouvrage.

Organisation des chapitres

Nous donnons dans cette section un aperçu du contenu de ce mémoire. Dans le premier chapitre, nous présenterons un état de l'art dans lequel nous passerons en revue les techniques servant à la reconnaissance de graphiques et de croquis. Nous verrons que nos travaux sont à la croisée de plusieurs domaines de recherche,

² Le projet Synchronédia est mené par l'École des technologies de l'information (ÉTI), consortium de 4 universités québécoises : l'École de Technologie Supérieure (ÉTS), l'Institut National de Recherche Scientifique (INRS), l'Université du Québec à Montréal (UQAM) et la Télé-Université du Québec (TÉLUQ)) sous la direction du professeur Mohamed Cheriet.

lesquels seront analysés en fonction de notre problématique. Nous situerons enfin notre méthode par rapport aux techniques existantes et justifierons le choix de notre approche.

La section suivante comportera le cœur de notre technique ; nous y exposerons en détails la méthode et l'architecture proposées. Nous verrons comment nous reconnaissons les symboles électriques à l'aide d'un moteur hiérarchique et de quelle manière nous utilisons les relations topologiques entre les composantes graphiques pour définir la structure des composantes.

Dans les quatre chapitres suivants, nous définirons les éléments de connaissance et les traitements intermédiaires du système de reconnaissance. La première de ces sections sera consacrée à une définition complète des relations topologiques sur lesquelles s'appuie le moteur pour arriver à un résultat. La section suivante expliquera la procédure de segmentation et d'extraction de primitives. Ensuite, nous décrirons la technique de pré-traitement des signaux et terminerons avec un bref exposé sur les défis posés par l'embellissement des symboles.

Finalement, nous présenterons une analyse des résultats expérimentaux obtenus suite à des tests réalisés auprès d'un groupe d'utilisateurs témoins et nous terminerons en résumant les principales perspectives de recherche en rapport à ce travail. Le schéma de la figure 6 illustre notre procédé de reconnaissance ainsi que les chapitres qui sont reliés à étape du traitement.

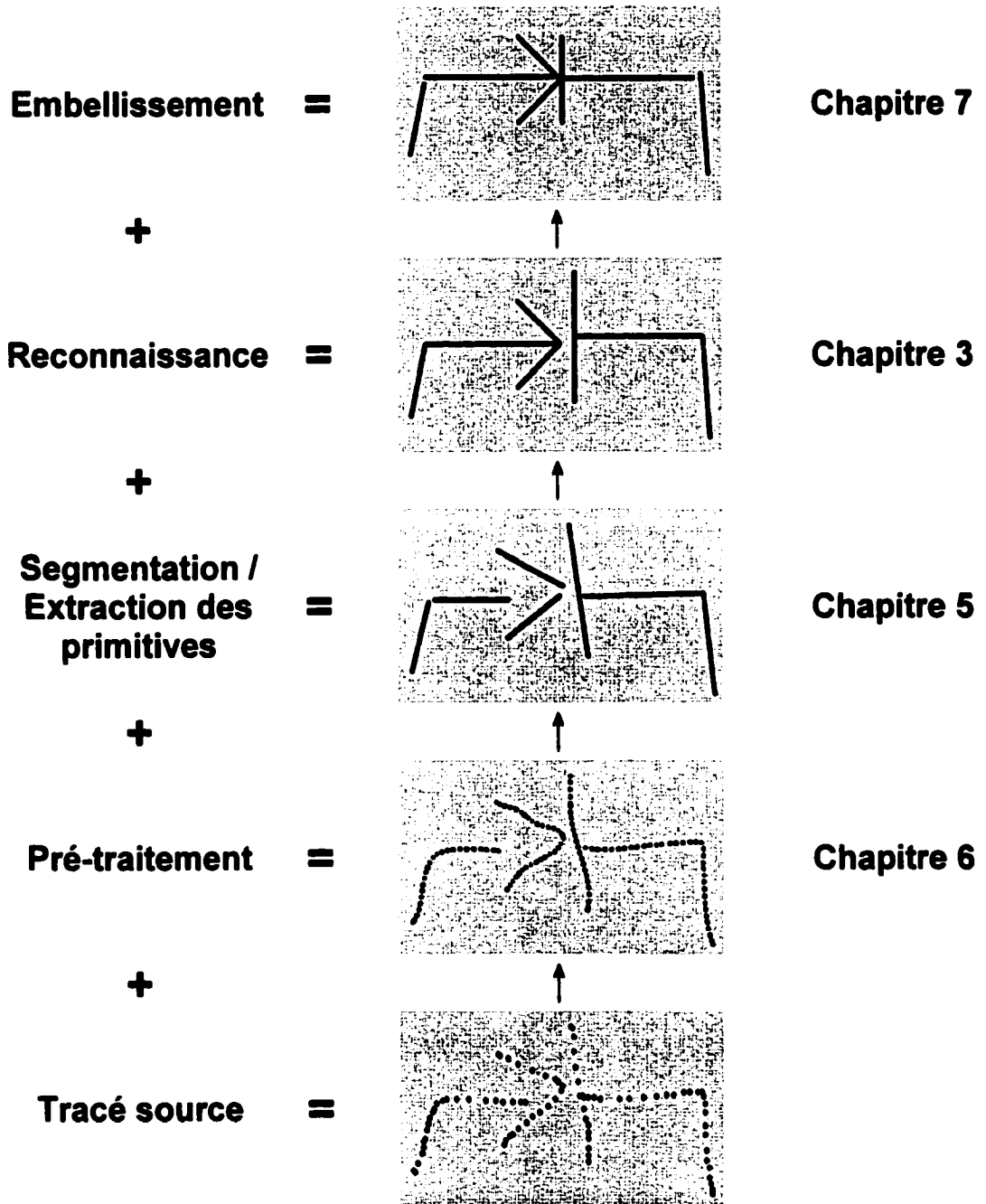


Figure 6 - Résumé du procédé complet

CHAPITRE 1

REVUE DE LA LITTÉRATURE

1.1 Introduction

La reconnaissance en ligne de diagrammes et de schémas techniques faits librement à la main est un domaine qui se situe à la croisée de plusieurs spécialités en reconnaissance de formes, notamment les suivantes :

- reconnaissance de symboles et de diagrammes faits à la main,
- interprétation et conversion de dessins d'ingénierie,
- reconnaissance en ligne de croquis,
- techniques d'embellissement par contraintes.

Dans la première catégorie, il est question de la reconnaissance de formes et symboles évolués, faits à la main, dont la disposition n'est pas uniquement linéaire. Les travaux dans ce domaine sont pour la plupart dérivés et réalisés en marge de la catégorie qui porte sur l'interprétation et la conversion des dessins d'ingénierie. Nous portons une attention particulière à cette dernière car depuis de nombreuses années, cet axe de recherche a suscité beaucoup d'intérêt dans le domaine de la reconnaissance de graphiques. Comme bon nombre des méthodes proposées avaient pour but la reconnaissance de schémas électroniques, ces travaux auront une grande influence sur les nôtres.

Le second domaine, celui de la reconnaissance de croquis, est aujourd'hui davantage étudié en raison des avancements récents en matière de technologies informatiques à base de crayon tactile (Zhao, 1993). Les systèmes existants dans ce domaine sont généralement limités à la reconnaissance de gestes et de formes simples, mais sont malgré cela d'un grand intérêt pour l'attention qu'elles portent à l'étude des signaux faits librement à la main. Enfin, nous répertorions les travaux parus en embellissement de dessins par l'application de contraintes sur les relations, principalement en raison de la

place qu'ont les croquis dans ces ouvrages, mais aussi parce que les applications développées dans ce domaine sont de nature interactive.

Plusieurs travaux ont été réalisés pour le compte de ces avenues de recherche, mais très peu se comparent en totalité à notre problématique. S'il en est une, l'avenue qui se rapproche le plus de la nôtre serait celle de la reconnaissance de graphiques ; la majeure partie des ouvrages réalisés dans ce champ d'expertise consiste à trouver une représentation électronique compacte et précise de dessins (en vaste partie d'ingénierie) imprimés ou faits à la main sur papier. Si les techniques de saisie de données et de pré-traitement qui y sont utilisées sont différentes des nôtres, les techniques et méthodologies de reconnaissance et d'interprétation de schémas et de diagrammes qui y sont développées sont transposables à notre problématique.

Dans les sections suivantes, nous passerons en revue de manière détaillée les méthodes qui abordent notre problématique. Finalement, nous verrons comment par l'analyse de ces travaux nous justifions notre approche.

1.2 Méthodes existantes

Dans cette section, nous décrivons et analysons les méthodes existantes qui présentent des éléments de solution à la problématique énoncée précédemment. Nous montrerons comment, à la lumière de cette étude, nous proposons une avenue de recherche spécifique.

Nous présenterons notre analyse des méthodes existantes de la façon suivante, selon les thèmes énoncés plus tôt dans la problématique :

- gestion de l'aspect interactif du système et le procédé intuitif de l'utilisateur,
- traitement d'un signal inexact et imprécis,
- séparation des symboles et des lignes de connexion,
- représentation des connaissances,
- choix d'une technique de reconnaissance,
- utilisation de techniques périphériques (rétroaction, incertitude).

1.2.1 Gestion de l'aspect interactif du système et le procédé intuitif de l'utilisateur

Selon Jenkins et Martin (1993), le croquis et les représentations sous forme de diagrammes sont importants au stade du design conceptuel, dans la mesure où les systèmes de DAO existants n'offrent peu ou pas d'outils qui en permettent facilement la création. Les auteurs font un état de l'art en répertoriant les travaux réalisés dans les trois secteurs de recherche suivants :

1. systèmes de DAO à base de contraintes,
2. systèmes qui permettent l'interprétation de croquis faits sur papier (et méthode d'interprétation de dessins d'ingénierie),
3. approches nouvelles de création.

Les auteurs concluent en avançant qu'aucun système existant n'est adéquat pour la création de dessins de croquis en raison de leur nature particulière, qu'ils définissent selon les critères suivants :

- les croquis doivent être faits de manière naturelle,
- l'inexactitude des croquis,
- l'outil ne doit pas interrompre le processus créatif de l'utilisateur,
- la satisfaction de contraintes (relations) doit être automatique,
- quand le système est au repos, le processus d'analyse doit être maximisé.

Nakagawa, Kato, Machii et Souya (1993) jugent qu'il est important pour la créativité et le développement du processus de pensée de permettre à l'utilisateur de travailler avec le croquis jusqu'à l'épuisement des idées ; c'est ce que les auteurs appellent la « reconnaissance paresseuse ». De plus, dans le but de permettre à l'utilisateur d'interagir comme il le souhaite avec le dessin, plusieurs représentations internes devraient être conservées : fonctionnelle, figurative et manuscrite. Par exemple, quand l'utilisateur sélectionne la gomme à effacer, il lui serait permis d'agir au niveau de la représentation manuscrite. Un concept similaire est présent dans le système de Saund et Moran (1994) qui proposent un schéma de représentation hiérarchique qui permet de multiples interprétations d'un même ensemble de données ; à ce chapitre, le système conserve en mémoire un bitmap du croquis original : chaque fois qu'un nouveau trait

est dessiné, les nouvelles intersections sont calculées, les traits existants divisés et la structure de représentation revue. Gross (1996) présente le « Cocktail Napkin », système qui permet à l'utilisateur d'ajouter et de raffiner le dessin en cours, mais qui l'oblige à définir les relations entre glyphes manuellement. Finalement, en vue de la reconnaissance de dessins d'ingénierie, le système de Yu, Samal, et Seth (1997) permet à l'utilisateur d'intervenir pour la correction d'erreurs résiduelles.

En 1997, Igarashi, Kawachiya, Matsuoka et Tanaka (1997a) répertorient plusieurs systèmes de DAO afin de trouver le système idéal. Les auteurs subdivisent le processus complet de création d'un dessin en trois étapes :

- planification : action de traduire une image mentale en une séquence de commandes disponibles dans l'outil,
- sélection : action de sélectionner les commandes qui serviront à la création (choix de menus, icônes, etc.),
- manipulation : action de manipuler les objets graphiques apparaissant sur la surface de dessin (bouger, pivoter, etc.).

Les auteurs assument que le problème principal des outils de DAO actuels est que ceux-ci exigent un effort considérable de la part de l'utilisateur au niveau du processus de planification cognitif. Quelques systèmes et techniques sont ensuite présentés, notamment les systèmes à base de contraintes, l'ajout de fonctionnalité, la programmation par exemples, les systèmes à base de gestes et les techniques d'embellissement. Les auteurs argumentent que ces approches souffrent toutes d'un même problème, soit la surcharge de travail reliée au processus de planification cognitif (processus cognitif par lequel l'utilisateur, afin d'obtenir l'image désirée, se fait un plan des combinaisons possibles avec les opérations physiques disponibles dans le logiciel de dessin).

En guise de solution à cette problématique, ils proposent un système à base de contraintes contextuelles entièrement exempt de commandes d'édition. Quand un trait est dessiné, le système déduit le diagramme précis souhaité en se basant sur des contraintes perceptives entre les traits et le contexte environnant. Divers traits

candidats embellis sont ensuite affichés; l'utilisateur n'a qu'à choisir celui qui traduit le mieux sa pensée.

Enfin, Badros (1998) fait un état de l'art de l'utilisation de contraintes dans les applications graphiques interactives. Selon lui, de nombreux systèmes graphiques interactifs, incluant des systèmes de dessin, de schémas, de visualisation et d'animation emploient un moteur de résolution à base de contraintes pour gérer la disposition et les relations entre les entités apparaissant à l'écran.

En résumé, les recherches réalisées dans le domaine de l'embellissement ont généralement pour objectif d'améliorer l'interaction homme-machine. Un avantage de la reconnaissance en ligne est évidemment la disponibilité de l'utilisateur. Non seulement ce dernier est-il en mesure de s'adapter au système, mais encore mieux, il peut l'aider en cas d'erreur. L'embellissement est soit fait sur une base subjective et contextuelle, soit selon les indications de l'utilisateur, mais dans tous les cas, les auteurs accordent de l'importance à la convivialité et l'aspect intuitif des systèmes. De plus, les systèmes dont il est question sont toujours des systèmes en ligne qui tentent d'embellir les tracés presque instantanément. C'est davantage en rapport à ces préoccupations et de la dynamique d'utilisation que nous nous intéressons à ces ouvrages.

1.2.2 Traitement d'un signal inexact et imprécis

L'étape de pré-traitement permet de transformer les suites de points (saisis de la tablette numérique) en des tracés ou agencements de tracés dont les caractéristiques propres nous permettront de *choisir des primitives* qui serviront à la représentation des formes que nous souhaitons interpréter. C'est à l'aide de ces primitives que le moteur de reconnaissance cherchera à faire l'appariement avec les données de référence présentes dans son dictionnaire.

Les tracés, données lues du crayon tactile, sont de par leur nature bruités et inexacts ; il s'agit là d'une caractéristique typique des signaux en ligne. Par surcroît, les croquis que nous cherchons à reconnaître sont, plus souvent qu'autrement, composés de traits

dessinés de manière libre et très approximative. Ainsi, l'étape de pré-traitement peut idéalement agir à deux niveaux : *nettoyage* et *embellissement*.

Le *nettoyage* consistera en l'application de principes traditionnels de pré-traitement tels ceux répertoriés par Tappert et al. (1990) : filtrage de points redondants, adoucissement, interpolation, correction des points aléatoires, enlèvement de l'effet de crochetage, etc. Si les techniques de pré-traitement définies précédemment sont bien connues, celles permettant d'*embellir* le tracé d'entrée le sont considérablement moins. Hutton, Cripps, Elliman et Higgins (1997) notent que l'embellissement doit être fait avec soin : deux lignes seront peut-être parallèles par hasard. Ainsi selon eux, la détection des relations spatiales valides devrait être confirmée par l'utilisateur.

Pavlidis et Van Wyk (1985) définissent l'embellissement comme une procédure par laquelle un dessin approximatif pourrait être automatiquement redessiné de façon plus propre. Leur système permet d'inférer les relations (sous la forme de contraintes) entre les entités du dessin et de modifier ce dernier de manière à satisfaire les contraintes. En outre, ils jugent important d'utiliser les contraintes négatives³ afin d'assurer que le résultat de l'embellissement n'aille pas à l'encontre des intentions de l'utilisateur.

Jenkins et Martin (1992), pour leur part, décrivent un système similaire, mais plus flexible que celui de Pavlidis et Van Wyk (1985), par lequel le croquis est embelli au moment même où il est dessiné. Pour le traitement, deux processus simultanés sont prévus : un qui permettra à l'utilisateur de dessiner sans interruption et l'autre qui embellira en transformant les suites de points en lignes droites, arcs et courbes de Bézier composées. Chaque nouveau trait créé se verra comparé avec l'ensemble des traits existants afin de déterminer l'existence de relations géométriques. Dans la même lignée, Igarashi, Matsuoka, Kawachiya et Tanaka (1997b) développent un système appelé *Pegasus* dont la particularité est de ne pas tenter d'inférer automatiquement les relations entre les traits, mais plutôt de les proposer à l'utilisateur.

³ Une contrainte négative fait en sorte qu'une condition ne soit pas atteinte.

De façon plus simple, Julia et Faure (1995) reconstruisent les jonctions entre les segments par l'application de « zones d'attraction », celles-ci étant définies en fonction de la boîte englobante d'un trait. Selon cette méthode, toute jonction située aux abords de l'extrémité d'un trait pourra être reconstruite si les traits sont jugés suffisamment près les uns des autres. À ce sujet, Hutton et al. (1997) notent l'importance de la séquence : les traits qui sont dessinés en séquence ont de plus grandes chances d'être reliés.

Nous avons montré dans cette section que la reconnaissance en ligne de croquis est problématique en ce qui a trait à la qualité du signal : diverses techniques doivent être employées pour nettoyer et dans certains cas, embellir le tracé. L'embellissement par contre n'est pas facilement réalisable sans que le processus créatif de l'utilisateur ne soit gêné.

1.2.3 Séparation des symboles et des lignes de connexion

Certaines techniques sont propres à la reconnaissance de diagrammes et de formes interconnectées, notamment les techniques servant à la séparation des symboles et des lignes de connexion et l'analyse syntaxique bidimensionnelle. Cette courte section est consacrée à un bref survol de ces méthodes.

De nombreux auteurs abordent le problème de la reconnaissance de diagrammes et de formes interconnectées en faisant d'abord la séparation des symboles et des lignes de connexion ; Kojima et Toida (1988) font la segmentation des figures en temps réel à l'aide d'heuristiques. Selon l'hypothèse qu'une extrémité d'un segment de ligne établit la connexion avec une forme, les auteurs avancent que si l'extension d'un segment de ligne croise le centre d'une forme, alors celui-ci doit être une ligne de connexion. Quant à eux, pour établir les connexions entre les symboles, Murase et Wakahara (1985) emploient quelque 52 règles de connexion (du genre « si-alors ») pour le domaine des organigrammes, tandis que Yu, Samal et Seth (1997) accomplissent la même tâche avec des heuristiques.

À leur façon propre, une fois les formes de base reconnues, Julia et Faure (1995) les soustraient du diagramme et classifient les traits restant comme lignes de connexion entre les formes. Le diagramme est ensuite rectifié et réorganisé par l'application d'une heuristique selon laquelle à un même niveau d'abstraction, la boîte englobante de chacun des symboles a la même dimension.

Sur ce point, nous préconisons la même approche que Julia et Faure. La nature interactive et progressive de notre domaine d'application fait en sorte qu'il est impossible de connaître à l'avance tout le contenu du dessin. La reconnaissance des symboles est réalisée au fur et à mesure que les tracés apparaissent et la séparation des lignes de connexion est complétée à ses dépens.

1.2.4 Représentation des connaissances

Blostein (1996) identifie trois catégories de connaissances propres au domaine de la reconnaissance de diagrammes :

1. la connaissance spécifique au *diagramme en cours* : généralement, il s'agit d'une structure de donnée représentant les étapes de l'interprétation,
2. la connaissance propre à la *notation du diagramme* : ce sont les conventions servant à la représentation du dessin,
3. la connaissance propre à la *reconnaissance de diagrammes* : opérations générales (pré-traitement et segmentation propres au domaine de la reconnaissance de graphiques).

Poursuivant dans cette direction, Kato et Inokuchi (1990), dans le domaine de la reconnaissance de symboles et de diagrammes faits à la main, représentent les formes à l'aide de 3 primitives : lignes droites, cercles et arcs. Accompagnées des relations de position entre elles, ces primitives serviront d'éléments de structure en vue de la définition de formes plus complexes. En effet, l'idée de représenter des formes évoluées à partir de formes simples est un concept fréquemment utilisé ; notamment, Zhao, Liu, Zheng, Pan et Yuan (1997) caractérisent les symboles mathématiques par 3 vecteurs à l'aide d'un ensemble d'éléments de structure de base, ainsi que des

relations de position et de séparation entre ceux-ci. De façon analogue, Kojima et Toida (1988) reconnaissent deux types de formes : 1) les « figures fondamentales », composées de lignes, d'arcs ou des traits fermés et 2) les « symboles », constitués de figures fondamentales et caractérisés par les positions relatives entre celles-ci.

En reconnaissance en ligne de croquis, Saund et Moran (1994) définissent l'élément de représentation fondamental en termes de « formes symboliques » ; les attributs sont des caractéristiques géométriques, d'emplacement, d'orientation, d'échelle, de lien avec les autres formes et de type. Ces éléments sont organisés de façon particulière : la méthode cherche à mimer le système perceptif humain en offrant la possibilité de représenter les objets en termes de « niveaux hiérarchiques ». Au niveau le plus bas, on retrouve les fragments de courbes, objets dits premiers, et à chaque niveau supérieur, les objets composés. Cette technique, conceptuellement très proche de la nôtre, est cependant conçue pour un assemblage subjectif des tracés selon des critères perceptifs.

À l'aide d'un langage descriptif, Mohr (1990) décompose les formes en sous-formes jusqu'au niveau des primitives que sont les lignes et les arcs de cercles. Au niveau d'abstraction le plus bas, ces primitives sont groupées en fonction de l'angle qui les relie et, à chaque niveau supérieur, en fonction d'opérateurs d'assemblage et d'opérateurs topographiques. Par un langage descriptif moins complexe mais similaire, Murase et Wakahara (1985) représentent la structure des symboles à l'aide d'un ensemble de traits, chacun étant de type ligne ou arc et défini par deux points. Une technique similaire est employée par Pasternak et Neumann (1993) qui, pour définir les lignes et arcs, utilisent leurs attributs géométriques (position, orientation, boîte englobante, etc.). Ils représentent les symboles à l'aide d'un langage de spécification déclaratif, permettant ainsi une définition significative des objets graphiques qui, selon les auteurs, peut être adaptée à divers domaines.

Fonseca et Jorge (2000) quant à eux n'emploient que des attributs géométriques pour définir les formes. D'abord, ils calculent la boîte englobante ainsi que le plus grand triangle et le plus grand parallélogramme inscrits ; ensuite, à l'aide de ces derniers, ils

calculent des valeurs caractéristiques pour classer les formes. Conceptuellement proche de cette dernière, la méthode de Gross (1996) permet la reconnaissance de glyphes (tracé résultant du poser au lever de crayon) composés de un ou plusieurs traits ; les primitives utilisées pour la représentation sont le nombre de segments, la boîte englobante, l'échelle, le rapport de l'aspect, les coins et le chemin du tracé dans une grille de dimension fixe. Une troisième méthode qui s'appuie sur des caractéristiques statistiques est celle de Julia et Faure (1995) ; les formes y sont définies en fonction d'un ensemble de segments, de directions principales, de la boîte englobante et de ses « points sensibles », zones d'intérêt définies autour de chacun des sommets de la boîte englobante.

Enfin, dans le domaine de l'interprétation et de la conversion de dessins d'ingénierie, Yu et al. (1997) expriment les symboles, leur emplacement et leurs interrelations à l'aide d'un graphe dans lequel les nœuds et arêtes représentent respectivement les symboles et connexions. Les symboles comprennent une ou plusieurs composantes, chacune d'elle étant faite d'une boucle ou d'un tracé ; la connaissance du domaine, nécessaire pour l'identification des symboles, est par conséquent saisie à l'aide de deux bibliothèques : une pour les tracés et l'autre pour les symboles. Cette méthode s'inspire fortement des travaux de Kasturi et al. (1990a) qui utilisent une liste de lignes droites et de courbes, mais surtout des travaux d'Okazaki, Kondo, Mori, Tsunekawa et Kawamoto (1988), qui basent la représentation sur les symboles à boucle fermée, desquels sont extraites certaines caractéristiques géométriques et de forme.

Dans le même esprit, Fahn, Wang et Lee (1988) caractérisent les primitives en fonction de la tendance (droite, courbe ou zigzag) et la déviation du tracé ; les segments ainsi identifiés sont représentés sous la forme d'une chaîne de vecteurs orientés. La méthode de Messmer et Bunke (1998a) définit elle aussi les tracés à l'aide d'un graphe attribué. Leur technique consiste à représenter chaque segment de ligne du dessin par un nœud du graphe et les relations entre ces segments (i.e. l'angle) par des connexions entre les nœuds. Enfin, Lee, Kim, et Groen (1990) proposent une technique semblable,

mais bonifient la représentation des caractéristiques statistiques des formes (nombre et variance des traits, de la courbure, etc.).

Nous nous inspirerons des techniques qui représentent les formes par un ensemble de primitives de base ainsi que des relations entre celles-ci, notamment, les méthodes de Yu et al. (1997), Zhao et al. (1997) et Kojima et Toida (1988). Notre schème de représentation sera également influencé à la fois par les travaux de Kato et Inokuchi (1990), qui représentent les symboles et connexions par un graphe orienté, et par ceux de Saund et Moran (1994) qui, à l'aide de leur approche multi-échelle, permettent la gestion de l'incertitude en rendant l'interprétation des données flexible et variable.

Au chapitre de la reconnaissance des attributs statistiques, la méthode de Fonseca et Jorge (2000), basée sur les attributs géométriques de la forme, ainsi que celle de Lee et al. (1990), centrée sur l'analyse des distributions, des moyennes et des variances, serviront de base à l'élaboration de notre technique. Si nous sommes d'avis que les techniques qui expriment les symboles selon leur structure et les interrelations entre les composantes sont préférables pour notre domaine, nous nous confrontons à la problématique de les appliquer à la reconnaissance de croquis de symboles évolués alors qu'elles furent jusqu'à maintenant majoritairement employées pour la reconnaissance hors-ligne de dessins d'ingénierie ou de schémas simples.

1.2.5 Techniques de reconnaissance

La composante qui sans doute est la plus importante en reconnaissance de formes est le *moteur de reconnaissance*. Celui-ci a pour objectif d'identifier la nature des formes encore inconnues à partir d'un ensemble de connaissances et de méthodes (génériques ou spécifiques).

Selon Kato et Inokuchi (1990), en reconnaissance de symboles et de diagrammes faits à la main, plusieurs types de connaissances devraient être utilisées : connaissances sur la forme des symboles, sur les dessins schématiques, leur sémantique, l'apprentissage et les vocabulaires génériques. En outre, les auteurs avancent qu'on

ne peut avoir recours aux techniques habituelles utilisant des informations spécifiques au domaine quand il est question de traiter des diagrammes faits à la main. Pour leur méthode, bien qu'ils n'en donnent pas les détails, les auteurs choisissent un moteur de reconnaissance hybride qui tient compte de plusieurs types de connaissances et niveaux d'abstraction et qui emploie une philosophie ascendante (*bottom-up*) et descendante (*top-down*).

Plus récemment, Blostein (1996) répertorie quelques-unes des méthodologies connues servant à la reconnaissance de diagrammes, parmi lesquelles le tableau noir, les méthodologies syntaxiques et la réécriture de graphes (grammaires de graphes). Elle conclut en disant :

« il n'existe pas de critère permettant de choisir la méthodologie la plus appropriée pour une problématique donnée, mais les systèmes existants illustrent comment les diverses méthodologies peuvent être adaptées et même combinées ».

S'il est vrai que le choix de la méthode de reconnaissance n'est pas une tâche facile, il n'en reste pas moins que pour plusieurs auteurs, certains grands principes doivent être respectés. La méthodologie choisie permettra notamment au système d'être flexible, de prendre en considération les informations contextuelles et de corriger les erreurs par la rétroaction.

Chhabra (1997) quant à lui fait un survol des techniques de reconnaissance de symboles graphiques ; il présente les articles par domaine d'application car selon lui, *« les travaux en reconnaissance de symboles ont tendance à être hautement spécifiques à l'application »*. En conclusion, l'auteur soulève quelques points importants auxquels il faudrait à son avis s'attarder, notamment celui de l'utilisation de la rétroaction pour la correction des erreurs faites à la segmentation ou à l'appariement. De façon analogue, Kasturi, Chennubhotla, Raman et O'Gorman (1990b) font un état de l'art détaillé des techniques portant sur la reconnaissance de graphiques. Bien que chaque phase ou niveau de traitement soit abordé, les auteurs mettent l'accent sur les techniques de reconnaissance et d'interprétation, au sujet desquelles ils concluent :

« dans les applications plus complexes, telles que le traitement de cartes et de dessins d'ingénierie, des méthodes à base de connaissances qui tiennent compte du contexte ont généralement été utilisées ».

Pour leur part, O'Gorman et Kasturi (1997) dédient un chapitre entier à la reconnaissance de documents graphiques, mais insistent davantage sur les méthodes de reconnaissance. Les auteurs manifestent notamment un intérêt marqué pour les méthodes de reconnaissance structurelles :

« les méthodes d'analyse structurelle sont pratiques pour les formes complexes ou pour celles où plusieurs formes sont interconnectées [...] Pour la reconnaissance de dessins de lignes tels que les organigrammes, tables et diagrammes, l'analyse structurelle des lignes et leurs interconnexions est requise pour donner une description sensée et succincte ».

Il est bon de noter ici que notre domaine d'application souscrit aux critères mentionnés ci-dessus. Puisque les symboles électriques sont de nature complexe, il est assuré que nous allons privilégier une approche structurelle à base de connaissance.

Cependant, sur la question de l'efficacité des méthodes structurelles et syntactiques appliquées à l'analyse des dessins de ligne, Tombre (1996) avance que :

« lors du design de systèmes plus complexes, on rencontre un certain nombre de "problèmes limite" dont la résolution, à première vue, paraît convenir aux méthodes structurelles, alors qu'en réalité, l'efficacité de ces dernières est souvent amoindrie par des difficultés sporadiques ».

À cet effet, l'auteur cite les quatre *problèmes limite* suivants :

1. **sensibilité au bruit et aux erreurs** : le bruit dont discute ici l'auteur est propre au dessins imprimés et la solution proposée, celle d'une forme d'interaction entre les procédés de segmentation et de reconnaissance (par la *vectorisation multi-échelle*) est novatrice, mais *« son efficacité reste encore à prouver »* Tombre (1996). Selon l'auteur, une telle solution peut aussi aider à résoudre le problème de la *sélection d'une région d'intérêt*. En somme, dit-il, *« il faut faire interagir la segmentation bas niveau avec l'analyse haut niveau »*,
2. **généralité** : les systèmes développés à ce jour sont pour la plupart *« spécifiques à un domaine d'application »*. En guise de solution, l'auteur mentionne les travaux de

Pasternak et Neumann (1993), lesquels portent sur une méthodologie générique par l'introduction d'apprentissage de modèles,

3. **inférence du modèle** : la syntaxe d'un domaine d'application est, selon l'auteur, souvent difficile à saisir en une seule tentative; il serait donc souhaitable d'inférer les modèles automatiquement à partir d'exemples,
4. **puissance descriptive** : enfin, la puissance descriptive des grammaires est considérée, mais l'auteur constate que celle-ci est limitée et qu'il s'agit d'un problème général en représentation des connaissances. Comme il y a plusieurs niveaux de connaissances, la solution au problème se situe dans l'utilisation du *contexte* dans lequel cette connaissance est représentée.

Les problèmes énumérés par Tombre (1996) sont importants et furent cités par plusieurs auteurs, mais des solutions existent. La sensibilité des méthodes au bruit peut être corrigée par une technique de réévaluation de la segmentation par rétroaction contextuelle, selon les éléments les mieux reconnus. Par ailleurs, dans la mesure où nous préconisons une approche hiérarchique, nous croyons à juste titre que le problème de la généralité sera amoindri grâce à la facilité d'adaptation de ces techniques. Enfin, en vue de solutionner le problème de représentation des connaissances, nous sommes d'avis que la prise en considération des informations contextuelles sera facilitée en avantageant une méthode où l'interaction entre les éléments graphiques voisins les uns des autres sera possible.

Nous répertorions maintenant quelques-unes des techniques utilisées que nous jugeons pertinentes. Nous verrons d'abord les techniques structurelles et ensuite les techniques statistiques.

1.2.5.1 Techniques de reconnaissance structurelles

Interprétation de la structure et organisation du dessin

Dans le domaine de la reconnaissance de dessins d'ingénierie, plusieurs méthodes emploient des techniques qui se basent fortement sur la structure et l'organisation des données. Okazaki et al. (1988) définissent une méthode hybride ; elle se base sur la

structure des éléments faits de *boucles* fermées et est modulée par l'utilisation d'*heuristiques*. La méthode de reconnaissance de Kasturi et al. (1990a) se base elle aussi sur l'identification de *boucles*, mais accorde cependant une plus grande importance à l'analyse des traits : par exemple, un trait terminal ne pourra faire partie d'une boucle et sera donc ignoré lors du procédé de reconnaissance. Plus récemment, Yu et al. (1997) poursuivent dans le même esprit, celui des boucles, mais en bonifiant la technique d'appariement hiérarchique à l'aide d'une *recherche dans un graphe* : ils reconnaissent d'abord les boucles et tracés, ensuite les composantes et enfin les symboles. De manière analogue, Kojima et Toida (1988) reconnaissent les formes en deux phases (reconnaissance de figures fondamentales et reconnaissance de symboles), décomposées en six étapes dont le but consiste à d'abord extraire les informations du tracé, les boucles, les arcs et lignes, à reconnaître les formes fondamentales (triangles, carrés), à définir les positions relatives en celles-ci et enfin, les symboles.

Recherche dans un graphe avec heuristiques

Pour Murase et Wakahara (1985), les symboles candidats sont identifiés par une recherche dans un *graphe orienté* et les résultats vérifiés de façon descendante par l'application d'*heuristiques* sur les connexions. Fahn et al. (1988) définissent un système semblable dans lequel les classes de primitives sont départagées à l'aide de critères de décision ; les formes sont ici aussi identifiées par une *recherche dans un graphe orienté*, appuyée par l'application de *règles* sur les relations de voisinage, d'adjacence et de topologie. Tout dernièrement, Messmer et Bunke (1998b) présentent une méthode accomplie de recherche dans un *graphe attribué* dans laquelle les symboles sont identifiés par la détermination de l'isomorphisme (problème de la comparaison) de sous-graphes.

Analyse syntaxique avec heuristiques

Alors que Mohr (1990) emploie un algorithme d'*analyse syntaxique globale* qui alterne entre le descendant et l'ascendant, Pasternak et Neumann (1993) définissent un procédé d'interprétation qui se base presque entièrement sur des *heuristiques*.

1.2.5.2 Techniques de reconnaissance statistiques

Appariement au modèle

Julia et Faure (1995) font la reconnaissance de formes simples en deux temps. Le premier moteur agit à la volée et reconnaît les formes (seulement celles faites d'un trait continu) par un *appariement au modèle* ; le second agit en différé et identifie les formes en cherchant à résoudre un *système d'inégalités*. De façon analogue, Gross (1996) classifie d'abord les glyphes à l'aide d'une technique d'*appariement au modèle* de type ascendant et poursuit avec une reconnaissance haut niveau par un procédé graphique de recherche et de remplacement (*search and replace*). Valveny et Marti (1999) présentent une méthode novatrice par laquelle ils tentent de reconnaître les formes par l'appariement à un *gabarit déformable* (*deformable template matching*). Enfin, pour la reconnaissance de symboles mathématiques, Zhao et al. (1997), dont l'ordre d'écriture est fixé par la méthode, emploient quelques *heuristiques* pour faire le classement préliminaire et un *appariement au modèle* pour la classification finale.

Logique floue

Fonseca et Jorge (2000) proposent une méthode novatrice de reconnaissance des formes à l'aide d'*ensembles flous* en se basant sur leurs attributs géométriques.

1.2.5.3 Choix d'une technique

La reconnaissance en ligne de diagrammes faits à la main comporte de nombreux problèmes qui à notre avis ne peuvent être solutionnés que par l'utilisation d'une technique de reconnaissance hybride. Celle-ci devra tirer profit d'un maximum d'informations, autant les données propres à la forme (statistiques et structurelles) que celles appartenant à son contexte.

Selon Kato et Inokuchi (1990), la reconnaissance de croquis schématiques et logiques faits à la main devrait être faite à l'aide d'une méthode hybride qui puisse gérer l'incertitude. En raison de la nature imprécise et bruitée du signal manuscrit en ligne et du caractère approximatif propre au croquis, nous sommes du même avis. Pour cette raison, nous préconisons une approche hybride par laquelle nous tenterons de mettre

en parallèle plusieurs moteurs de reconnaissance, chacun agissant sur un ensemble de primitives unique ou partagé. Dans cet esprit, nous prévoyons une méthode similaire à celle de Lee et al. (1990) par laquelle la reconnaissance s'appuie à la fois sur les caractéristiques structurelles et statistiques de la forme.

Le moteur structurel sera basé sur l'appariement hiérarchique de Yu et al. (1997) : dans la mesure où les symboles sont exprimés selon un modèle hiérarchique (formes composées de sous-formes), il va de soi que la reconnaissance se fasse de manière progressive selon une complexité croissante. Ainsi, les formes simples seront d'abord reconnues et de celles-ci les symboles identifiés. La méthode statistique par logique floue de Fonseca et Jorge (2000) fait la reconnaissance à l'aide des propriétés géométrique des traits dont un symbole est composé ; bien qu'elle ne reconnaisse que les formes simples, nous croyons qu'il sera possible de la transformer afin de tenir compte des paramètres statistiques de la structure, tel que le font Lee et al. (1990) et Gross (1996).

Enfin, par souci de généralité, nous reconnaissons l'importance de séparer la reconnaissance des formes simples et des formes complexes. Le moteur hiérarchique, de par sa nature, facilite cette approche. Un peu à la manière des moteurs à deux temps de Julia et Faure (1995) et Kojima et Toida (1988), notre système pourra ainsi reconnaître en premier lieu les formes simples (ou primaires) communes à toutes les composantes et en second lieu, les transmettre au moteur de reconnaissance de formes spécialisées.

1.2.6 Techniques périphériques (rétroaction, incertitude)

Les méthodes employées en reconnaissance de croquis doivent faire preuve d'une certaine flexibilité car leur nature esquissée les rend souvent très approximatifs. Cette flexibilité se traduit généralement par l'utilisation de facteurs d'incertitude qui, lorsque utilisés en conjonction avec des méthodes de rétroaction, peuvent être d'une grande pertinence pour le succès de la reconnaissance. Nous voyons ici comment les divers systèmes étudiés gèrent l'incertitude, la rétroaction et les processus parallèles ou

intelligents dont le but est de permettre un traitement haut niveau du problème par une approche globale.

Sur la question de l'incertitude, Blostein (1996) discute du procédé de reconnaissance et mentionne deux stratégies de contrôle de base pour en faire la gestion :

1. *procédé séquentiel avec une liste d'alternatives* : ici, toutes les interprétations possibles sont transférées d'un niveau de reconnaissance à l'autre,
2. *rétroaction contextuelle* : l'exécution concurrente des procédés de reconnaissance permet une rétroaction contextuelle de plus haut niveau qui peut compenser le bruit dans le signal d'entrée (approche descendante).

Ces stratégies prennent forme dans les travaux de Kato et Inokuchi (1990) qui, tout au long du processus de reconnaissance, emploient des facteurs d'incertitude et rendent prioritaire la reconnaissance des symboles dont le facteur de certitude est élevé ; une fois ceux-ci reconnus, il est plus facile d'identifier les lignes de connexion entrant et sortant, et ainsi de reconnaître les symboles à faible niveau de certitude. Selon les auteurs, aucune segmentation stricte ne doit être faite ; des hypothèses de symboles doivent être calculées avec des facteurs de certitude. Au pré-traitement, par exemple, un facteur de certitude est attribué à chaque trait en fonction de la similarité à une forme connue (ce trait est une ligne à 86%).

Le système de Gross (1996), quant à lui, permet les raffinements successifs et, à chaque niveau de reconnaissance et d'analyse, la représentation interne tolère l'ambiguïté et l'incertitude et peut à la fois être rendue plus formelle et structurée. Mohr (1990) inscrit un mécanisme de correction d'erreurs dans son moteur d'analyse syntaxique et note enfin que : 1) dans la mesure du possible, à chaque phase de l'analyse, il est important de conserver autant d'informations que possible afin d'aider la détection et l'élimination des erreurs et 2) l'algorithme de correction doit être flexible ; par exemple, d'abord analyser les zones où les informations ne sont pas ambiguës pour aider la prise de décision dans les zones adjacentes.

L'organisation de la librairie de symboles standards utilisée par Yu et al. (1997) permet un appariement hiérarchique qui permet d'identifier les formes avec un niveau de confiance variable. Dans le même esprit que celui de Mohr (1990), une phase d'interprétation automatique débute en partant des symboles qui ont été reconnus avec un niveau de confiance élevé. Quand un symbole est identifié avec une trop grande incertitude, le système lance une réévaluation de la segmentation en modifiant l'algorithme de type division-fusion (*split and merge*). Enfin, Messmer et Bunke (1995) permettent la reconnaissance de dessins bruités en ajoutant à la représentation sous forme de graphe des instances inexactes et déformées des symboles de référence.

Pour leur part, Saund et Moran (1994) présentent une méthode basée sur une structure de données à échelles multiples de type tableau noir (*blackboard*) qui permet plusieurs interprétations d'un dessin et l'utilisation de bases de connaissances pour l'interprétation spécifique à un domaine. Dans cette méthode, les fonctions d'analyse d'images travaillent en permanence en arrière-plan afin de garder à jour une description multi-niveau des structures visuelles. Au plus bas niveau de la hiérarchie de cette structure, les formes symboliques sont des objets premiers et au plus haut, des objets composés. La méthode tente de grouper des formes symboliques en employant des procédures de groupement basées majoritairement sur les relations de co-termination et d'alignement. Les autres règles identifient la fermeture, le parallélisme, les coins et jonctions en T ; elles sont appliquées localement par l'utilisation d'un tableau noir espace/échelle (*scale-space blackboard*), structure qui permet un regroupement contextuel des données selon leur emplacement spatial et leur facteur d'échelle.

Selon les recommandations des divers auteurs cités précédemment, notre méthode devra tenir compte de l'ambiguïté et de l'incertitude et devra idéalement être suffisamment flexible pour permettre la mise en œuvre d'une technique de correction rétroactive. Par ailleurs, dans la mesure où elle tient compte des informations contextuelles par l'application de procédés ascendants et descendants, la méthode de Kato et Inokuchi (1990) influencera également notre approche. Enfin, de façon globale, par le choix de la méthodologie de reconnaissance et des heuristiques, nous nous inspirerons des principes cognitifs de l'inspection (reconnaissance et interprétation)

évoqués par Petre (1995), notamment, **savoir anticiper, savoir où chercher et savoir quoi chercher** (par les connaissances spécialisées).

1.3 Justification de notre approche

À la lumière des techniques et méthodes décrites dans les sections précédentes, il nous paraît clair que notre problématique en est une qui ne peut trouver de solution adéquate que dans l'implantation d'une méthode hybride et flexible (Kato et Inokuchi, 1990). Dans cet état de l'art, nous croyons avoir montré l'intérêt qu'ont pour nos recherches les travaux réalisés en reconnaissance de diagrammes et symboles faits à la main, en reconnaissance de croquis et de formes simples, en interprétation et conversion de dessins d'ingénierie électrique et enfin, l'utilité des techniques d'embellissement.

La nature approximative des croquis justifie l'utilisation conjointe d'une technique d'embellissement (Jenkins et Martin, 1992) et d'une méthode qui puisse négocier avec des signaux ambigus telles celles de Kato et Inokuchi (1990) et de Saund et Moran (1994). Par ailleurs, si au chapitre de la reconnaissance de figures simples les méthodes statistiques sont souvent utilisées (Fonseca et Jorge, 2000 et Lee et al., 1990), nous avons vu que pour le cas des formes spécialisées, les méthodes structurelles sont par contre préférées (Kojima et Toida, 1988, Zhao et al., 1997 et Yu et al., 1997). Dans la mesure où notre dictionnaire de formes, présent et futur, souscrit à ces deux types, et en raison de la nature inexacte des croquis, nous sommes d'avis que l'utilisation d'une méthode hybride est un choix approprié.

Enfin, autant en interprétation de dessins d'ingénierie qu'en reconnaissance de diagrammes et de croquis, les divers travaux répertoriés témoignent de l'importance d'appuyer le moteur de reconnaissance de techniques auxiliaires. À ce chapitre, par le moyen de facteurs de confiance attribués aux symboles, nous utiliserons la notion d'incertitude pour traiter le problème du signal approximatif et variable.

Par ailleurs, en raison de la difficulté que pose la représentation des connaissances par une méthode syntaxique appliquée à la description de symboles complexes, nous favorisons l'utilisation ultérieure de données contextuelles en préconisant une approche qui tient compte de la description spatiale et d'une représentation compréhensive du voisinage des éléments graphiques. Finalement, nous reconnaissons la nécessité de corriger la segmentation selon une approche descendante ; si le système développé ne remplit pas actuellement ces capacités fonctionnelles, nous sommes toutefois d'avis que la nature du moteur de reconnaissance favorise l'ajout d'une technique de révision par rétroaction.

1.4 Bibliographie thématique de l'état de l'art

Tableau I

Récapitulation des états de l'art pertinents par domaine d'application

Auteur	Reconnaissance de graphiques			
		Reconnaissance de l'écriture manuscrite en ligne et hors-ligne		
				Embellissement et contraintes
Nagy (2000)	√	√	√	
Plamondon et Srihari (2000)		√		
Loncaric (1998)	√			
Badros (1998)			√	
Tombre, Ah-Soon, Dosch, Habel, et Masini (1998)	√			
Igarashi et al. (1997a)			√	
Chhabra (1997)	√			
Kasturi et Luo (1997)	√			
O'Gorman et Kasturi (1997)	√			
Tombre et al. (1998)	√			
Blostein (1996)	√			
Jenkins et Martin (1993)			√	
Kasturi et al. (1990b)	√			
Nouboud et Plamondon (1990)		√		
Tappert et al. (1990)		√		
Pavlidis (1980)	√			

Tableau II

Récapitulation des articles pertinents en reconnaissance de graphiques par domaine d'application

Reconnaissance de symboles et de diagrammes faits à la main	Reconnaissance en ligne de <u>croquis</u>	Interprétation et conversion de <u>dessins</u> d'ingénierie	Techniques d'<u>embellissement</u> par contraintes
Kato et Inokuchi (1990)	Saund et Moran (1994)	Yu et al. (1997)	Jenkins et Martin (1992) Jenkins et Martin (1993)
Kojima et Toida (1988)	Nakagawa et al. (1993)	Fahn et al. (1988)	Pavidis et Van Wyk (1985)
Murase et Wakahara (1985)	Gross (1996)	Messmer et Bunke (1998a) Messmer et Bunke (1995)	Igarashi et al. (1997b)
Zhao et al. (1997)	Fonseca et Jorge (2000) Jorge et Fonseca (1999)	Lee et al. (1990)	
Julia et Faure (1995)	Apte, Vo, et Kimura (1993)	Kasturi et al. (1990a)	
Mohr (1990)	Zhao (1993)	Okazaki et al. (1988) Hutton et al. (1997) Valveny et Marti (1999) Lank (2000) Pasternak et Neumann (1993) Bunke (1982)	

CHAPITRE 2

MÉTHODE ET ARCHITECTURE PROPOSÉES

2.1 Introduction

Que ce soit sur un bout de papier ou un écran d'ordinateur, l'action de dessiner est un processus interactif par lequel l'utilisateur peut ajouter, enlever ou modifier à souhait les tracés formant les éléments du dessin. De ce fait, les composantes formées par ces tracés sont de nature dynamique car elles peuvent changer selon les modifications que l'utilisateur y apporte (ex. un tracé reconnu comme cercle peut devenir un symbole *source de tension* si l'utilisateur y ajoute un *signe plus* et un *signe moins*). Le croquis prend donc forme de manière incrémentale au fur et à mesure que de nouveaux traits sont ajoutés aux traits existants et les symboles spécifiques émergent des regroupements de tracés perceptiblement significatifs.

Ce concept, celui de l'émergence de signification selon des critères perceptifs, est inspiré des travaux de Saund et Moran (1995) et Côté, Lecolinet, Cheriet et Suen (1998). La signification accordée aux regroupements est définie d'une part selon l'agencement structurel des éléments graphiques (ex. un ensemble de lignes situées dans le même voisinage) et d'autre part selon les relations topologiques qui les unissent (ex. parallélisme, proximité, etc.).

Par ailleurs, en raison de la nature imprécise, bruitée et non-contrainte du signal traité, notre technique doit faire preuve de souplesse et de flexibilité (Kato et Inokuchi, 1990). Nous verrons comment l'utilisation d'un facteur de confiance permet de tenir compte de ces variations.

Notre système est centré sur ces concepts d'interactivité, d'organisation perceptive et de flexibilité, aussi bien que sur l'importance de prendre en compte les interprétations possiblement conflictuelles, partiales et ambiguës d'une scène à divers niveaux

d'abstraction. De plus, comme il est en ligne, notre système compte sur la coopération de l'utilisateur pour faciliter la reconnaissance.

2.2 Système dans son ensemble

Le système que nous avons réalisé est illustré à la figure 7. Il est composé des trois modules de saisie du tracé, de pré-traitement et de reconnaissance. L'utilisateur dessine à l'aide d'un crayon tactile sur une tablette numérique et les points ainsi générés sont reçus par le module de saisie du tracé. Ce dernier a pour tâche de recueillir les points, et leur attribuer un paramètre indiquant le moment auquel chacun d'eux fut saisi. La saisie est initiée lorsque l'utilisateur appuie le crayon sur la tablette et se termine lorsqu'il le soulève ; nous qualifions de *tracé* l'ensemble de points ainsi obtenu.

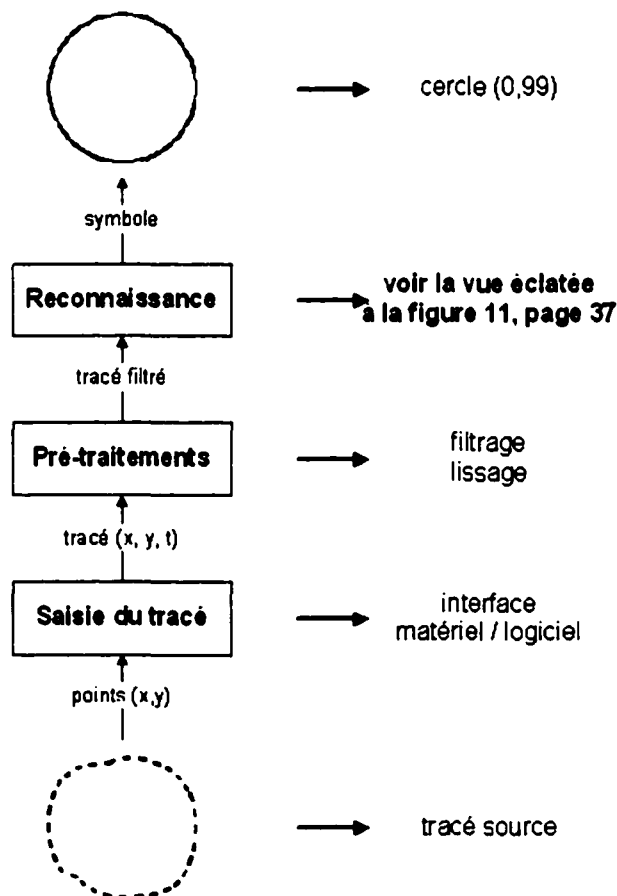


Figure 7 - Système dans son ensemble

Quand un tracé est complété, le module de pré-traitement le prend en charge. Le rôle de l'étape de pré-traitement est de transformer le tracé initial en le nettoyant de ses imperfections de manière à faciliter la tâche au module de reconnaissance. Divers filtres et fonctions de lissage sont alors appliqués au tracé, lesquels agissent directement au niveau de l'organisation des points. À la sortie de cette étape, le tracé est transmis au module de reconnaissance.

La dernière étape est celle de la reconnaissance. Les tracés filtrés sont transmis au module de reconnaissance, lequel a pour objectif de les analyser afin d'identifier les symboles qui pourraient y être présents. Le moteur de reconnaissance agit à plusieurs niveaux, notamment en comparant la structure et la topologie des composantes graphiques à des modèles et définitions de référence. À la sortie du module, chaque symbole est reconnu avec un facteur de confiance qui témoigne de son degré de similitude avec les formes de référence. La section suivante décrit plus en détails le fonctionnement du moteur de reconnaissance.

2.3 Module de reconnaissance

L'objectif du moteur de reconnaissance est d'identifier les symboles de notre domaine à partir d'un ou plusieurs tracés selon la disposition spatiale et l'organisation structurelle de ces derniers au sein du croquis. Comme la sémantique de notre domaine s'exprime par la structure des primitives et les relations topologiques entre elles, il va de soi que notre technique de reconnaissance soit basée sur ces deux aspects, d'autant plus qu'il s'agit là de caractéristiques qui sont invariantes par rapport à la rotation ou la dimension des formes qui doivent être reconnues.

Notre moteur de reconnaissance est dit hiérarchique en raison de la manière progressive par laquelle les symboles sont reconnus, allant de la reconnaissance d'éléments simples, telles les primitives, jusqu'à la reconnaissance d'éléments complexes, tels les symboles. Suivant cette approche et afin de permettre une reconnaissance qui s'appuie sur les principes d'organisation perceptive énoncés plus tôt, nous avons choisi de représenter les éléments du croquis selon une structure

également hiérarchique dans laquelle les éléments sont créés à partir des éléments issus des niveaux inférieurs.

Les éléments graphiques du système sont appelés des *composantes* (un exemple de chacun des trois niveaux est montré aux figure 8, figure 9 et figure 10), lesquelles appartiennent nécessairement à l'une des trois catégories suivantes :

- primitives
- composantes spécifiques
- symboles



Figure 8 - Primitive
« ligne »

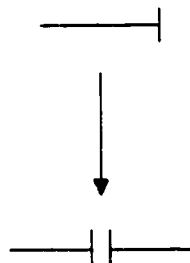


Figure 9 - Composante
spécifique « bout en T »

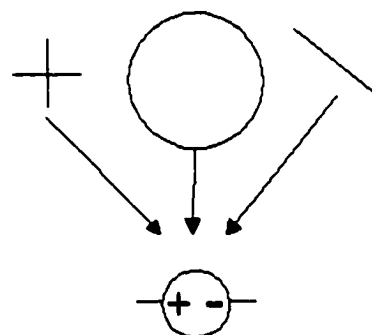


Figure 10 - Symbole « source de
tension »

Un module de reconnaissance est associé à chacune des catégories définies précédemment. Les tâches de chacun de ces modules sont départagées entre des *tâches de traitement*, dont le rôle est secondaire et les *tâches de reconnaissance*, qui jouent un rôle de premier plan. L'objectif de ces dernières est d'identifier de nouvelles composantes en analysant les caractéristiques structurelles et topologiques des éléments qui sont traités. À cette étape, un facteur de confiance est calculé, lequel témoigne de la justesse de la nouvelle composante reconnue.

Les *tâches de traitement* consistent quant à elles à calculer les relations avec les composantes avoisinantes et à diffuser la nouvelle composante à travers le système. On peut voir à la figure 7 comment le moteur hiérarchique agit. Les tracés filtrés et

lissés sont d'abord envoyés dans le module de reconnaissance de primitives ; celui-ci lance un processus de segmentation du tracé qui a pour but d'identifier les primitives de base (ligne, arcs, etc.). Chaque primitive ainsi reconnue est d'abord traitée afin que ses voisins significatifs soient identifiés, est ensuite transmise au module de *reconnaissance de composantes spécifiques* et enfin insérée dans la base de donnée spatiale. Le module de *reconnaissance de composantes spécifiques* agit de façon similaire, à la différence près que son objectif est de reconnaître des composantes complexes à partir de composantes simples.

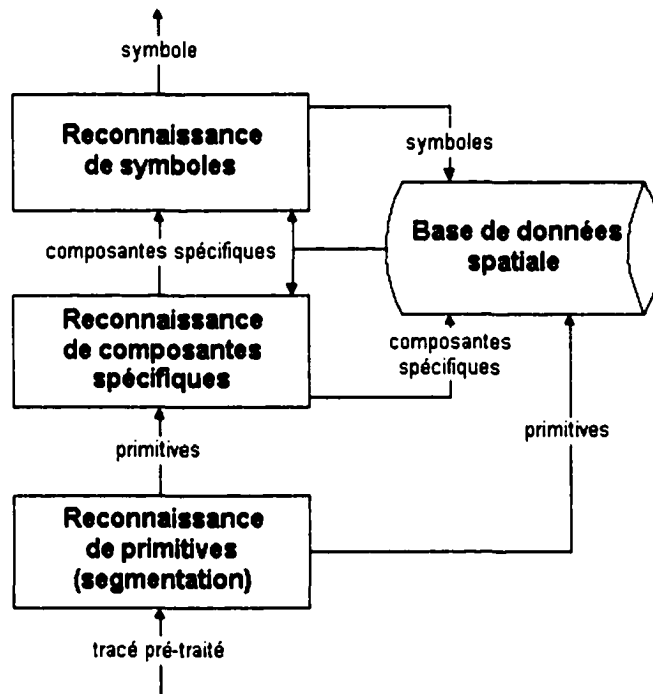


Figure 11 - Module de reconnaissance hiérarchique

Au plus haut niveau de complexité, le module de reconnaissance de symboles cherche à identifier les symboles parmi les composantes qui lui sont fournies. Enfin, notons que chacun des modules pourra se nourrir des composantes présentes dans la base de données spatiale afin d'obtenir un résultat de reconnaissance.

Les sections suivantes expliquent plus en détails le fonctionnement du moteur hiérarchique. Nous décrivons d'abord la structure des composantes et le principe par

lequel celles-ci évoluent à travers le système. Nous montrons ensuite comment fonctionne le cœur du système de reconnaissance et terminons en discutant de la notion d'incertitude.

2.3.1 Structure des composantes

Afin de comprendre notre motivation pour le choix du moteur de reconnaissance hiérarchique, rappelons que les éléments graphiques du système sont des *composantes* qui, en ordre croissant de complexité, peuvent être soit des *tracés d'encre*, des *primitives*, des *composantes spécifiques* ou des *symboles* (voir à ce sujet les figure 8, figure 9 et figure 10).

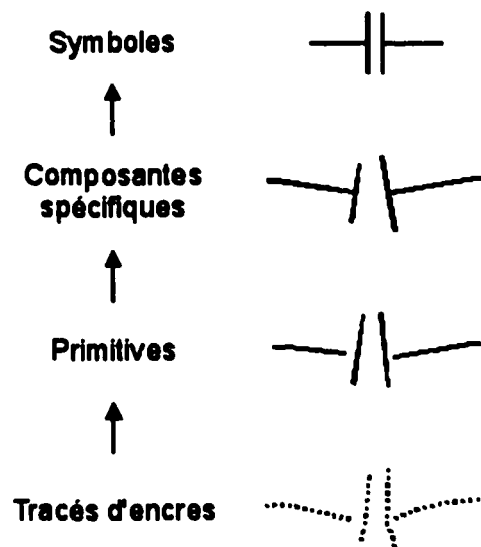


Figure 12 - Niveaux de reconnaissance

Selon notre modèle, une composante d'un niveau donné de complexité peut être un ensemble de composantes provenant de n'importe lequel des niveaux inférieurs ou du niveau même duquel est issue la composante. Prenons le cas des figure 13 et figure 14 pour expliquer ce principe.

Le symbole *condensateur variable* (figure 13), du niveau des symboles, est obtenu à partir de la composante spécifique *flèche* (niveau des composantes spécifiques) et du

symbole *condensateur*, lequel est issu du même niveau structurel (le niveau des *symboles*).

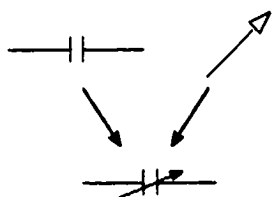


Figure 13 - Structure du symbole *condensateur variable*

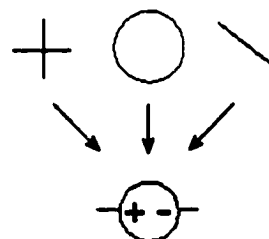


Figure 14 - Structure du symbole *source de tension*

De façon analogue, le symbole *source de tension* (figure 14) est composé d'éléments provenant de deux niveaux différents : du *signe plus*, provenant du niveau des composantes spécifiques et des primitives *cercle* et *ligne*. On voit à la figure 11 comment, par exemple, la boîte *reconnaissance de symboles* reçoit à la fois en entrée des primitives, des composantes spécifiques et des symboles.

2.3.2 Principe de « notification »

Le principe de base par lequel les données circulent entre les divers modules du système est celui de la *notification*. Quand un *tracé d'encre* est dessiné, le module de *reconnaissance des primitives* est notifié. Ce dernier extrait du tracé une ou plusieurs primitives et chacune desquelles notifiera le module de reconnaissance des niveaux supérieurs ; il en va ainsi des autres modules du système.

La notion de notification revêt un rôle important pour solutionner le problème des composantes inactives ou oubliées. En effet, si, à la figure 15, l'utilisateur trace en a) une ligne unique *C1*, celle-ci sera traitée, segmentée, emmagasinée dans la base de données des objets et, en bout de ligne, reconnue comme primitive de type *ligne*. Cependant, comme celle-ci n'a encore aucun voisin, le système ne pourra avancer d'hypothèse supplémentaire quant à son identité ; de ce fait, elle tombera en *latence* et, telle que perçue par le système, deviendra inactive. Quand en b) une deuxième ligne *C2* est tracée, superposée à la première, tout change ; la même suite d'événements se

déroule pour la nouvelle composante qui, contrairement au cas premier, trouve *C1* dans son voisinage et l'ajoute à sa liste de voisins significatifs. C'est à ce moment que *C1* est réactivée et à nouveau envoyée à travers le système. Nous verrons l'importance de la réactivation et de la notification à la section suivante.

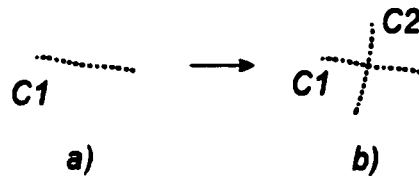


Figure 15 - Composante latente

2.3.3 Moteur de reconnaissance

Le moteur de reconnaissance est divisé en trois modules : un premier pour les primitives, un second pour les composantes spécifiques et un dernier pour les symboles. Le premier module, celui qui reconnaît les primitives, effectue en réalité une segmentation du tracé en une ou plusieurs primitives de base. Une description détaillée de son fonctionnement est donnée au chapitre 5.

Les deux autres modules de reconnaissance fonctionnent selon le même principe fondamental : le moteur tente de reconnaître une *composante spécifique* ou un *symbole* en fonction de l'agencement spatial et structurel d'une composante donnée et de ses voisins. Un *facteur de confiance* est alors calculé par une comparaison des relations topologiques existant entre elles. On peut voir cette technique comme à mi-chemin entre la reconnaissance par logique floue et l'appariement à un modèle déformable. En guise d'exemple, observons la figure 16. En a), l'utilisateur dessine un trait simple ; le moteur ne peut avancer aucune hypothèse. En b), le moteur est notifié à deux reprises : une fois pour la ligne *C2*, qui vient d'être dessinée, et une seconde fois pour la ligne *C1* car un nouvel élément a été ajouté à son voisinage. Ce n'est qu'à l'étape c), lorsque toutes les *conditions requises* sont réunies, que le moteur sera en mesure de reconnaître la composante flèche.

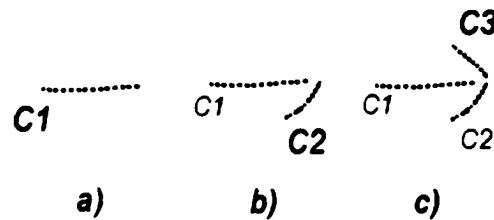


Figure 16 - Reconnaissance de la composante spécifique flèche

Pour l'exemple de la figure 16, trois conditions sont nécessaires afin qu'une composante spécifique de type *flèche* puisse être identifiée :

1. le moteur doit être notifié par la composante de référence (C1 dans ce cas) car le degré de similitude au modèle est calculé par rapport à elle seule,
2. le nombre et type requis de composantes structurantes doit être présent (3 lignes dans le cas de la flèche),
3. la disposition des composantes C1, C2 et C3 les unes par rapport aux autres doit être suffisamment proche du modèle de référence.

Afin de réaliser l'appariement au modèle de référence, le moteur de reconnaissance applique ces trois conditions d'une part à l'aide d'heuristiques et d'autre part selon un ensemble de comparaisons floues sur les relations qui unissent les composantes. Les heuristiques appliquées pour la reconnaissance de la flèche (figure 16) sont les suivantes :

1. **SI** la primitive de notification est une ligne
 2. **SI** le voisin1 est une ligne **ET** le voisin2 est une ligne
 3. **SI** la cardinalité de chacune de ces lignes est simple ou nulle
 4. **SI** les deux voisins sont au-dessus de C1
 5. **SI** les deux voisins ne sont pas du même côté de C1
- ALORS** calculer relations floues

...

Les premier et second tests nous permettent de savoir si nous sommes en présence des bons types de composantes alors que les trois suivants vérifient les conditions topologiques d'admissibilité générale. Si ces conditions sont toutes respectées, on

procède au calcul du facteur de confiance par l'application de comparaisons floues ; les heuristiques servent donc en quelque sorte à limiter le nombre de fois où ce calcul, qui peut être long, sera effectué.

2.3.4 Calcul du degré de similitude

Le degré de similitude (ou de confiance) au symbole de référence est calculé d'après un ensemble de comparaisons flexibles faites entre les relations qui unissent les composantes par rapport à des valeurs définies suivant un modèle de référence. Rappelons ici que la relation (chapitre 1) entre deux composantes prendra toujours une valeur entre 0 et 1, où 0 indique un résultat faible et 1 un résultat fort (ex. un test de parallélisme entre deux lignes perpendiculaire produira le résultat 0). Pour calculer le facteur de confiance global, le facteur de confiance obtenu à partir de chacune des comparaisons sur les relations doit être uniformisé entre 0 et 1 ; nous utilisons pour cela une *transformée de confiance gaussienne* (voir figure 18).

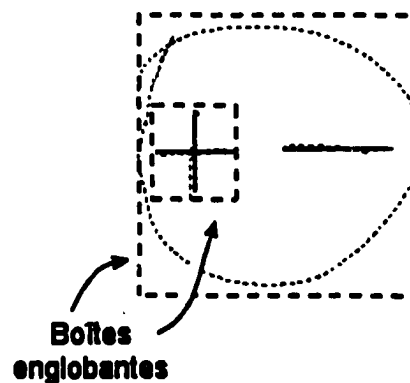


Figure 17 - Boîtes englobantes des composantes *signe plus* et *cercle*

Prenons pour exemple la *relation de dimension* entre deux composantes : dans la figure 17, la boîte englobante du signe plus étant approximativement cinq fois plus petite que celle du cercle, la *relation de dimension* prend la valeur 0,2. Or, dans le symbole de référence, la valeur de cette relation doit être centrée aux environs de 0,15. La *proximité* de la valeur 0,2 à la référence de 0,15, ou plutôt, la *confiance* en cette valeur, est obtenue en transformant 0,2 selon une fonction gaussienne tel qu'on peut le

voir à la figure 18, qui illustre trois transformées de confiance typiques pouvant être ajustées selon la moyenne μ et l'écart type σ .

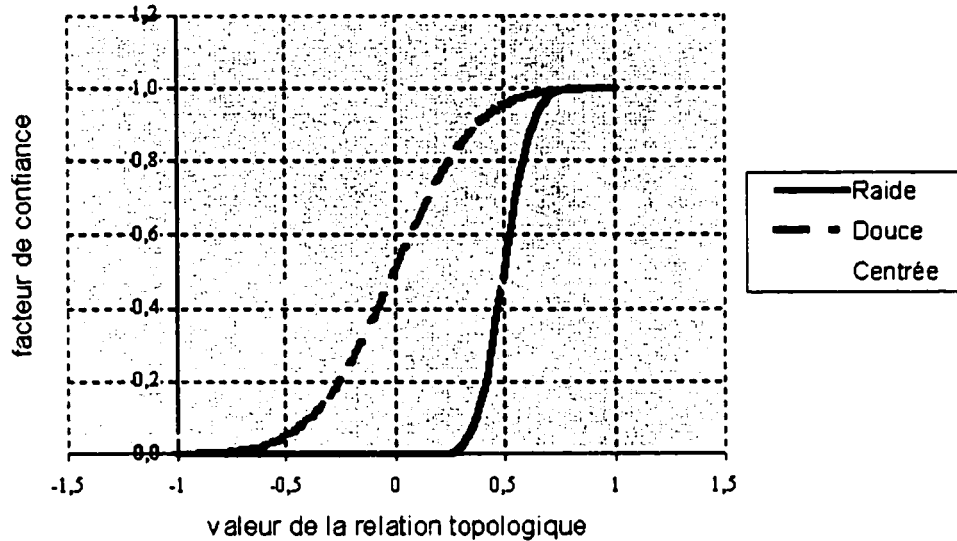


Figure 18 - Transformées typiques pour le calcul des facteurs de confiance

Si nous appelons r_i la valeur entre 0 et 1 d'une relation entre deux composantes et choisissons μ_i et σ_i comme paramètres gaussiens pour la relation i , nous pouvons calculer le facteur de confiance C_{g_i} pour la relation i :

$$C_{g_i} = T_g(\mu_i, \sigma_i, r_i)$$

où T_g est la transformée gaussienne. Nous définissons le facteur de confiance pour la composante que nous cherchons à reconnaître par l'équation suivante :

$$C_t = \frac{\sum_{i=1}^n C_{g_i}}{n}$$

Nous affirmons que l'ensemble des composantes en entrée forme la composante que nous tentons de reconnaître si C_t est supérieur à un seuil déterminé expérimentalement. Dans l'exemple donné plus tôt, si l'on choisissait un écart type de 0,25 (la moyenne étant de 0,15), on obtiendrait la transformée illustrée à la figure 19. D'après cette illustration, on voit que pour une relation de valeur 0,2, indiquée par le

pointillé blanc, donnerait un résultat près du maximum (dans ce cas, la confiance est à 0,98). Si par ailleurs la valeur de la relation de dimension entre les deux éléments avait été plus grande, par exemple, 0,5, la confiance aurait chuté à 0,4.

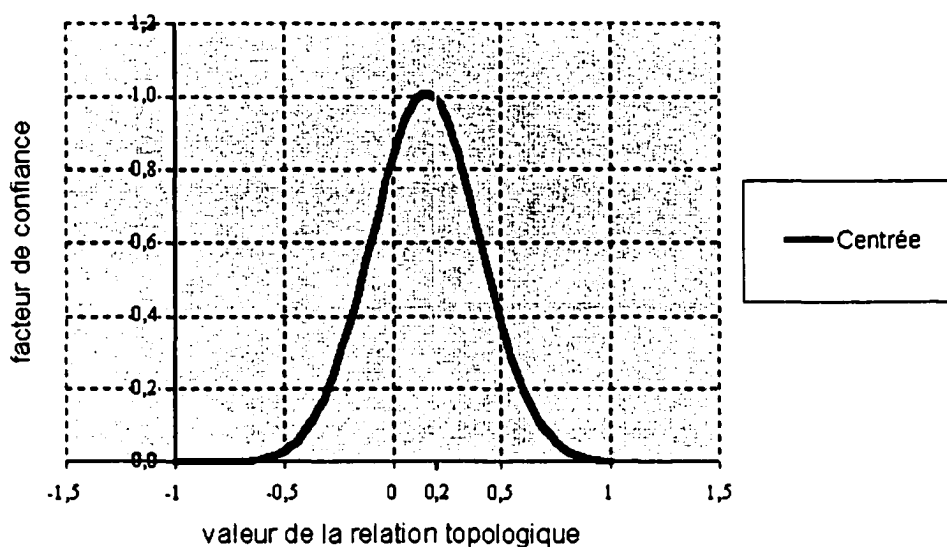


Figure 19 - Transformée de confiance pour un moyenne de 0,15 et un écart type de 0,25

Finalement, notons que nous avons également considéré pour le calcul du facteur de confiance une transformée linéaire simple, mais qu'en définitive, les transitions douces des gaussiennes semblaient plus appropriées à notre domaine, car dans la mesure où nous propageons les facteurs d'incertitude à travers toute la chaîne de décision, nous tentons d'éviter les décisions radicales (soit les valeurs entières 0 et 1).

2.3.5 Tâches de traitement

Pour chacune des nouvelles composantes identifiées, le module de reconnaissance doit accomplir des tâches de traitement qui consistent à calculer les relations de la nouvelle composante et de la diffuser ensuite à travers le système. Au moment où une nouvelle composante est reconnue avec un seuil de confiance suffisamment élevé, elle est ajoutée à la base de données spatiale ; c'est à l'aide de cette dernière qu'il est possible de retracer les voisins situés dans le même voisinage spatial que celui de la

nouvelle composante. La notion de « même voisinage spatial » ou « voisinage significatif » est directement liée à la dimension de l'objet : un gros objet aura un voisinage spatial étendu alors qu'un petite composante aura un voisinage restreint (voir figure 20).

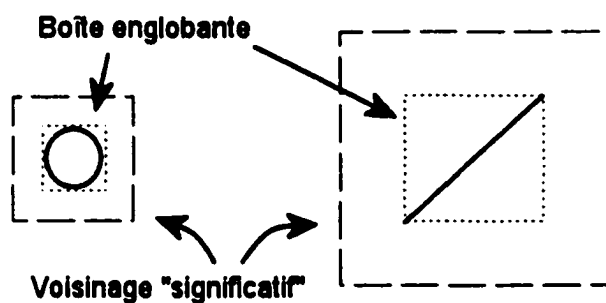


Figure 20 - Voisinage « significatif » pour deux composantes différentes

Une fois les voisins significatifs identifiés, toutes les relations possibles sont calculées entre la composante et ses voisins et les résultats sont stockés afin qu'ils puissent être réutilisés lors de l'étape de reconnaissance subséquente.

2.4 Conclusion

Une méthode de reconnaissance hiérarchique basée sur des principes d'organisation perceptive a été proposée. À partir des observations faites sur les symboles de notre domaine, nous avons élaboré une méthode de reconnaissance qui s'appuie sur la disposition spatiale et l'organisation structurelle des composantes graphiques. Enfin, nous avons montré comment le calcul d'un facteur de confiance nous permet de juger du degré de similitude d'un symbole à une référence.

Maintenant que le moteur de reconnaissance est présenté, nous décrivons dans les chapitres qui suivent comment les relations topologiques sont calculées et de quelle manière se fait l'extraction des primitives. Enfin, nous présenterons brièvement comment nous pré-traitons et embellissons les tracés.

CHAPITRE 3

RELATIONS TOPOLOGIQUES

3.1 Introduction

Nous avons vu dans la section précédente que le moteur de reconnaissance cherche à identifier les symboles en se basant sur les relations topologiques qui unissent des regroupements significatifs de tracés. Celles-ci sont essentielles pour le calcul du facteur de confiance d'un symbole ; c'est à l'aide des relations de parallélisme, de perpendicularité et d'encore bien d'autres qu'il est possible de juger du degré de similitude d'un symbole par rapport à sa référence.

Nous avons choisi les relations topologiques selon un besoin : celui de représenter la structure et la topologie des symboles électriques. Il est donc clair que ce choix est dépendant de notre domaine d'application. Cependant, nous croyons que les relations décrites ici sont suffisamment générales pour exprimer la structure et l'agencement d'un nombre varié de formes.

Dans les prochaines sections, nous verrons comment nous calculons ces relations topologiques⁴. Notons qu'une relation a toujours une valeur entre 0 et 1, où 0 indique un relation nulle et 1 une relation forte. En outre, elle exprime un lien topologique entre deux composantes, C1 étant la « composante source » et C2 la composante qui est en quelque sorte « reliée » à C1.

3.2 Relations topologiques

3.2.1 Connexion

L'indice de connexion permet de savoir si deux composantes partagent un point sur la même primitive de support. Comme le tracé de la figure 21 fut réalisé en un seul trait,

la *relation de connexion* de la composante C1 avec C2 (ici de type **Primitive Ligne**) aura la valeur booléenne **VRAI** (voir figure 22) ; autrement, elle serait fausse.



Figure 21 - Tracé continu

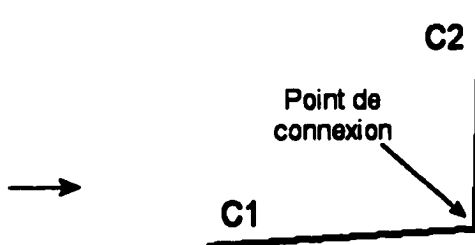


Figure 22 - Primitives connectées, relation de connexion vraie



Figure 23 - Tracés distincts

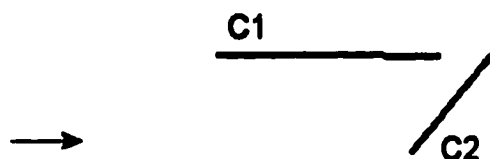


Figure 24 - Relation de connexion fausse

3.2.2 Distance de connexion

La distance de connexion est une mesure de la position du centre de C2 par rapport à la ligne directrice de C1. Cette relation ne peut être calculée que si au moins une des deux composantes reliées possède une direction (i.e., une ligne a une direction alors qu'un cercle n'en a pas). Elle est utile lorsque l'on cherche à savoir si deux lignes forment une intersection en T.

⁴ La méthode employée pour le calcul des relations topologiques est spécifique à nos travaux de recherche.

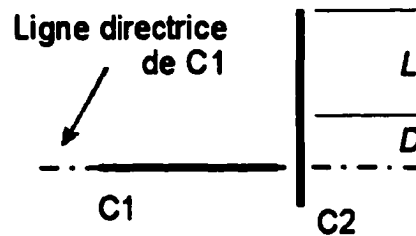


Figure 25 - Calcul de la relation distance de connexion

La mesure de distance de connexion est la suivante :

$$\text{distance de connexion} = \frac{D}{L}$$

où L représente la demi-longueur de $C2$ et D la distance entre le point où $C1$ (ou la prolongation de $C1$) croise $C2$ et le centre de $C2$. Les tracés illustrés aux figure 26 et figure 27 montrent deux cas opposés de relation de distance de connexion.

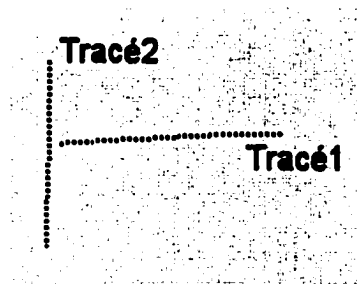


Figure 26 - Relation distance de connexion élevée

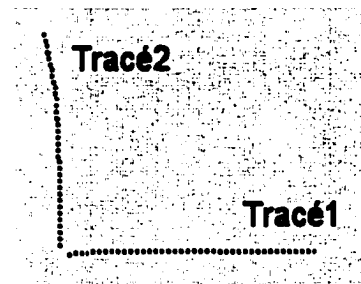


Figure 27 - Relation distance de connexion faible

3.2.3 Degré de contact

Le contact entre deux composantes est une mesure de leur proximité : une valeur de 1 indique qu'elles se touchent à peine alors qu'une valeur de zéro nous informe que leurs frontières se chevauchent, ou encore, qu'elles sont largement éloignées.

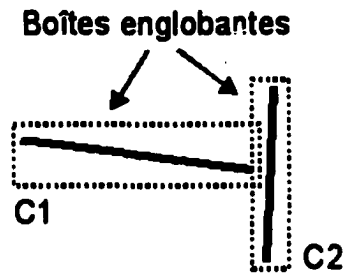


Figure 28 - Relation de contact entre deux composantes

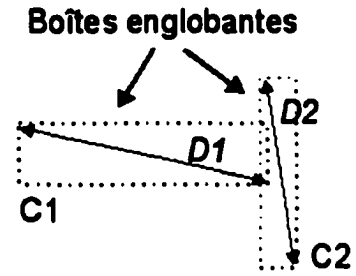


Figure 29 - Dimension des boîtes englobantes

Le calcul de la relation de contact se fait en prenant l'intersection des deux boîtes englobantes de C1 et C2. Le degré de contact est obtenu en comparant la dimension $D3$ du nouveau rectangle ainsi créé avec la dimension $D2$ à l'aide du calcul suivant :

$$\text{degré de contact} = 1 - \frac{D3}{D2}$$

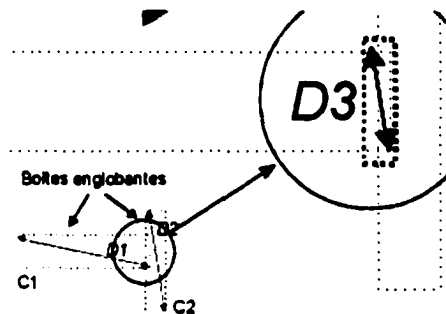


Figure 30 - Intersection des boîtes englobantes de C1 et C2

3.2.4 Degré de similitude

Le degré de similitude sert à exprimer la différence de « dimension » entre deux composantes. On l'obtient avec l'équation suivante :

$$\text{degré de similitude} = \frac{\min(D2, D3)}{\max(D2, D3)}$$

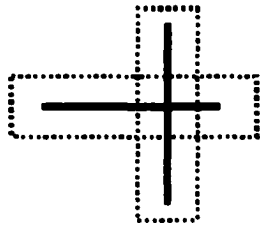


Figure 31 - Degré de similitude = 1.0



Figure 32 - Degré de similitude = 0.5

3.2.5 Degré de concentricité

Cette valeur permet de mesurer à quel point deux composantes partagent le même point central. Nous la calculons en comparant la distance D qui sépare les deux centres à la dimension de la plus grande des deux composantes.

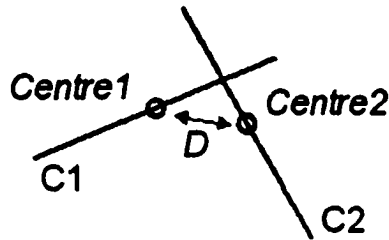


Figure 33 - Calcul du degré de concentricité

De cette façon, la concentricité sera uniforme, autant pour les grands que les petits voisins. L'équation qui la définit est la suivante :

$$\text{degré de concentricité} = 1 - \frac{D}{\max(D1, D2)}$$

La figure 34 illustre le degré de concentricité pour quelques situations typiques.

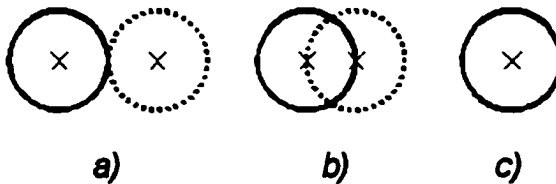


Figure 34 - Degré de concentricité en a) 0,0, en b) 0,5 et en c) 1,0

3.2.6 Degré de parallélisme

La mesure de parallélisme entre deux composantes est obtenue par une comparaison de leur orientation. La formule pour son calcul est :

$$\text{degré de parallélisme} = \cos(\alpha)$$

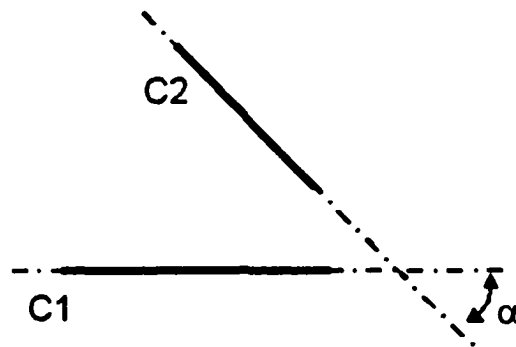


Figure 35 - Mesure du parallélisme

La figure 35 illustre comment la mesure est réalisée et la figure 36 les valeurs que peut prendre le facteur parallélisme.

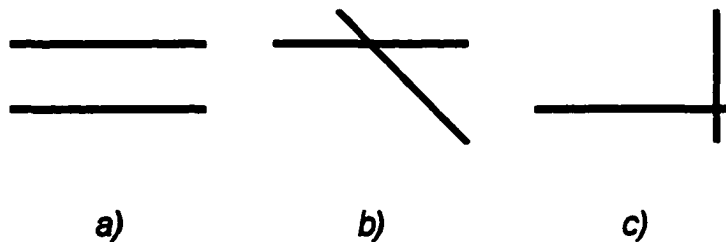


Figure 36 - Valeurs de parallélisme. a) 1,0, b) 0,71 et c) 0,0

3.2.7 Degré de colinéarité

La colinéarité est calculée en fonction des deux relations *parallélisme* et *distance de connexion* ; elle permet de savoir à quel degré deux composantes sont alignées selon le même axe. Le degré de colinéarité est calculé à l'aide de la formule suivante :

$$\text{degré de colinéarité}^5 = \frac{\text{degré de parallélisme} + (1 - \text{distance de connexion})}{2}$$

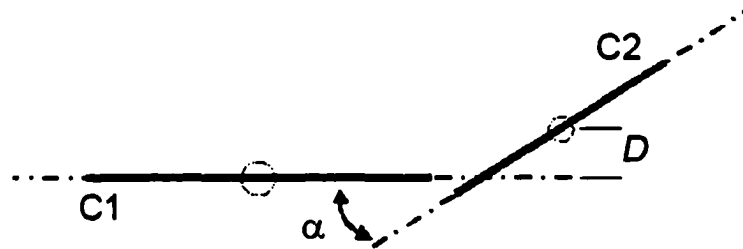


Figure 37 - Facteurs pour le calcul du degré de colinéarité

3.2.8 Degré de codirection

Le degré de codirection permet de savoir à quel point deux composantes partagent la même orientation. Il est obtenu en faisant le produit scalaire des vecteurs exprimant l'orientation de chacune des composantes à mettre en relation (voir figure 38) :

$$\text{degré de codirection} = v1 \cdot v2$$

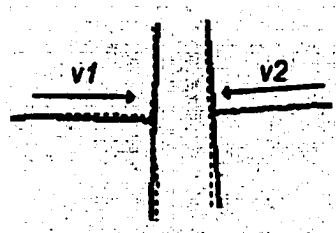


Figure 38 - Vecteurs de direction pour le calcul du degré de codirection

3.2.9 Degré de perpendicularité

La mesure de perpendicularité entre deux composantes est obtenue par une comparaison de leur orientation (voir figure 35). La formule pour son calcul est :

⁵ Une façon plus efficace de calculer le degré de colinéarité serait de prendre le *produit* plutôt que la *moyenne* de la distance de connexion et du degré de parallélisme.

$$\text{degré de perpendicularité} = \sin(\alpha)$$

Le degré de perpendicularité correspond à l'inverse du facteur de parallélisme. Pour ce calcul, nous avons choisi la fonction sinus en raison de sa forme et selon notre désir de donner de la flexibilité aux résultats des relations en prenant des décisions optimistes : si l'angle entre deux composantes fait 45°, le résultat sera plus proche du 1 (relation forte) que du 0 (relation nulle).

3.2.10 Degré de superposition

Le degré de superposition est un indice de la portion de C1 recouverte par C2. On l'obtient en calculant d'abord le rectangle C3 formé par l'intersection de C1 et C2 et en comparant son aire (A3) avec celle de C1 (dont l'aire est A1).

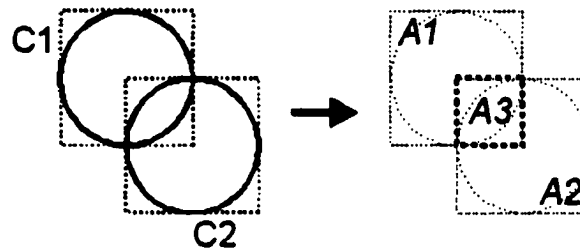


Figure 39 - Calcul du degré de superposition

La mesure du degré de superposition est obtenue par l'équation suivante :

$$\text{degré de superposition} = \frac{A3}{A1}$$

La figure 40 illustre les valeurs que peut prendre le degré de superposition.

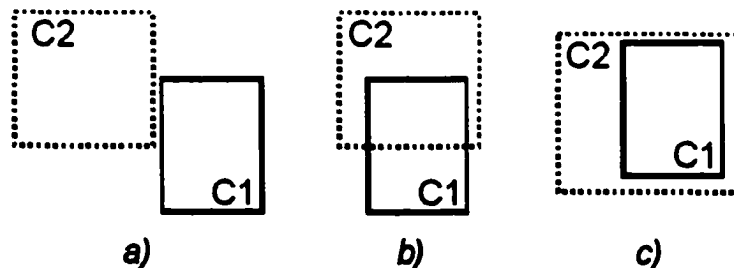


Figure 40 - Valeurs de superposition. a) 0,0, b) 0,5 et c) 1,0

3.2.11 Facteur gauche / droite

Le facteur gauche / droite, défini en fonction de la distance D du point centre de C2 par rapport à la ligne directrice de C1 (voir figure 41), peut prendre trois valeurs : GAUCHE, DROITE et AUCUN.

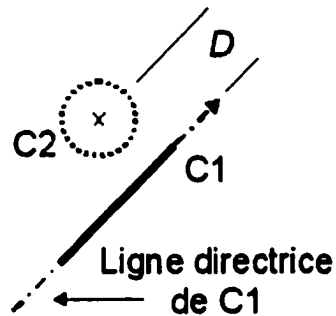


Figure 41 - Calcul de la relation gauche / droite

Il est bon de noter que la composante C1 doit posséder une *direction* (généralement celle dans laquelle le trait a été dessiné). La figure 42 illustre les valeurs possibles de cette relation.

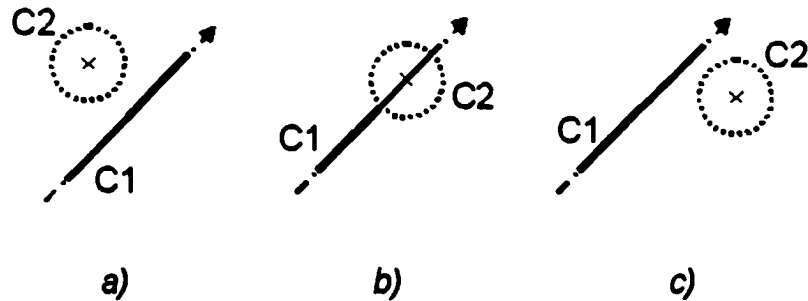


Figure 42 - Valeurs possibles de la relation gauche/droite. a) GAUCHE, b) AUCUN et c) DROITE

3.2.12 Facteur avant / arrière

Cette relation, analogue à la relation gauche / droite, sert à déterminer si un voisin est situé surtout vers l'*avant* ou surtout vers l'*arrière* d'une composante. Une droite $D1$, directrice de $C1$ est d'abord tracée selon l'orientation de $C1$ (voir figure 43). Ensuite, nous calculons $P1$, perpendiculaire à $D1$ et passant par le centre de celle-ci, comme référence pour le calcul de la relation avant / arrière : tout voisin situé à gauche de $P1$ sera considéré comme étant vers l'*arrière* de $C1$ et inversement, tout voisin situé à droite de $P1$ sera considéré comme étant vers l'*avant* de $C1$.

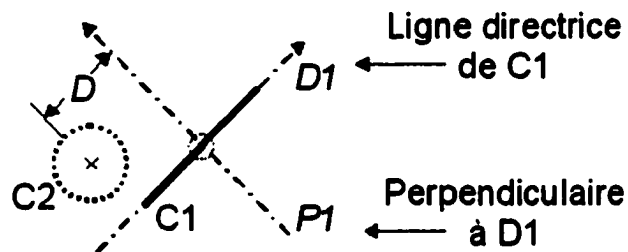


Figure 43 - Calcul de la relation avant / arrière

Les valeurs que peut prendre la relation avant / arrière sont montrées à la figure 44.

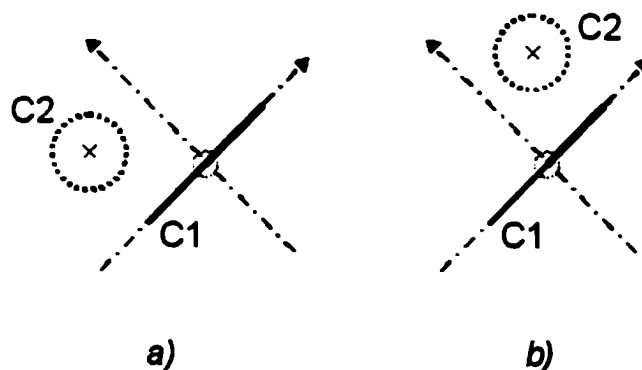


Figure 44 - Valeurs de la relation avant / arrière. a) ARRIÈRE, b) AVANT

3.2.13 Facteur intérieur / extérieur

La relation de contenance intérieur / extérieur est de type booléenne ; C2 sera définie comme étant à l'*intérieur* de C1 si sa boîte englobante est entièrement recouverte par celle de C1 et inversement, C2 sera à l'*extérieur* si les boîtes englobantes ne se touchent pas.

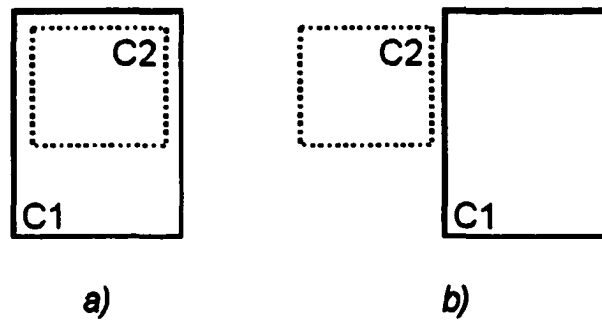


Figure 45 - Valeurs de la relation intérieur / extérieur. a) INTÉRIEUR, b) EXTÉRIEUR

3.2.14 Ordre temporel

L'ordre est une valeur booléenne qui permet de connaître la séquence dans laquelle les composantes furent dessinées. Comme l'information temporelle est stockée dans chaque point d'un tracé, il est simple de vérifier si une composante est apparue avant ou après son voisin dans l'espace.

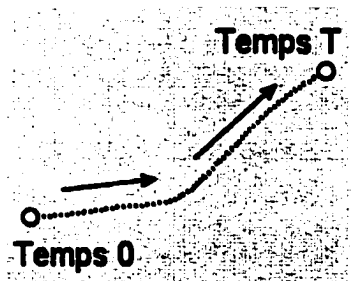


Figure 46 - Relation ordre temporel

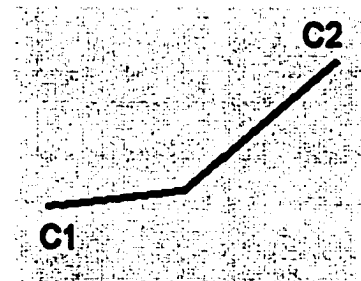


Figure 47 - Relation temporelle : C1 est apparue avant C2

3.2.15 Distance

Le facteur distance sert à quantifier l'éloignement de C2 par rapport au centre de C1 ; il est obtenu en prenant le rapport de D (distance du centre de C2 projetée sur la directrice de C1) sur L (demi-longueur de C1).

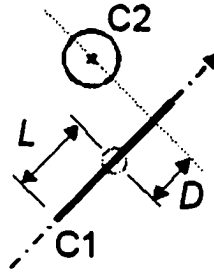


Figure 48 - Calcul de la relation « distance »

Diverses valeurs de *distance* sont illustrées à la figure 49.

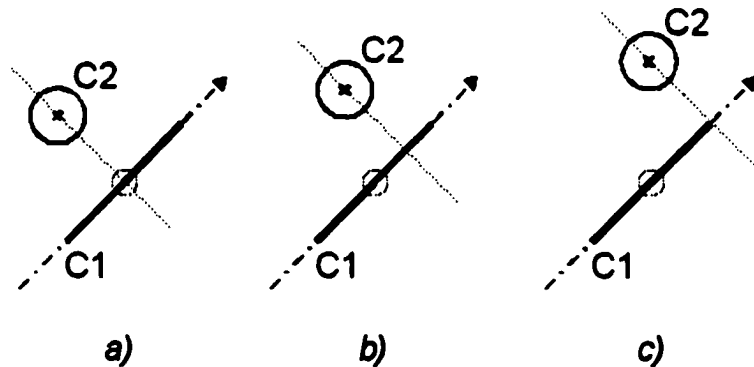


Figure 49 - Valeurs de la relation *distance*. a) 0,0, b) 0,5, c) 1,0

3.3 Conclusion

Nous avons vu en détails dans cette section les relations topologiques qui sont utilisées par le moteur de reconnaissance pour l'appariement des symboles avec leur modèle de référence. Dans la section suivante, nous décrivons la technique employée pour l'extraction des primitives.

CHAPITRE 4

EXTRACTION DES PRIMITIVES

4.1 Introduction

L'extraction des primitives est le procédé par lequel un tracé est analysé afin d'en identifier les parties significatives ou les caractéristiques de forme. Dans la littérature, on réfère à cette problématique comme celle de la segmentation de tracés ; pour notre part, les méthodes qui cherchent à qualifier la nature de la courbure d'un tracé seront d'un intérêt particulier.

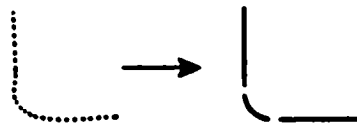


Figure 50 - Procédé de segmentation du tracé

Les premières techniques de segmentation consistent à observer les changements d'angles, ou encore les points d'inflexion, en un point donné du tracé. Ces techniques, bien que simples, donnent de bons résultats et sont robustes (Rosenfeld et Johnston, 1973 et Rosenfeld et Weszka, 1975). Deux techniques plus spécialisées, celles de Rosin et West (1989) et de O'Gorman (1988), permettent la segmentation de tracés en lignes et en arcs, ce qui est précisément ce que nous cherchons à accomplir.

Plus récemment, des méthodes plus complètes, mais aussi plus complexes, furent proposées par Saund (1993), Wuesher et Boyer (1991), Deguchi (1988), Wu et Wang (1993) et Teh et Chin (1989) ; celles-ci cherchent généralement à améliorer les méthodes définies précédemment par l'application d'analyses multi-échelle, de filtrage adaptatif par une meilleure analyse du voisinage de chaque point (région de support) ou encore, comme c'est le cas dans Saund (1993), par une analyse plus poussée du contexte de la courbure par des critères subjectifs ou par une analyse du profil d'accélération.

4.2 Choix d'une méthode de segmentation pour l'extraction des primitives

La vaste majorité des techniques dont il est question à la section précédente servent à segmenter des tracés provenant de documents statiques bidimensionnels numérisés. Ces méthodes s'appliquent donc avant tout à un type de tracé où les points sont voisins les uns des autres et où la pente ne peut prendre qu'une des huit directions de *Freeman* (code de *Freeman* en 8-connectivité, voir figure 51).

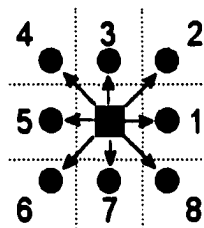


Figure 51 - Code de Freeman pour connectivité d'images raster

Par opposition, les points qui forment un tracé dynamique sont disposés dans l'espace bidimensionnel à des distances variables et par le fait même, à des angles variables.

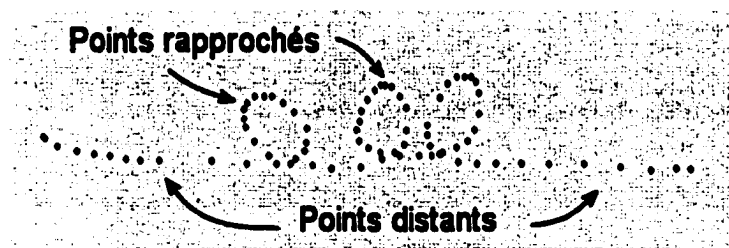


Figure 52 - Caractéristiques du tracé dynamique

Cependant, dans les deux cas, la problématique posée par l'analyse de la courbure consiste à évaluer la différence d'angles entre un ou plusieurs points successifs du tracé. Par la suite, la segmentation du tracé en lignes et en arcs est réalisée par une analyse de cette courbure.

Notre objectif pour l'étape de segmentation était dans un premier temps de réussir à identifier, dans un tracé, les arcs et les lignes. Pour cette tâche, la technique simple des K-Cos décrite par Rosenfeld et Weszka (1975) nous semblait appropriée. Cette méthode par contre ne sert qu'à différencier les zones de courbure élevée des zones de faible courbure ; pour qualifier la nature des arcs, nous nous sommes inspirés de la technique de Rosin et West (1989).

4.3 Description de la méthode de segmentation

Dans cette section, nous présenterons d'abord la technique utilisée pour définir le profil de courbure du tracé et ensuite celle par laquelle les primitives sont extraites à partir de ce profil.

4.3.1 Identification des maximums et minimums locaux

Le profil de courbure du tracé est obtenu par la méthode des k -cosinus de Rosenfeld et Weszka (1975) ; la technique consiste à identifier les points de courbure maximum en déterminant celle-ci à l'aide d'un nombre variable k de voisins en chaque point du tracé numérique. Dans un premier temps, nous définissons k -vecteurs (voir figure 53) au point p_i :

$$s_{ik} = (x_i - x_{i+k}, y_i - y_{i+k})$$

$$t_{ik} = (x_i - x_{i-k}, y_i - y_{i-k})$$

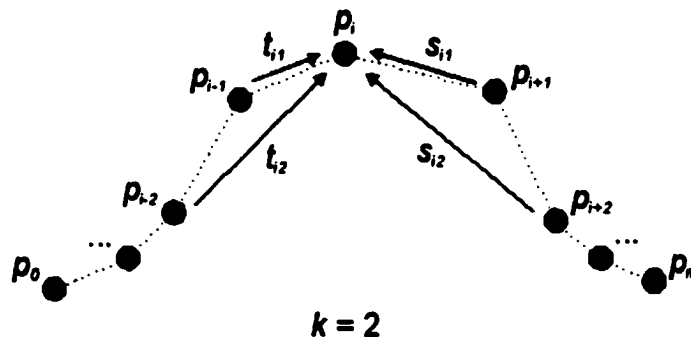


Figure 53 - k -vecteurs pour le calcul de la courbure

et les k -cosinus au point $p_i = (x_i, y_i)$ sont :

$$\cos c_i^{(k)} = \frac{(s_{ik} \cdot t_{ik})}{|s_{ik}| |t_{ik}|}$$

La valeur $\cos c_i^{(k)}$ est le cosinus de l'angle entre s_{ik} et t_{ik} , donc $-1 \leq \cos c_i^{(k)} \leq 1$, où $\cos c_i^{(k)}$ est près de 1 si s_{ik} et t_{ik} font un angle proche de 0° et $\cos c_i^{(k)}$ est près de -1 si l'angle est proche de 180° . En somme, $\cos c_i^{(k)}$ est grand lorsque la courbure est grande et faible lorsque le tracé est plat.

La prochaine étape consiste à trouver la meilleure valeur de k pour chaque point p_i . D'abord, nous calculons $\cos c_i^{(1)}, \cos c_i^{(2)}, \dots, \cos c_i^{(m)}$ où m est choisi selon les caractéristiques de notre domaine (ex. $m = n/10$, où n = nombre de points dans le tracé). Ensuite, nous définissons $h(i)$ de sorte que :

$$\cos c_i^{(m)} < \cos c_i^{(m-1)} < \dots < \cos c_i^{(h(i))}$$

mais en conservant :

$$\cos c_i^{(h(i))} \geq \cos c_i^{(h(i)-1)}$$

À ce stade, le meilleur k est identifié pour chacun des points du tracé.

Cependant, selon les indications de Rosenfeld et Weszka (1975), la méthode perd en efficacité lorsque des changements brusques de direction sont situés trop près les uns des autres (ce qui nuit à la reconnaissance du symbole *résistance*). On peut améliorer la technique en adoucissant les k -cosinus sur une étendue plus vaste selon l'équation suivante :

$$\bar{c}_i^{(k)} = \begin{cases} \frac{2}{k+2} [\cos c_i^{(k)} + \cos c_i^{(k-1)} + \dots + \cos c_i^{(k/2)}] & \text{si } k \text{ est impair} \\ \frac{2}{k+3} [\cos c_i^{(k)} + \cos c_i^{(k-1)} + \dots + \cos c_i^{((k-1)/2)}] & \text{si } k \text{ est pair} \end{cases}$$

Enfin, une fois tous les k-cosinus $\cos c_i^{(h(i))}$ optimaux calculés, nous les analysons afin d'identifier les minimums et maximums locaux. Nous dirons qu'il y a un *maximum local* au point p_i si $\cos c_i^{(h(i))} \geq \cos c_i^{(h(j))}$ pour tout j tel que $|i - j| \leq h(i)/2$. Autrement, nous avancerons que p_i est un minimum local si $\cos c_i^{(h(i))} \leq \cos c_i^{(h(j))}$ pour tout j tel que $|i - j| \leq h(i)/2$. Les figure 54 et figure 55 illustrent deux cas typiques de maximum et de minimum locaux.

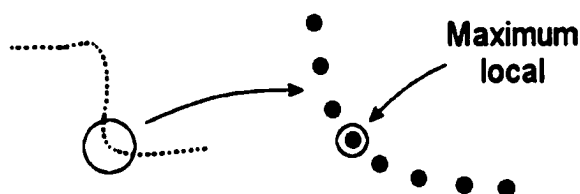


Figure 54 - Identification d'un *maximum local*

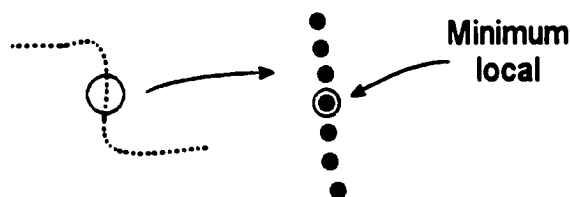


Figure 55 - Identification d'un *minimum local*

4.3.2 Identification des points de transition

Le traitement réalisé à la section précédente permet d'identifier les points du tracé où la courbure est soit maximale, soit minimale. Cependant, comme nous souhaitons démarquer les segments fortement courbés des segments droits, nous introduisons la notion de *points de transition* et de *seuil de transition*. Le *seuil de transition* est la valeur à partir de laquelle un k-cosinus est considéré comme faisant partie d'une courbe ; autrement, le k-cosinus sera considéré comme faisant partie d'une ligne. Considérons le tracé formant un coin de la figure 56 et son profil de courbure, à la figure 57. Le *seuil de transition* est illustré par le trait pointillé de la figure 57.

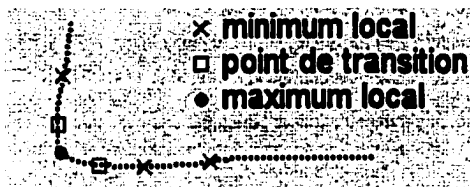


Figure 56 - Tracé en « coin »

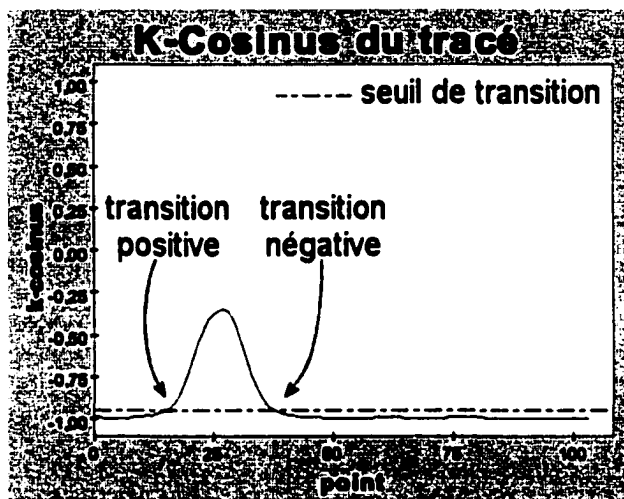


Figure 57 - Profil de courbure du tracé en « coin »

Les points de transition démarqués par des carrés à la figure 56 correspondent, à la figure 57, aux croisements entre le pointillé et le tracé des *k*-cosinus. Ainsi, la partie du tracé qui est au-dessus du pointillé est considérée comme « courbée » et la partie au-dessous comme « droite ». De plus, une transition sera considérée comme *positive* s'il s'agit d'un passage vers une zone courbée et *négative* dans le cas inverse. Les figure 58 et figure 59 illustrent un cas simple : une ligne droite et son profil de courbure, illustrant bien l'absence de changements de direction.

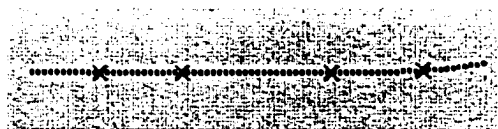


Figure 58 - Tracé droit avec minimums locaux

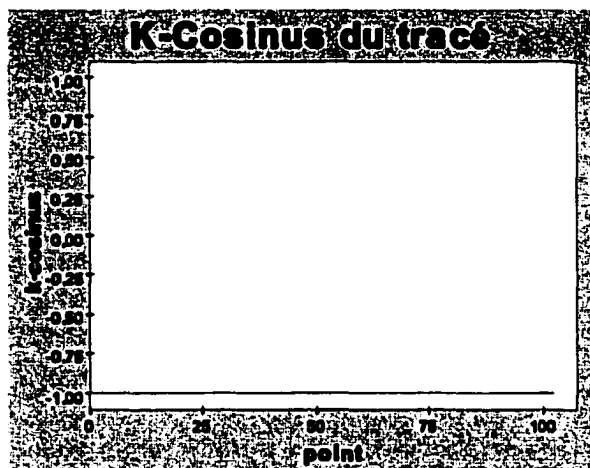


Figure 59 - Profil de courbure du tracé droit

Dans les figures suivantes (figure 60 et figure 61), on peut voir comment un tracé presque entièrement courbé est démarqué. En raison de la courbure continue, la presque totalité du tracé est située au-dessus du *seuil de transition*.



Figure 60 - Tracé en forme de cycloïde

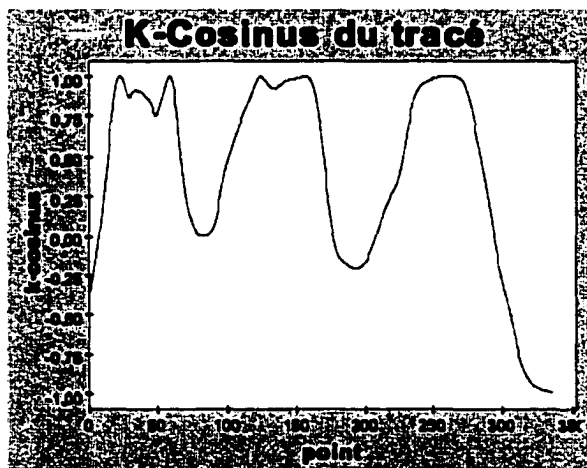


Figure 61 - Profil de courbure du tracé en forme de cycloïde

Dans la section suivante, nous verrons comment les attributs identifiés ici serviront à extraire les primitives base du tracé.

4.4 Extraction des primitives

L'extraction des primitives se fait en deux temps : d'abord, en nous basant sur les points minimums, maximums et de transition, nous identifions dans le tracé les lignes et les arcs « bruts », après quoi nous appliquons un ensemble d'heuristiques (voir 4.4.3) qui ont pour but de filtrer les primitives initiales et en découvrir de nouvelles.

4.4.1 Extraction préliminaire des lignes et arcs

La première étape du processus d'extraction de primitives consiste à analyser les attributs identifiés à la section précédente afin de qualifier d'*arcs* les segments fortement courbés et de *lignes* les segments droits. Pour y arriver, nous appliquons un ensemble de règles sous la forme d'une machine d'états (aussi connue sous le nom d'automate). Ceci signifie que le procédé ne peut être que dans un seul état à la fois, parmi un nombre fini d'états (tableau III). Les transitions entre les états sont possibles grâce aux événements entrants (tableau IV), qui à leur tour entraînent des événements sortants (tableau V). Il est possible qu'un événement sortant parmi plusieurs soit généré par un même événement entrant ; le cas échéant, l'événement sortant choisi sera déterminé en fonction du calcul d'un ou plusieurs prédicats (tableau VI) selon une ou plusieurs variables d'état (tableau VII). Enfin, en plus de générer un événement sortant, l'événement entrant pourra aussi être responsable d'une ou plusieurs actions spécifiques (tableau VIII). Le diagramme de la figure 62 illustre les états du système et les transitions selon l'événement entrant.

Tableau III

États

Nom	Description
INIT	étape de pré-traitement, initialisation des variables d'état
AUCUN	le système est dans un état incertain : il est possible que les points appartiennent autant à une ligne qu'à un arc
DEB_LIGNE	le système a reconnu le début d'une ligne et cherche à en identifier le point de fin
DEB_COURBE	le système a reconnu le début d'un segment de type arc et cherche à en identifier le point de courbure maximum
MIL_COURBE	les points jusqu'ici identifiés forment un arc et le système cherche à en identifier le point de fin
FIN	étape de post-traitement, vérifications finales

Tableau IV

Événements entrants

Nom	Description
MIN_LOC	le point courant est un <i>minimum local</i>
MAX_LOC	le point courant est un <i>maximum local</i>
TR_POS	le point courant est une <i>transition positive</i>
TR_NEG	le point courant est une <i>transition négative</i>

Tableau V
Événements sortants

Nom	Description
AJOUT_LIGNE	une composante de type ligne a été trouvée et doit être ajoutée à la liste temporaire des composantes
AJOUR_ARC	une composante de type arc a été trouvée et doit être ajoutée à la liste temporaire des composantes

Tableau VI
Prédicats

Nom	Description
P0	une ligne a été débutée (Ligne_Deb = VRAI)
P1	un arc a été débuté (Arc_Deb = VRAI)

Tableau VII
Variables d'état

Nom	Description
Ligne_Déb	indique qu'une ligne a été débutée
Arc_Déb	indiquc qu'un arc a été débuté
Pt_Deb	point de « début de caractéristique »
Pt_Mid	point de « milieu de caractéristique »

Tableau VIII
Actions spécifiques

Nom	Description
[1]	Ligne_Déb = VRAI
[2]	Ligne_Déb = FAUX
[3]	Arc_Déb = VRAI
[4]	Arc_Déb = FAUX
[5]	Pt_Deb = point courant
[6]	Pt_Mid = point courant

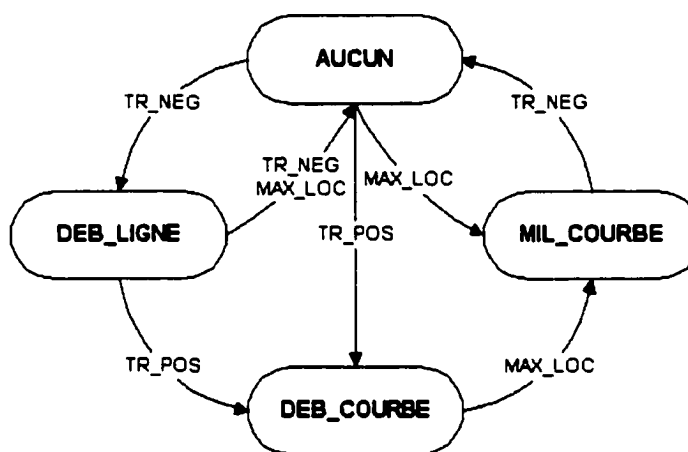


Figure 62 - Diagramme état / transition pour la recherche des primitives

Afin d'alléger le diagramme, nous avons reporté les événements sortants et actions spécifiques à prendre dans les tableau IX et tableau X.

Tableau IX

Table état / événement

État \ Événement	MIN_LOC	MAX_LOC	TR_POS	TR_NEG
<i>INIT</i>	A			
AUCUN	-	B	C	D
DEB_LIGNE	-	E	C	-
DÉB_COURBE	-	E	-	-
MIL_COURBE	-	E	-	D
<i>FIN</i>	F			

voir tableau X

Tableau X

Détails des actions à prendre en fonction de l'état / événement

A	[1], [3]
B	[2], [6]
E	[6]
D	P1 : AJOUT_ARC, [4], [1], [5] Non P1 : [1], [5]
C	P0 : AJOUT_LIGNE, [2], [3], [5] Non P0 : [3], [5]
F	P0 : AJOUT_LIGNE P1 : AJOUT_ARC

4.4.2 Définition des lignes et des arcs

L'opération de segmentation permet de découper le tracé en segments de type ligne et arc, mais elle ne donne aucune information sur la définition comme telle de ces segments. Par exemple, dans la figure 63, selon toute probabilité, une ligne sera

reconnue entre les paires $[p_a, p_b]$ et $[p_c, p_d]$ et un arc entre $[p_b, p_c]$, mais sans plus ; il nous revient de définir la nature de ces primitives.



Figure 63 - Points de début et de fin des segments

4.4.2.1 Définition de la primitive ligne

La primitive de type ligne est simplement définie par ses deux extrémités ; il a été jugé préférable de ne pas altérer la position des points situés aux extrémités car ceux-ci ont plus de chance d'être correctement positionnés par le dessinateur. Nous évaluons la qualité de la ligne en faisant une régression linéaire sur les points et en prenant le carré de l'erreur moyenne comme base pour le calcul du *facteur de certitude* de la ligne. La figure 64 illustre deux cas de facteurs de certitude : en a), un facteur de certitude faible et en b), un facteur de certitude fort. On montre aussi l'emplacement où l'erreur est la plus grande ; cette dernière pourrait servir comme indicateur additionnel de la qualité de la ligne.

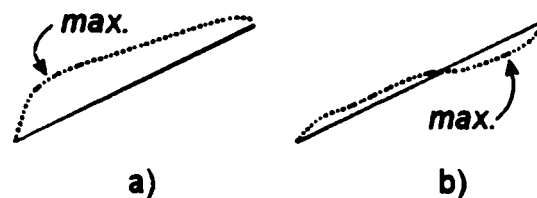


Figure 64 - Exemple de facteurs de certitude : a) erreur grande, donc certitude faible b) erreur faible, donc certitude élevée

4.4.2.2 Définition de la primitive arc

La primitive de type arc est plus complexe à définir que la *ligne*. L'idée de base consiste à rechercher la meilleure estimation de la courbure des points du tracé et ce, sous la forme d'un arc. Diverses techniques furent déjà proposées pour résoudre ce problème, notamment celles de Dosch, Masini et Tombre (2000) et Rosin et West (1989), spécialisées cependant au niveau de la détection d'arcs dans les documents graphiques numérisés (ex. dessins d'architecture).

Nous avons jugé pertinent et suffisant de prendre l'essentiel des techniques existantes en cette matière afin de résoudre notre problématique de façon satisfaisante. La figure 65 illustre le fonctionnement de la méthode ; le but est de déterminer le centre et le rayon de l'arc qui se rapproche le plus des points du segment.

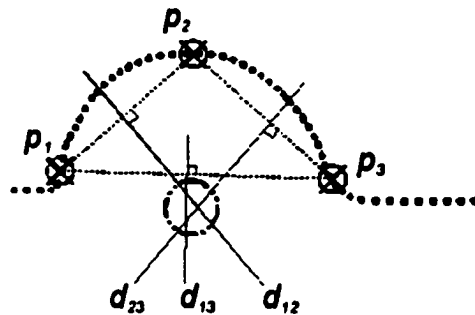


Figure 65 - Technique pour déterminer les caractéristiques d'un arc

Pour y arriver, nous prenons trois points témoins⁶ de la courbe, p_1 , p_2 et p_3 et traçons une droite entre chacun d'eux. Ensuite, perpendiculairement à celles-ci, nous calculons trois nouvelles droites d_{12} , d_{13} et d_{23} . Le *point centre* de l'arc correspondra à l'intersection formée par ces trois droites et le *rayon* à la moyenne des distances entre le centre et les trois points témoins. Enfin, un facteur de certitude sera attribué à l'arc en fonction de la distance moyenne entre chacun des points du segment et l'approximation arc.

⁶ Les points p_1 et p_3 sont respectivement les premiers et derniers points de la section courbée alors que p_2 est le point où la courbure est maximale.

4.4.3 Validation et recherche des autres primitives

Si l'opération de segmentation réalisée à l'étape précédente a permis de démarquer dans le tracé les segments de type ligne des segments d'arc, il nous reste encore à identifier les primitives de type cercle, cycloïde et résistance. Pour y arriver, nous appliquons un ensemble d'heuristiques dont l'objectif est de filtrer les primitives superflues et de consolider celles qui sont en place. Ces heuristiques sont les suivantes :

1. Tronquer extrémités bruitées

- SI un tracé commence ou se termine par une *ligne* dont le voisin est un *arc* ET la *ligne* est substantiellement plus petite que l'*arc* (ex. 4 fois plus petite), ALORS tronquer la *ligne* et allonger l'*arc*.
- SI un tracé commence ou se termine par un *arc* dont le voisin est une *ligne* ET l'*arc* est substantiellement plus petit que la *ligne* (ex. 4 fois plus petite), ALORS tronquer l'*arc* et allonger la *ligne*.

2. Éliminer primitives intercalées bruitées

- SI une *ligne* située entre deux *arcs* est de dimension négligeable, ALORS éliminer la *ligne* et allonger les deux *arcs*.
- SI un *arc* situé entre deux *lignes* est de dimension négligeable, ALORS éliminer l'*arc* et allonger les deux *lignes*.

3. Éliminer ligne superflue entre 2 arcs

- SI une *ligne* est située entre deux *arcs* de dimension similaire ou inférieure à celle des deux *arcs* ET son niveau de confiance est similaire ou inférieur à celui des *arcs* ET les deux *arcs* ont la même direction, ALORS éliminer la *ligne* et allonger les *arcs*.

4. Fusionner arcs

- SI deux *arcs* sont voisins les uns des autres ET ils ont la même direction, ALORS fusionner les deux *arcs*.

5. Rechercher cercles et cycloïdes

- SI un *arc* fait un « angle total » supérieur à 340° et inférieur à 500°, ALORS transformer l'*arc* en un *cercle*.
- SI un *arc* fait un « angle total » supérieur à 500°, ALORS transformer l'*arc* en un *cycloïde*.

6. Fusionner lignes

- SI deux *lignes* sont adjacentes ET elles ont une direction similaire, ALORS fusionner les deux *lignes*.

7. Rechercher résistances

- SI un minimum de 5 *lignes* sont adjacentes ET qu'elles sont toutes de dimension similaire ET que l'angle entre chaque voisin alterne entre une valeur positive et une valeur négative ET ces angles sont dans une étendue de 75° à 170°, ALORS transformer l'ensemble de *lignes* en une *résistance*.

4.4.4 Qualité de la segmentation et raffinement des paramètres

La qualité de la segmentation réalisée à l'étape précédente dépend en grande partie du facteur de voisinage défini à la section 4.3.1. Ce facteur est déterminant du nombre de voisins qui seront considérés lors du calcul de la courbure en un point donné ; souvent choisi arbitrairement en fonction du domaine d'application, nous verrons ici comment nous optimisons le choix de ce paramètre en fonction du tracé qui est traité.

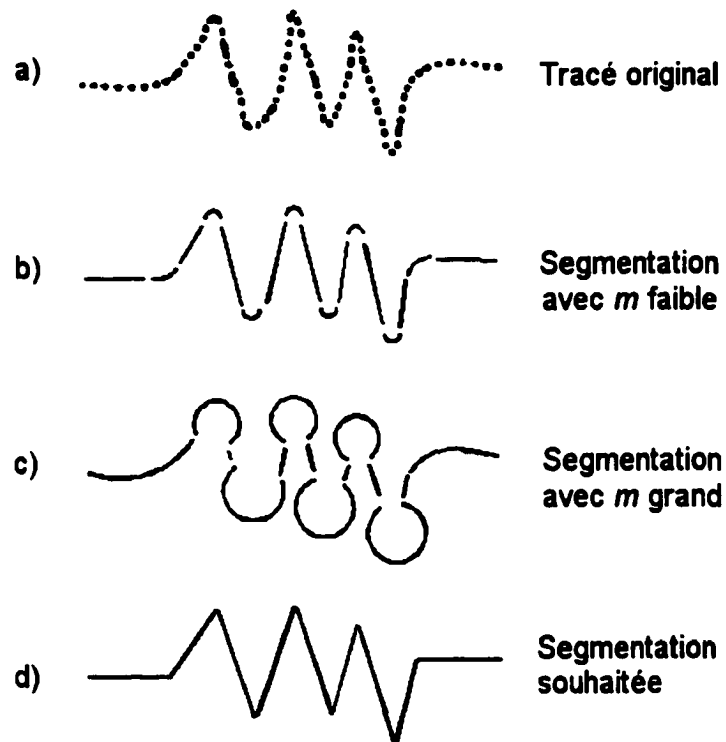


Figure 66 - Segmentation avec facteur de voisinage m variable

La figure 66 illustre la problématique que pose le choix du facteur de voisinage : s'il est trop grand, les changements brusques de courbure ne seront pas détectés ; inversement, s'il est trop faible, un trop grand nombre de points caractéristiques sera détecté par la méthode.

Les images apparaissant en b) et c) sont le résultat de la segmentation brute par k-cosinus, avant que les heuristiques ne soient appliquées. Une inspection visuelle de ces images nous permet d'affirmer que la segmentation avec un m faible est préférable à celle avec m grand, mais comment quantifier cette certitude? Afin de juger de la qualité de la segmentation obtenue suite à l'application d'un facteur m donné, nous prenons la somme des facteurs de confiance de chacune des primitives ainsi segmentées (voir chapitre 13 pour le calcul de cette valeur).

Cependant, il n'existe pas un seul facteur m idéal : un tracé long et composé de grands arcs sera mieux segmenté avec un grand m , alors qu'un m faible conviendra davantage aux tracés faits de petites caractéristiques et de changements d'orientation brusques. Afin de choisir un m optimal pour un tracé donné, nous procéderons de manière itérative en raffinant successivement les paramètres de segmentation m tant qu'il y a amélioration de la qualité de la segmentation.

4.5 Conclusion

Le choix de cette méthode de segmentation est venu suite à de nombreuses tentatives. Nous avons essayé de segmenter le tracé en nous basant d'abord uniquement sur les changements d'angle entre points successifs, mais avons rapidement observé que le bruit présent dans le tracé nous obligeait à choisir une méthode plus évoluée ou de procéder à un adoucissement du tracé ; nous avons choisi de faire les deux, c'est-à-dire d'implanter la méthode des k-cosinus pour une meilleure segmentation, mais aussi de filtrer davantage le tracé source en le transformant en une interpolation spline.

Les heuristiques (section 4.4.3) servant à extraire les primitives complexes (cercle, cycloïde et résistance) furent déterminées de manière expérimentale. Cette approche est extensible car on peut facilement ajouter des règles afin d'identifier de nouvelles primitives, mais elle est aussi quelque peu rigide dans la mesure où elle ne fonctionnera correctement que si toutes les règles menant à l'identification d'une primitive donnée sont énumérées.

CHAPITRE 5

PRÉ-TRAITEMENT

5.1 Introduction

Les systèmes de saisie de données en ligne utilisent des techniques de pré-traitement afin de simplifier la tâche des algorithmes de reconnaissance de forme. Ces techniques sont généralement utilisées pour deux raisons principales : 1) afin de réduire la quantité d'information et 2) pour éliminer les imperfections dans le tracé. Nous verrons dans cette section les techniques utilisées pour remplir ces deux fonctions de pré-traitement.

5.2 Sectionnement des extrémités

Quand le crayon est déposé sur la tablette numérique, il arrive souvent que le tracé soit fortement bruité en raison de mouvements indésirables au niveau du poignet et de l'avant-bras. Nous tenons compte de ces imperfections en tronquant le tracé à ses extrémités (voir zones encadrées « E » dans la figure 67).

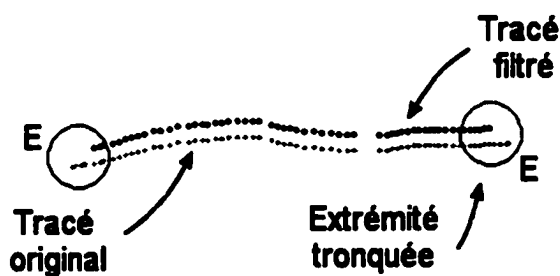


Figure 67 - pré-traitement : tronquer les extrémités

5.3 Enlèvement des points coïncidents

Le nombre de points recueillis à la seconde de la tablette numérique est une valeur fixe (ex., 100 points par seconde). Ainsi, quand l'utilisateur déplace le crayon rapidement,

l'espacement entre les points individuels est grand. Par opposition, quand il s'applique en dessinant lentement, avec soin et précision, les points sont rapprochés, voire coïncidents. C'est dans ce deuxième cas qu'il est nécessaire d'appliquer un traitement au tracé, d'une part afin d'améliorer la performance des traitements ultérieurs et d'autre part afin d'augmenter leur taux de réussite.

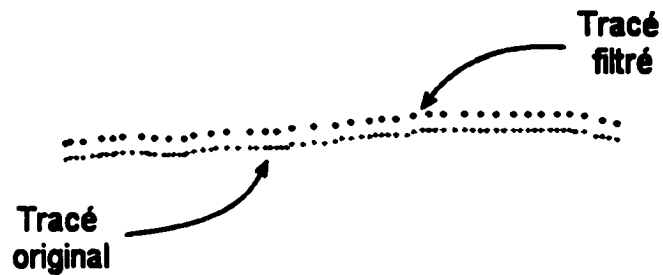


Figure 68 - Pré-traitement : enlèvement des points coïncidents

Ce filtre consiste simplement à enlever de la trame de point originaux ceux dont la distance est inférieure à un seuil fixé à l'avance.

Notons que tout traitement qui modifie le nombre ou la disposition des points originaux a pour effet de perturber l'information dynamique du tracé ; en effet, suite à l'application de ces traitements, il deviendra impossible d'obtenir les profils de vitesse et d'accélération du tracé. Ces derniers pourraient cependant être conservés afin de servir d'intrant additionnel lors de l'étape de segmentation.

5.4 Égalisation

L'opération d'égalisation est en quelque sorte l'inverse de l'enlèvement des points coïncidents ; elle a pour objectif d'uniformiser la distance entre les points. Toute information dynamique est perdue, mais en contrepartie, les anomalies qui peuvent se produire au moment de l'échantillonnage du tracé sont palliées.

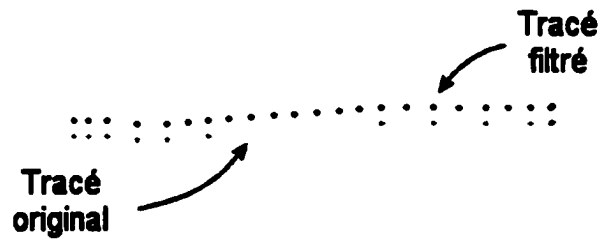


Figure 69 - Pré-traitement : opération d'égalisation

5.5 Adoucissement

Nous adoucissons le tracé en remplaçant chaque point par une moyenne pondérée de ses deux voisins (nos expériences ont démontré que des poids de 0,5 pour le point central et 0,25 pour chacun des points voisins donnent de bons résultats), exception faite des points de début et de fin. Si les points du tracé sont utilisés pour le transformer en une interpolation spline (voir prochaine section), l'utilité de ce filtre est discutable.

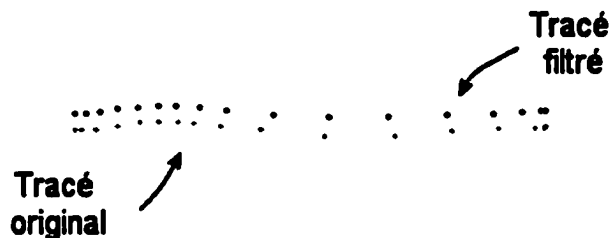


Figure 70 - Pré-traitement : adoucissement

5.6 Interpolation spline

L'opération d'interpolation spline consiste à utiliser les points originaux du tracé comme points de contrôle d'une courbe (du troisième degré) appelée « B-spline cubique » (voir Foley, van Dam, Feiner et Hughes, 1990). La figure 71 illustre une b-spline cubique

ainsi que ses points de contrôle et nœuds interpolés. L'approximation spline peut causer une perte dans la précision du signal, mais ce n'est pas notre cas, étant donnée la faible distance qui sépare les points échantillonnés. Au contraire, la spline a pour nous un grand avantage, celui de générer un signal dont les dérivées, première et seconde, sont continues ; ceci permet de grandement faciliter le procédé de segmentation car celui-ci réagit aux discontinuités dans la courbure du tracé.

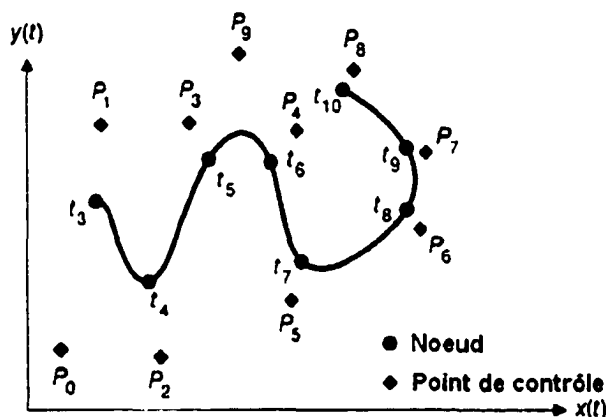


Figure 71 - B-spline cubique et ses points de contrôle

Selon Foley et al. (1990), la spline d'interpolation Catmull-Rom est souvent utilisée pour lisser une série de points provenant d'une tablette numérique ou d'une souris car celle-ci passe par les points de contrôle. Cependant, notre expérimentation nous a montré qu'elle devient instable et qu'elle introduit des anomalies dans le tracé original lorsque les points sont rapprochés et la courbure élevée.

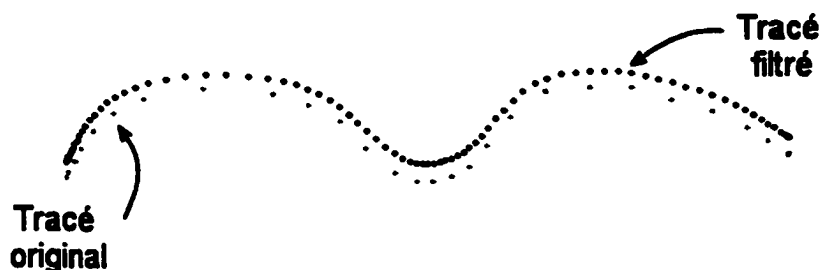


Figure 72 - Pré-traitement : interpolation spline

CHAPITRE 6

EMBELLISSEMENT

6.1 Introduction

Le procédé décrit jusqu'à maintenant a permis de faire la reconnaissance des tracés manuscrits, procédé par lequel on cherche à identifier la nature d'éléments graphiques en les classifiant selon une catégorie définie préalablement. Cette classification reste cependant à un niveau descriptif : le système est en mesure de dire « le tracé en entrée est une *ligne* » ou que « la série d'arcs en entrée forme une *bobine* » (figure 73).

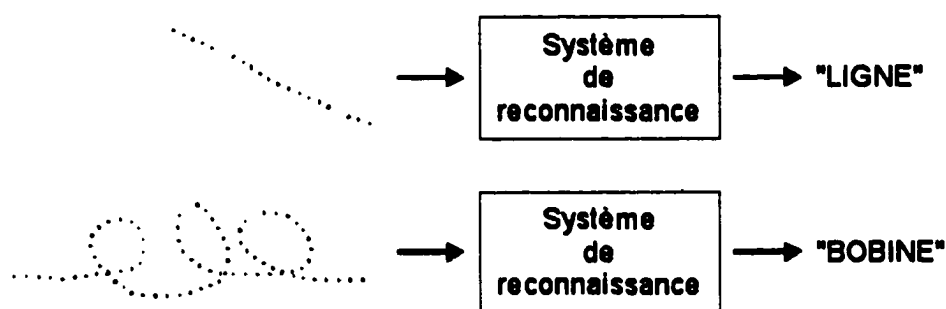


Figure 73 - Procédé de reconnaissance (sans embellissement)

L'embellissement est un procédé qui va une étape plus loin en tentant de remplacer le tracé original par une représentation embellie (figure 74) ; les symboles sont remplacés par leur modèle de référence et les lignes de connexion par une version géométriquement plus exacte.

Il est bon de noter cependant que certains auteurs réfèrent à l'embellissement comme un procédé par lequel des traits dessinés à la main de manière libre sont automatiquement redessinés avec plus d'exactitude ou plus proprement ; ce traitement est généralement réalisé à l'aide de techniques par lesquelles des contraintes sont imposées sur les tracés de manière à les redresser en fonction de critères qui varient selon les méthodes.

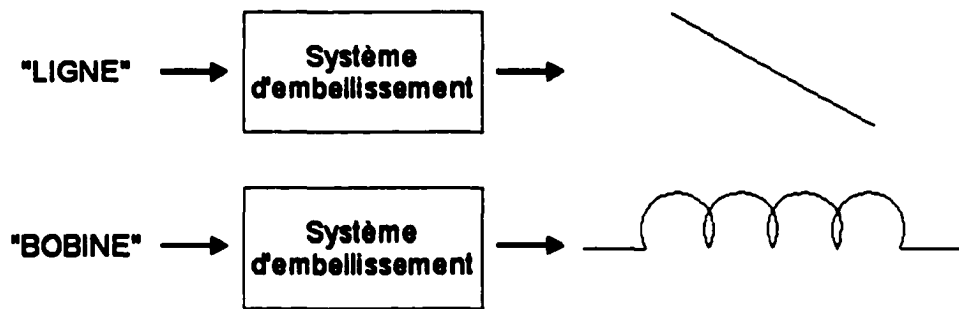


Figure 74 - Procédé d'embellissement

6.2 Problématique

Le processus d'embellissement est fort bien accompli par certains logiciels de reconnaissance optique de caractères (*OCR*). Cependant, pour ce domaine de spécialité, la problématique de reconnaissance est unidimensionnelle car les caractères de l'alphabet occidental sont toujours tracés de la gauche vers la droite. Quand il est question de croquis de schémas, une dimension s'ajoute. En effet, l'orientation des éléments du dessin est variable ; un symbole peut être dessiné de la droite vers la gauche ou du haut vers le bas. Ce facteur a pour effet de rendre l'embellissement plus difficile car s'il est une chose de reconnaître la nature d'un tracé, il en est une autre que de savoir le qualifier entièrement au point d'en connaître suffisamment d'attributs structurels et géométriques (position, orientation et dimension) pour le reproduire de façon complète et juste.

6.3 L'embellissement

L'embellissement, que ce soit pour une ligne ou un symbole, s'appuie donc sur les attributs de la composante. Dans certains cas, seuls les points de début et de fin sont requis pour la reconstitution de l'élément ; c'est le cas de la ligne.

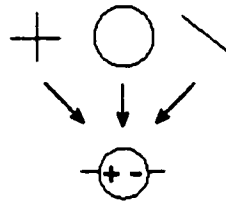


Figure 75 - Structure du symbole source de tension

D'autres cas, comme ceux de la *flèche* et du symbole *source de tension* (figure 75), sont plus complexes ; dans ces deux cas, l'orientation de la composante est également nécessaire si l'on doit la reproduire dans sa version embellie. Dans toutes les situations, la dimension de l'élément est déterminée par ses points de début et de fin ou par sa boîte englobante. L'embellissement est réalisé indépendamment du procédé de reconnaissance et ne sert qu'à des fins esthétiques. La forme des symboles et des diverses composantes intermédiaires de référence est inspirée des normes canadiennes (ACNOR).

6.4 Résultats d'embellissement

Voici quelques résultats d'embellissement :

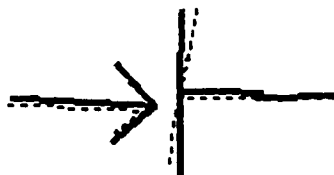


Figure 76 - Embellissement d'une flèche et d'un « bout en T »

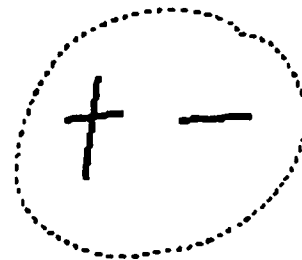


Figure 77 - Embellissement d'un cercle d'un « bout en T »

CHAPITRE 7

RÉSULTATS EXPÉRIMENTAUX ET DISCUSSION

7.1 Introduction

Dans les chapitres précédents, nous avons présenté les divers aspects du système de reconnaissance développé au cours de ce travail. Dans cette section, nous présentons les résultats qui ont été obtenus suite aux tests de validation effectués par un groupe témoin de dix personnes, ainsi que l'analyse de ces tests.

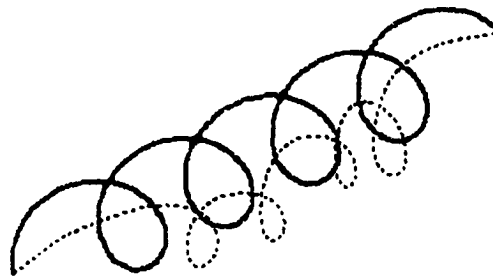


Figure 78 - Reconnaissance du symbole bobine

La figure 78 illustre un symbole reconnu par le système : le trait pointillé indique le tracé source tel que dessiné par l'utilisateur et le tracé plein montre le symbole reconnu et embelli. Comme nous pouvons le voir dans cette image, les symboles furent tracés individuellement, détachés du contexte d'un schéma complet.

7.2 Protocole de validation et résultats de reconnaissance

Dans le but de valider le système de reconnaissance développé, nous avons jugé pertinent de recueillir 200 exemplaires de 4 des symboles pouvant être reconnus ; pour compléter le test, nous avons demandé à 10 personnes de tracer 20 fois chacun les symboles suivants :

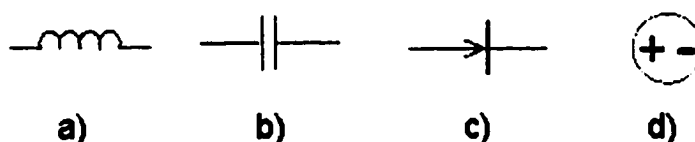


Figure 79 - Symboles de référence. a) bobine, b) condensateur, c) diode, d) source de tension

Une période d'essais libres fut accordée aux usagers afin qu'ils puissent se familiariser avec le système, l'interface (la manipulation du crayon) et les symboles à dessiner. Le nombre de 200 tests à réaliser nous paraît approprié dans la mesure où le système s'appuie avant tout sur des règles structurelles et ne requiert aucun apprentissage. En effet, aucun entraînement n'est requis, si ce n'est celui de l'utilisateur pour s'habituer au système ; seule la nature des interrelations entre les composantes des symboles nous importe. La reconnaissance s'appuie par contre sur un ensemble de techniques qui dépendent bel et bien de valeurs ajustables, mais celles-ci prennent la forme de paramètres, lesquels font l'objet de la prochaine section.

7.2.1 Paramètres du système

L'efficacité de la plupart des modules qui forment le système dépend de paramètres de configuration, notamment les modules de pré-traitement, de segmentation et de reconnaissance. Les paramètres du module de pré-traitement sont déterminés majoritairement en fonction des caractéristiques du matériel utilisé pour la saisie de données (tablette numérique) et du contexte de saisie (dans notre cas, il s'agit d'une tablette pour la saisie et d'un écran d'ordinateur pour l'affichage). Si nous ne pouvons pour l'instant que le présumer, nous croyons que le choix de certains de ces paramètres aurait été différent si nous avions utilisé, par exemple, un tableau numérique avec affichage sur la surface même du tableau.

La segmentation des tracés est fortement dépendante du nombre de points voisins observés pour la détection de la courbure en un point donné. Une valeur initiale doit être spécifiée, et de là, il est possible de raffiner la méthode pour arriver à une segmentation optimisée selon le tracé. Cependant, pour des questions d'efficacité, il


n'est pas concevable en un point donné d'observer tous les points du tracé ; une valeur de départ doit être fixée et en définitive, celle-ci sera déterminante de la plus grande courbe qui pourra être dessinée.

Les paramètres spécifiques à la reconnaissance servent à ajuster la flexibilité du mécanisme par lequel les relations entre sous-composantes sont évaluées. Plus les paramètres sont permissifs, plus les formes seront reconnues facilement, mais en contrepartie, plus il y aura de chances qu'une forme soit mal reconnue. Inversement, si les paramètres sont rigides, la reconnaissance sera davantage stricte, mais aussi plus juste. Nous considérons qu'il revient à l'utilisateur de fixer ces paramètres selon son niveau d'expérience.

7.2.2 Reconnaissance de la bobine

Tableau XI

Taux de reconnaissance du symbole *bobine* selon l'utilisateur

		
<i>Usager</i>	Taux d'erreur (symbole non-reconnu)	Taux de reconnaissance (symbole bobine reconnu)
1	0	20
2	3	17
3	0	20
4	1	19
5	0	20
6	2	18
7	5	15
8	8	12
9	1	11
10	0	20
11	0	20
Somme	20	192
Taux moyen	9%	91%

Dans la plupart des cas, la bobine est bien reconnue. Sur 212 tentatives, elle est reconnue 192 fois (91%) avec un taux de confiance de 84% (défini selon la similitude du tracé à une forme de référence).



Figure 80 - Symbole bobine erroné

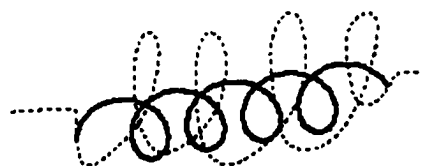


Figure 81 - Symbole bobine reconnu

La figure 80 présente un symbole bobine qui n'est pas reconnu par le moteur de reconnaissance ; dans ce cas, l'utilisateur a tracé des vagues plutôt que des boucles et notre schéma de reconnaissance structurelle ne tient pas compte de cette particularité. La figure 81 montre un symbole où au contraire les boucles, bien que légèrement exagérées, sont bien tracées.



Figure 82 - Symbole bobine erroné

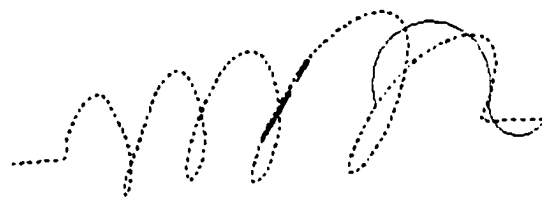


Figure 83 - Symbole bobine erroné

Dans les figures 82 et 83, nous voyons qu'une déficience dans la segmentation est la cause d'une mauvaise reconnaissance ; des heuristiques additionnelles auraient pu contribuer à l'améliorer. Par exemple, dans ce cas, une heuristique aurait facilement pu corriger le problème des deux bobines adjacentes.

7.2.3 Reconnaissance du condensateur

Tableau XII

Taux de reconnaissance du symbole *condensateur* selon l'usager

Usager	Taux d'erreur			Taux de reconnaissance
	(aucune composante reconnue)	(une composante reconnue)	(deux composantes reconnues)	(symbole condensateur reconnu)
1	6	6	6	2
2	6	4	8	2
3	3	4	8	5
4	1	8	7	4
5	1	3	12	4
6	4	2	5	9
7	4	12	4	0
8	7	5	7	1
9	9	5	3	3
10	1	3	13	3
11	11	4	4	1
Somme	53	56	77	34
Taux moyen	24,1%	25,5%	35,0%	15,5%

* les composantes sont les *primitives* ou *composantes spécifiques* dont est composé un symbole.

Dans 34 cas seulement sur 220, le symbole condensateur a été reconnu avec un taux de confiance supérieur à 70%, alors que dans 77 cas, le taux de confiance était inférieur à 70%. Dans 56 cas, une seule des 2 composantes du condensateur fut reconnue et enfin, dans 53 cas, soit près de 25%, aucune composante ne fut reconnue correctement.

Les images de la figure 84 illustrent toutes des situations où le symbole condensateur n'a pas été correctement reconnu. En a), les traits sont trop éloignés les uns des autres de sorte que le moteur ne peut les associer ; en b), la relation de perpendicularité qui doit exister entre les composantes d'une « connexion en T » n'est pas suffisamment élevée. Le problème en c) est causé par l'angle d'inclinaison des composantes ; comme la boîte englobante des entités sert à déterminer leur voisinage

significatif, une composante qui normalement serait trop éloignée pour en faire partie pourra à tort y être incluse. Enfin, en d), le symbole condensateur n'a pu être reconnu en raison de la trop grande échelle à laquelle il fut dessiné ; une solution à ce problème serait d'ajuster les paramètres de reconnaissance afin de tenir compte du contexte dans lequel se trouvent les éléments du dessin.

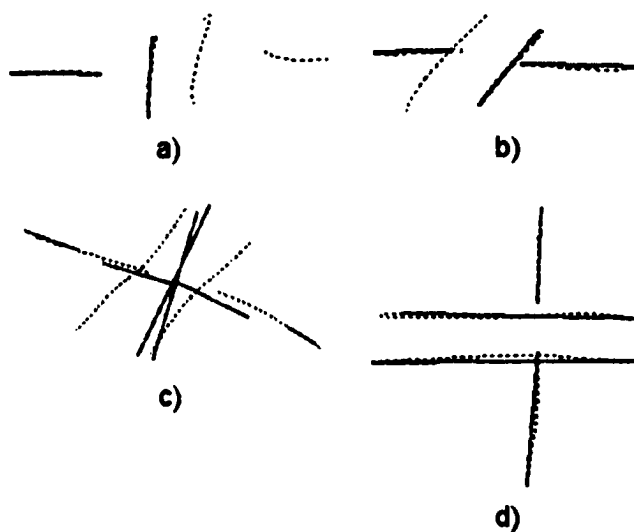


Figure 84 - Symboles condensateur erronés

7.2.4 Reconnaissance de la diode

Tableau XIII

Taux de reconnaissance du symbole *diode* selon l'usager

Usager	Taux d'erreur			Taux de reconnaissance
	(aucune composante reconnue)	(une composante reconnue)	(deux composantes reconnues)	(symbole diode reconnu)
1	2	2	1	15
2	7	4	1	8
3	5	4	0	11
4	3	5	0	12
5	0	4	6	10
6	0	2	0	18
7	7	6	1	6
8	4	8	0	8
9	6	6	0	8
10	1	2	2	15
11	0	7	0	13
Somme	35	50	11	124
Taux moyen	15,9%	22,7%	5,0%	56,4%

* les composantes sont les *primitives* ou *composantes spécifiques* dont est composé un symbole.

La diode est reconnue avec un taux de succès beaucoup plus élevé que celui du condensateur, malgré le fait que ces deux symboles soient très similaires. Nous remarquons que de façon générale, si les deux composantes qui constituent la diode sont reconnues avec succès, la diode le sera aussi (le symbole n'est pas reconnu dans seulement 5% des cas). Dans le cas du condensateur, les choses sont différentes : les composantes constituantes sont souvent présentes, mais le symbole n'est pas reconnu. Ceci s'explique probablement par des facteurs de reconnaissance plus stricts ou moins bien ajustés pour l'identification du condensateur.

La figure 85 illustre une situation où, pour le symbole diode, une seule des deux sous-composante a été reconnue correctement ; dans ce cas les facteurs de reconnaissances n'étaient pas adéquats et doivent vraisemblablement être revus.

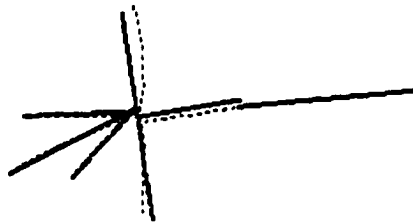


Figure 85 - Composante flèche non reconnue

Le tracé de la figure 86 illustre un cas où la flèche a été reconnue dans la direction inverse de celle souhaitée par l'utilisateur. Comme ce fut le cas plus tôt, cette erreur est vraisemblablement causée par un mauvais calcul du facteur de confiance ; un ajustement des règles de reconnaissance serait souhaitable.

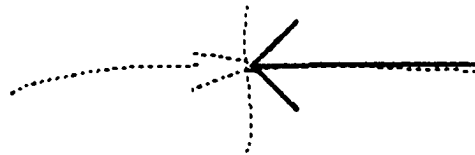


Figure 86 - Composante flèche mal reconnue

La figure 87 illustre l'un des 11 cas où les deux composantes du symbole furent reconnues correctement, mais où le symbole n'a pu être identifié avec un seuil de confiance suffisamment élevé pour être considéré comme juste.



Figure 87 - Symbole diode non-reconnu

7.2.5 Reconnaissance de la source de tension

Tableau XIV

Taux de reconnaissance du symbole *source de tension* selon l'usager

+ -				
Usager	Taux d'erreur			Taux de reconnaissance
	(aucune composante reconnue)	(une composante reconnue (+ une ligne))	(deux composantes reconnues (+ une ligne))	(symbole source de tension reconnu)
1	2	13	1	4
2	4	8	0	8
3	1	12	1	6
4	3	8	1	8
5	1	11	1	7
6	2	10	0	8
7	7	9	0	4
8	5	6	1	8
9	8	7	0	5
10	0	6	0	14
11	1	6	1	12
Somme	34	96	6	84
Taux moyen	15,5%	43,6%	2,7%	38,2%

* les composantes sont les *primitives* ou *composantes spécifiques* dont est composé un symbole.

Sur 220 tentatives, 84 sources de tension furent identifiées correctement. Le plus grand problème avec la reconnaissance de ce symbole est situé au niveau de la composante cercle : d'une part, il n'est pas facile de les tracer avec la tablette, et d'autre part, les techniques de segmentation et de reconnaissance sont plutôt sensibles aux formes courbées (voir la section 7.3 pour plus de détails à ce sujet). Les statistiques indiquent que dans 96 cas, 1 seule composante fut reconnue ; pour certains usagers, le cercle était difficile à tracer alors que pour d'autres, le *signe plus* posait un problème.

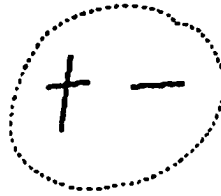


Figure 88 - Symbole *source de tension* non-reconnu

Dans quelques rares cas, toutes les sous-composantes furent reconnues sans que le symbole le soit pour autant : ceci illustre à notre avis un bon ajustement des paramètres de segmentation.

7.3 Discussion

Nous croyons qu'il y a un important problème au niveau de l'utilisation de la tablette numérique. Dans un premier temps, le fait d'avoir à tracer les symboles sur la tablette tout en gardant les yeux sur l'écran est problématique : l'utilisateur s'y fait, mais seulement après un temps d'apprentissage, de toute évidence plus long que ce que nous avons prévu, malgré la période d'essais libres qui lui a été accordée. Dans un second temps, contrairement à la souris dont la position est obtenue de manière incrémentale, la tablette fonctionne selon des positions absolues : la surface de la tablette correspond à celle de l'écran, de sorte que le crayon doit être physiquement déplacé sur celle-ci là où à l'écran l'utilisateur veut que le symbole apparaisse. Il en résulte un inconfort qui rend l'utilisation de la tablette plutôt difficile et peu naturelle.

Si notre technique permet à l'utilisateur de tracer les symboles pratiquement sans aucune contrainte, il est toutefois difficile de faire la reconnaissance dans toutes les situations. Certains utilisateurs sont ainsi particularisés par leur façon propre de tracer les symboles, c'est-à-dire selon un mouvement qui leur vient naturellement, mais qui pour le succès de la méthode doit être assujéti aux contraintes minimales imposées par le système. La figure 89 illustre cette problématique.

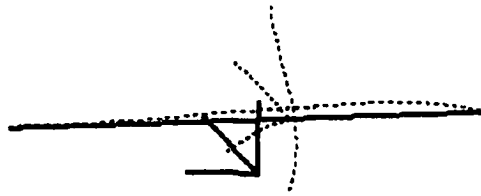


Figure 89 - Symbole diode erroné

Dans cette image, l'usager a tracé une seule ligne horizontale, y a ajouté la tête de la flèche et enfin le trait vertical ; dans cette situation, le procédé de reconnaissance ne peut identifier les sous-composantes car les éléments structurels doivent être distincts les uns des autres (la ligne aurait dû être dessinée en deux parties, soit scindée à l'intersection de chaque sous-composante).

Dans la figure 90, nous voyons comment sur la gauche, l'usager a tracé la sous-composante « connexion en T » en laissant traîner le crayon sur la tablette tout au long du mouvement. Le problème posé par la reconnaissance d'un tel tracé est similaire à celui illustré à la figure 89.



Figure 90 - Symbole condensateur erroné

Dans ce cas cependant, si le procédé de segmentation agit efficacement, le tracé sera segmenté en trois parties que forment le trait horizontal, le trait court vertical et le trait vertical long qui croise celui qui est à l'horizontal, mais la reconnaissance ne pourra se faire en raison d'une règle qui impose une cardinalité nulle (aucun voisin) au tracé vertical. Les erreurs résultant de ces problèmes pourraient potentiellement être réglées à l'aide d'heuristiques, mais le coût en termes de temps de calcul serait d'autant plus élevé. De manière plus générale, cette difficulté est causée par le problème des tracés dédoublés : si un usager trace deux fois, au même endroit, une ligne identique, le système n'y porte aucune attention spéciale. Ce problème, tout comme celui de la

figure 89, peut être solutionné par l'ajout d'une couche de gestion des tracés bruts sous la forme d'un bitmap. Un tel mécanisme permettrait de scinder les tracés là où de nouveaux croisements seraient détectés et de ce fait faciliter la tâche au moteur de reconnaissance.

Nous remarquons que les usagers ont de la difficulté à tracer les cercles (figure 91) ; de manière générale, tel que nous le notons à la section sur la reconnaissance de la source tension, notre système est en effet sensible à la détection des arcs. Ceci s'explique par la liberté qui est accordée à l'utilisateur dans le choix de l'échelle à laquelle les formes sont dessinées. S'il est vrai que nous espérons de meilleurs résultats du module de détection des primitives, nous croyons toutefois que son efficacité pourrait être améliorée en généralisant davantage les règles de détection ou encore en ajoutant à la méthode un module auxiliaire qui permette un classement préliminaire des tracés selon un ensemble de critères donnés (à la manière de Fonseca et Jorge (2000b)).

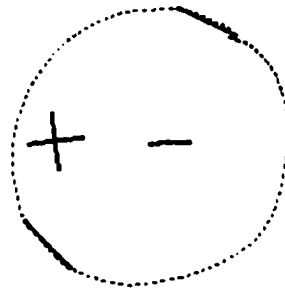


Figure 91 - Primitive cercle difficile à tracer

Tel que décrit au chapitre 3, le facteur de confiance total d'une composante est obtenu d'après la moyenne des facteurs de confiance de chacune des relations testées pour cette composante. Si l'on observe la flèche de la figure 92, nous constatons que cette façon de déterminer le facteur de confiance total n'est pas idéale.

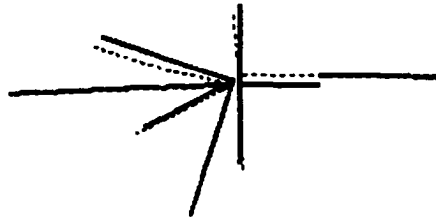


Figure 92 - Symbole diode erroné en raison d'un mauvais calcul de relations

En effet, dans cette image, un symbole diode a bel et bien été reconnu, mais la flèche n'a pas été identifiée correctement (celle illustrée a un faible facteur de confiance de 75%). Le moteur de reconnaissance a vu qu'il y avait trois lignes pouvant former une flèche, mais si plusieurs relations avaient un taux de confiance élevé, quelques-unes étaient très faibles. Cependant, comme la moyenne des facteurs était suffisamment forte, la composante spécifique flèche a été reconnue avec un assez grand succès pour juger la composante valide.

Notre utilisation de la boîte englobante (rectangulaire) pour déterminer l'espace occupé par les composantes est adéquate lorsque celles-ci sont à l'horizontale ou la verticale, mais, tel que décrit à la section 7.2.3, son efficacité se détériore lorsque les composantes sont orientées autrement. La partie c) de la figure 84) illustre ce problème. Pour solutionner ce problème, nous avons également considéré la boîte convexe, mais les calculs requis affectaient considérablement le temps de réponse du système.

Enfin, le problème de l'échantillonnage demeure une limite physique qui ne peut être surmontée et à laquelle l'utilisateur doit être préparé. Il doit s'assurer de ne pas dessiner les symboles avec une vitesse démesurée, sans quoi le nombre de points échantillonnés ne sera pas suffisant pour identifier la forme.

7.4 Conclusion sur la reconnaissance des symboles

Les résultats obtenus suite aux tests sur 200 symboles dessinés par 10 personnes ont été présentés. Les statistiques montrent que les symboles furent reconnus avec un

taux de confiance supérieur à 0,7 (70%) dans seulement 50% des cas. Cependant, ce chiffre se rapproche du seuil des 60% si l'on considère les cas où tous les éléments requis pour un symbole étaient correctement reconnus, sans pour autant que le symbole lui-même le soit (c'est le cas de la figure 87).

Ces tests ont permis de mettre en évidence les faiblesses et lacunes du système. Nous constatons que dans certains cas, l'identification des primitives dans un tracé source n'est pas correcte : ceci est causé par un mauvais ajustement des paramètres de segmentation. Par ailleurs, l'incapacité du système à gérer les tracés dédoublés pose un problème si l'utilisateur ne prend pas soin de lever le crayon pour bien séparer les composantes d'un même symbole.

Étant donné l'inflexibilité dans la segmentation des tracés source, le moteur de reconnaissance ne sait encore proposer qu'un seul résultat. De plus, nous remarquons qu'un mauvais ajustement des paramètres servant au calcul des relations a souvent été la cause de l'échec du moteur de reconnaissance. Inversement, de mauvaises conclusions furent tirées par ce dernier en raison du calcul erroné du facteur de confiance.

Par ailleurs, l'impossibilité de prendre en compte le contexte d'un symbole pour assister sa reconnaissance est une lacune importante. La figure 88 montre une situation où, sur les trois composantes requises pour compléter le symbole *source de tension*, un seul, le *signe plus*, ne fut pas reconnu correctement.

CONCLUSION

Principales contributions

Nous avons présenté dans ce travail un système de reconnaissance en ligne de croquis de symboles électriques. Nous avons d'abord passé en revue les méthodes existantes dans le domaine et avons constaté que notre problématique se trouvait à la croisée de plusieurs avenues de recherche en reconnaissance de graphiques, certaines nouvelles, d'autres moins. À la suite de l'analyse de diverses techniques en cette matière, nous avons élaboré une méthode de reconnaissance hiérarchique flexible qui s'appuie sur des critères structurels et topologiques.

Dans la partie principale de ce travail, nous avons décrit en détails la technique élaborée et avons vu dans ces sections comment notre méthode tente de résoudre la problématique posée antérieurement. Dans un premier temps, nous avons illustré le fonctionnement du moteur de reconnaissance et avons montré comment il était possible, par une représentation structurelle des relations topologiques entre les composantes, d'aborder la complexité de la reconnaissance de symboles spécialisés tout en laissant à l'usager la possibilité de dessiner pratiquement sans contraintes. En outre, nous avons expliqué de quelle manière le « voisinage significatif » des éléments graphiques était pris en compte lors de l'étape de reconnaissance et avons subséquemment défini comment, par l'utilisation de facteurs de confiance, nous gérons l'incertitude dans le but de rendre la méthode flexible. Finalement, nous avons expliqué comment il a été possible, grâce à une technique de reconnaissance hiérarchique agissant de concert avec un module de notification permettant le recyclage des composantes, de tenir compte de l'interactivité dans l'application en reconnaissant les formes au fur et à mesure qu'elles étaient dessinées.

Dans un second temps, nous avons décrit brièvement comment nous procédions pour extraire les primitives de base et de quelle manière nous embellissons les tracés en nous basant sur les informations structurelles et positionnelles obtenues du moteur de reconnaissance. Finalement, l'analyse des résultats obtenus suite aux tests de

validation nous a permis de constater l'efficacité de la méthode et de mettre à jour les faiblesses et lacunes du système. Dans la prochaine section, nous décrivons les améliorations qui à notre avis devraient être apportées au système afin de palier ces lacunes.

Perspectives

Nous proposons les points suivants comme suites à donner à ce travail.

- Terminer l'écriture des heuristiques de détection pour toutes les composantes du système ;
- Modifier la technique de calcul du facteur de confiance en faisant la pondération de la contribution de chaque relation topologique et en accordant plus d'importance aux relations qui ont un trop faible taux de confiance ;
- Améliorer la détection des arcs en choisissant une méthode qui tient compte du contexte (voir Saund, 1993) ou qui fait un recoupement de plusieurs techniques (voir Dosch et al., 2000) ;
- Incorporer un module de reconnaissance par hypothèses statistiques selon une approche connexe aux travaux de Côté et al. (1998). Le but d'un tel module serait de produire un résultat de reconnaissance par un mécanisme d'accumulation d'hypothèses, dans l'espace bidimensionnel, en s'appuyant fortement sur la structure des composantes ;
- Ajouter un module de traitements en différé pour la gestion des tracés discontinus. Le système de reconnaissance que nous avons développé ne prévoit aucun mécanisme capable de gérer les tracés discontinus ; nous savons découper un tracé en ses multiples parties, mais pas l'inverse. Afin de pallier cette lacune, il serait souhaitable d'incorporer au système un module de post-traitement dont les

tâches permettraient de fusionner les lignes droites discontinues et corriger les connexions en T et en L ;

- Incorporer un module de réévaluation du pré-traitement par rétroaction contextuelle pour les symboles qui sont reconnus avec un trop faible degré de confiance. C'est par un tel processus qu'il sera possible d'évaluer de multiples hypothèses structurelles pour un même tracé ;
- Gérer le problème des tracés dédoublés par le maintien permanent d'une couche de représentation du signal brut sous la forme d'un bitmap. Les tracés seraient d'abord inscrits dans cet espace mémoire de même dimension que celle de la surface de dessin. Les dédoublements pourraient être identifiés lorsqu'un nouveau tracé chercherait à occuper une zone du bitmap déjà occupée (à l'aide d'un seuil définissant la distance de contact). Un tel mécanisme permettrait à la fois de scinder les tracés là où de nouveaux croisements seraient détectés, facilitant ainsi la tâche au moteur de reconnaissance (à ce sujet, voir Saund et Moran, 1995) ;
- Considérer l'ajout d'un module auxiliaire de détection statistique des primitives comme le font Fonseca & Jorge (2000b). Ce module pourrait agir rapidement sur les tracés complets selon des critères statistiques ; un classement serait possible sans qu'aucune segmentation ne soit nécessaire. Certaines primitives, tels les cercles, pourraient ainsi être facilement détectés.

Il est clair qu'il reste encore beaucoup de travail à faire pour que ces systèmes progressent. Si la nature interactive du système nous permet de faire une première segmentation parmi les éléments graphiques au moment où le crayon est levé, nous ne tenons cependant pas compte de l'information dynamique des tracés individuels pour assister la segmentation ou encore de l'ordre dans lequel ils sont tracés pour améliorer la reconnaissance. Par ailleurs, tel que nous l'avons mentionné précédemment, la prise en compte des informations contextuelles, la correction de la segmentation par rétroaction et l'action d'un moteur de reconnaissance statistique sont des ajouts qui contribueraient à améliorer largement la méthode. Cependant, en raison de la grande

capacité de calcul requise, ces améliorations seraient apportées au coût d'un prolongement du temps de réponse du système, lequel doit être minimal dans ce type d'application.

En somme, nous sommes convaincus que les systèmes intelligents d'aide à la création et de reconnaissance en ligne de formes spécialisées faites à la main occuperont une place importante dans les systèmes d'édition graphique futurs. Ce travail marque les premiers pas réalisés dans cette direction.

ANNEXE 1

Heuristiques de reconnaissance

Cette annexe est consacrée à une description détaillée des heuristiques et paramètres à l'aide desquels la reconnaissance des primitives, composantes spécifiques et symboles est rendue possible.

1.1 Définitions

Les relations topologiques utilisées par les heuristiques présentées dans cette section sont les suivantes (voir chapitre 3, p. 46 pour de plus amples détails) :

Définitions	Formules
Indice de connexion : valeur booléenne qui détermine si deux composantes partagent un point sur la même primitive de support	Valeur = CONNECTÉ, DISCONTINU
Distance de connexion : mesure de la position du centre de la composante voisine par rapport à la ligne directrice de la référence	$\text{Distance de connexion} = \frac{D}{L}$ <p>L : demi-longueur de CV[*], D : distance entre le point où CR^{**} croise CV et le centre de CV.</p>
Degré de contact : mesure de la proximité de deux composantes	$\text{Degré de contact} = 1 - \frac{D3}{D2}$ <p>$D2$: boîte englobante de CV, $D3$: boîte englobante formée par l'intersection de CR et CV.</p>
Degré de similitude : exprime la différence de « dimension » entre deux composantes	$\text{degré de similitude} = \frac{\min(D2, D3)}{\max(D2, D3)}$ <p>$D2$: boîte englobante de CV, $D3$: boîte englobante de CR.</p>

* CV est la composante voisine
 ** CV est la composante voisine

<p>Degré de concentricité : permet de mesurer à quel point deux composantes partagent le même point central</p>	$\text{degré de concentricité} = 1 - \frac{D}{\max(D1, D2)}$ <p>D : distance euclidienne entre le centre de CR et le centre de CV, $D1$: boîte englobante de CR, $D2$: boîte englobante de CV.</p>
<p>Degré de parallélisme : mesure obtenue par une comparaison de l'orientation de deux composantes</p>	$\text{degré de parallélisme} = \cos(\alpha)$ <p>α : angle formé par les directrices de CR et CV.</p>
<p>Degré de colinéarité : permet de savoir à quel degré deux composantes sont alignées selon le même axe</p>	$\text{degré de colinéarité} = \frac{\text{degré de parallélisme} + (1 - \text{distance de connexion})}{2}$
<p>Degré de codirection : permet de savoir à quel degré deux composantes sont dirigées dans le même sens</p>	$\text{degré de codirection} = v1 \cdot v2$ <p>$v1$: vecteur d'orientation de CR, $v2$: vecteur d'orientation de CV.</p>
<p>Degré de perpendicularité : mesure obtenue par une comparaison de l'orientation de deux composantes</p>	$\text{degré de perpendicularité} = \sin(\alpha)$ <p>α : angle formé par les directrices de CR et CV.</p>
<p>Degré de superposition : indice de la portion de la composante de référence recouverte par son voisin</p>	$\text{degré de superposition} = \frac{A3}{A1}$ <p>$A1$: aire de CR, $A3$: aire de l'intersection formée par CR et CV.</p>

Facteur gauche / droite : permet de savoir de quel côté est situé le voisin par rapport à la référence	Valeur = GAUCHE, DROITE, AUCUN
Facteur avant / arrière : permet de savoir si le voisin est situé surtout vers la partie avant ou la partie arrière de la référence	Valeur = AVANT, ARRIÈRE, AUCUN
Facteur intérieur / extérieur : relation d'inclusion entre le voisin et la référence	Valeur = INTÉRIEUR, EXTÉRIEUR, AUCUN
Ordre temporel : permet de connaître la séquence dans laquelle les composantes furent dessinées	Valeur = AVANT, APRÈS, AUCUN
Distance : quantifie l'éloignement du voisin par rapport au centre de la référence	$\text{Distance} = \frac{D}{L}$ <p><i>D</i> : distance projetée du centre de CV sur la directrice de CR, <i>L</i> : demi-longueur de CR.</p>

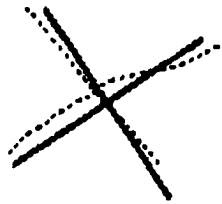
Rappelons que ces relations expriment toutes la nature du lien topologique existant entre une composante graphique de référence et une composante voisine.

Certaines autres valeurs sont utilisées par les heuristiques, notamment les suivantes :

- **Angle total de l'arc** : il s'agit de l'angle total balayé formé par la tangente à un arc, du début jusqu'à la fin de ce dernier (par exemple, un cercle aura un angle total de 360° et un cycloïde composé de trois boucles un angle de 1080°) ;
- **Ratio de la boîte englobante** : il s'agit du ratio de la hauteur sur la largeur de la boîte englobante.

Facteur de confiance

Le facteur de confiance d'une composante donnée est calculé de sorte à tenir compte, à l'aide d'une moyenne arithmétique simple, de chacune des relations topologiques qui entrent dans sa composition. Prenons l'exemple du « *signe plus* » suivant (le pointillé illustre le tracé original et le tracé plein le symbole embelli) :



Au moment de la reconnaissance, lorsque les conditions générales structurelles d'admission sont vérifiées (dans ce cas, lorsque 2 lignes sont analysées), la valeur des relations topologiques suivantes est calculée entre les deux lignes :

concentricité :	0.94
perpendicularité :	0.95
similitude :	0.90
superposition :	1.00

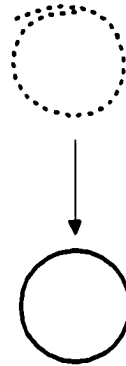
Le facteur de confiance est ensuite obtenu d'après la moyenne arithmétique de ces relations :

$$\begin{aligned} \text{facteur de confiance} &= (0.94 + 0.95 + 0.90 + 1.00) / 4 \\ &= 0.95 \end{aligned}$$

1.2 Reconnaissance des primitives

Les primitives simples, c'est-à-dire les lignes et les arcs, sont obtenues à la suite de l'étape de segmentation par la méthode des k-cosinus. C'est à partir de ces primitives que nous tentons d'identifier les primitives complexes décrites dans cette section.

1.2.1 Cercle



Heuristique de reconnaissance

SI primitive est de type ARC

SI ($angle_total_arc > ANGLE_MIN$)

ET ($angle_total_arc < ANGLE_MAX$)

ET ($ratio_boîte_englobante > RATIO_MIN$)

ET ($ratio_boîte_englobante < RATIO_MAX$)

ALORS calculer facteur de confiance pour un CERCLE

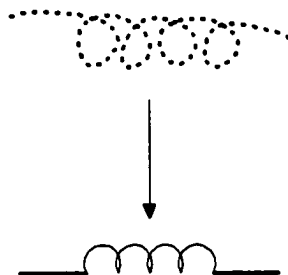
FIN_SI

FIN_SI

Paramètres

Paramètre	Valeur
ANGLE_MIN	340°
ANGLE_MAX	500°
RATIO_MIN	0,85
RATIO_MAX	1 / 0,85

1.2.2 Cycloïde (bobine)



Heuristique de reconnaissance

SI primitive est de type ARC

SI ($angle_total_arc > ANGLE_MIN$)

ET ($angle_total_arc < ANGLE_MAX$)

ET ($ratio_boîte_englobante > RATIO_MIN$)

ET ($ratio_boîte_englobante < RATIO_MAX$)

ALORS calculer facteur de confiance pour un CYCLOÏDE

FIN_SI

FIN_SI

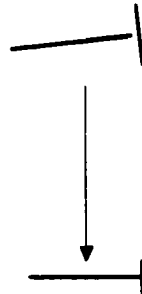
Paramètres

Paramètre	Valeur
ANGLE_MIN	500°
ANGLE_MAX	1440°
RATIO_MIN	0,33
RATIO_MAX	1 / 0,33

1.3 Reconnaissance des composantes spécifiques

Les composantes spécifiques décrites dans cette section sont obtenues à partir des primitives de type ligne et arc.

1.3.1 Bout en T



Heuristique de reconnaissance

(note : *primitive_1* est la référence et *primitive_2* est la partie qui forme l'extrémité perpendiculaire à la référence)

SI *primitive_1* est de type LIGNE

ET *primitive_2* est de type LIGNE

SI *cardinalité* de *primitive_1* est NULLE

ET *cardinalité* de *primitive_2* est NULLE

ALORS SI (*concentricité* < SEUIL_CONCENTRICITÉ)

ET (*perpendicularité* > SEUIL_PERPENDICULAIRE)

ET (*dist_connexion* < SEUIL_DIST_CONNEXION)

ET (*contact* > SEUIL_CONTACT)

ALORS calculer facteur de confiance pour un BOUT EN T

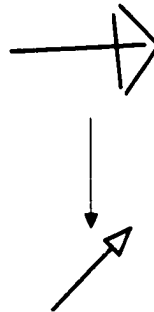
FIN_SI

FIN_SI

Paramètres

Paramètre	Valeur
SEUIL_CONCENTRICITÉ	0,2
SEUIL_PERPENDICULAIRE	0,85
SEUIL_DIST_CONNEXION	0,3
SEUIL_CONTACT	0,50

1.3.2 Bout en flèche



Heuristique de reconnaissance

(note : *primitive_1* est la référence et *primitive_2* et *primitive_3* sont les deux parties obliques qui forment la tête de la flèche ; leurs relations topologiques sont calculées par rapport à la référence)

SI *primitive_1* est de type LIGNE

ET *primitive_2* est de type LIGNE

ET *primitive_3* est de type LIGNE

SI cardinalité de *primitive_1* est SIMPLE ou NULLE

ET cardinalité de *primitive_2* est SIMPLE ou NULLE

ET cardinalité de *primitive_3* est SIMPLE ou NULLE

ET facteur_avant_arrière de *primitive_2* = facteur_avant_arrière de *primitive_3*

ET facteur_gauche_droite de *primitive_2* ≠ facteur_gauche_droite de *primitive_3*

ALORS SI (concentricité_primitive1_et_2 < SEUIL_CONCENTRICITÉ)

ET (perpendicularité_primitive1_et_2 ENTRE SEUIL_PERP_MIN et SEUIL_PERP_MAX)

ET (parallélisme_primitive1_et_2 ENTRE SEUIL_PARA_MIN et SEUIL_PARA_MAX)

ET (dist_connexion_primitive1_et_2 < SEUIL_DIST_CONNEXION)

ET (contact_primitive_1_et_2 > SEUIL_CONTACT)

ET (superposition_primitive1_et_2 < SEUIL_SUPERPOSITION)

ET (similitude_primitive1_et_2 < SEUIL_SIMILITUDE)

ET (distance_primitive_1_et_2 > SEUIL_DISTANCE)

ALORS calculer facteur de confiance pour un BOUT EN
FLÈCHE

FIN_SI

FIN_SI

Paramètres

Paramètre	Valeur
SEUIL_CONCENTRICITÉ	0,2
SEUIL_PERP_MIN	0,5
SEUIL_PERP_MAX	0,8
SEUIL_PARA_MIN	0,5
SEUIL_PARA_MAX	0,8
SEUIL_DIST_CONNEXION	1,0
SEUIL_CONTACT	0,7
SEUIL_SUPERPOSITION	0,2
SEUIL_SIMILITUDE	0,3
SEUIL_DISTANCE	0,5

1.3.3 Signe plus (+)



Heuristique de reconnaissance

(note : *primitive_1* est la référence et *primitive_2* est le voisin perpendiculaire)

SI *primitive_1* est de type LIGNE

ET *primitive_2* est de type LIGNE

SI *cardinalité de primitive_1* est NULLE

ET *cardinalité de primitive_2* est NULLE

ALORS SI (*concentricité* > SEUIL_CONCENTRICITÉ)

ET (*perpendicularité* > SEUIL_PERPENDICULARITÉ)

ET (*similitude* > SEUIL_SIMILITUDE)

ET (*superposition* > SEUIL_SUPERPOSITION)

ALORS calculer facteur de confiance pour un **SIGNE PLUS**

FIN_SI

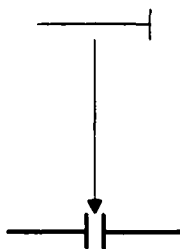
FIN_SI

Paramètres

Paramètre	Valeur
SEUIL_CONCENTRICITÉ	0,85
SEUIL_PERPENDICULAIRE	0,85
SEUIL_SIMILITUDE	0,65
SEUIL_SUPERPOSITION	0,85

1.4 Reconnaissance des symboles

1.4.1 Condensateur



Heuristique de reconnaissance

(note : *primitive_1* est le bout en T de référence et *primitive_2* est le bout en T voisin)

SI *primitive_1* est de type BOUT_EN_T

ET *primitive_2* est de type BOUT_EN_T

SI cardinalité de *primitive_1* est NULLE

ET cardinalité de *primitive_2* est NULLE

ALORS SI (*concentricité* < SEUIL_CONCENTRICITÉ)

ET (*parallélisme* > SEUIL_PARALLÈLE)

ET (*similitude* > SEUIL_SIMILITUDE)

ET (*superposition* < SEUIL_SUPERPOSITION)

ET (*codirection* < SEUIL_CODIRIGÉ)

ET (*colinéaire* < SEUIL_COLINÉAIRE)

ET (*contact* > SEUIL_CONTACT)

ET (*distance* > SEUIL_DISTANCE)

ALORS calculer facteur de confiance pour un
CONDENSATEUR

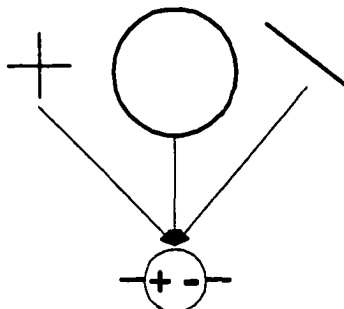
FIN_SI

FIN_SI

Paramètres

<i>Paramètre</i>	<i>Valeur</i>
SEUIL_CONCENTRICITÉ	0,3
SEUIL_PARALLÈLE	0,75
SEUIL_SIMILITUDE	0,6
SEUIL_SUPERPOSITION	0,4
SEIUL_CODIRIGÉ	0,6
SEUIL_COLINÉAIRE	0,75
SEUIL_CONTACT	0,1
SEUIL_DISTANCE	0,8

1.4.2 Source de tension



Heuristique de reconnaissance

(note : *primitive_1* est le cercle de référence, *primitive_2* est le signe plus et *primitive_3* est la ligne ; les relations topologiques des *primitive_2* et *primitive_3* sont calculées par rapport à la référence *primitive_1*)

SI *primitive_1* est de type CERCLE

ET *primitive_2* est de type SIGNE_PLUS

ET *primitive_3* est de type LIGNE

ALORS SI (*superposition_primitive1_et_2* > SEUIL_SUPERPOSITION)

ET (*similitude_primitive1_et_2* ENTRE SEUIL_SIMILITUDE_MIN et SEUIL_SIMILITUDE_MAX)

ET (*concentricité_primitive1_et_2* ENTRE SEUIL_CONCEN_MIN et SEUIL_CONCEN_MAX)

ALORS calculer facteur de confiance pour une SOURCE DE TENSION

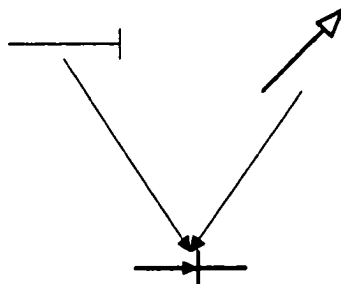
FIN_SI

FIN_SI

Paramètres

Paramètre	Valeur
SEUIL_SUPERPOSITION	0,4
SEUIL_SIMILITUDE_MIN	0,1
SEUIL_SIMILITUDE_MAX	0,6
SEUIL_CONCEN_MIN	0,2
SEUIL_CONCEN_MAX	0,8

1.4.3 Diode



Heuristique de reconnaissance

(note : *primitive_1* est le bout en T de référence et *primitive_2* est le bout en flèche voisin)

SI *primitive_1* est de type BOUT_EN_T

ET *primitive_2* est de type BOUT_EN_FLÈCHE

SI *cardinalité de primitive_1* est NULLE

ET *cardinalité de primitive_2* est NULLE

ALORS SI (*concentricité* < SEUIL_CONCENTRICITÉ)

ET (*parallélisme* > SEUIL_PARALLÈLE)

ET (*similitude* > SEUIL_SIMILITUDE)

ET (*superposition* < SEUIL_SUPERPOSITION)

ET (*codirection* < SEUIL_CODIRIGÉ)

ET (*colinéaire* < SEUIL_COLINÉAIRE)

ET (*contact* > SEUIL_CONTACT)

ET (*distance* > SEUIL_DISTANCE)

ALORS calculer facteur de confiance pour une DIODE

FIN_SI

FIN_SI

Paramètres

<i>Paramètre</i>	<i>Valeur</i>
SEUIL_CONCENTRICITÉ	0,3
SEUIL_PARALLÈLE	0,75
SEUIL_SIMILITUDE	0,6
SEUIL_SUPERPOSITION	0,4
SEIUL_CODIRIGÉ	0,6
SEUIL_COLINÉAIRE	0,75
SEUIL_CONTACT	0,1
SEUIL_DISTANCE	0,8

ANNEXE 2

Description structurelle des composantes

Nous décrivons dans cette annexe la structure des composantes qui peuvent être reconnues par notre système.

2.1 Primitives

Les primitives sont les composantes du plus bas niveau ; elles sont nécessairement obtenues à partir d'une ou plusieurs lignes ou arcs.



Figure 93 - Ligne

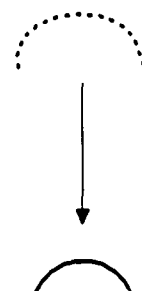


Figure 94 - Arc



Figure 95 - Cercle

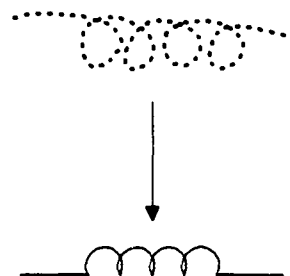


Figure 96 - Bobine

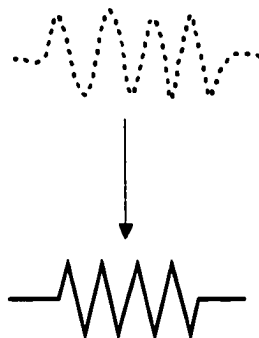


Figure 97 - Résistance

2.2 Composantes spécifiques

Les composantes spécifiques sont faites à partir de primitives.

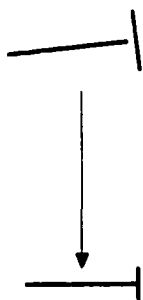


Figure 98 - Bout en T



Figure 99 - Bout en C

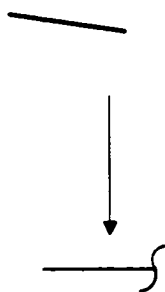


Figure 100 - Bout en S

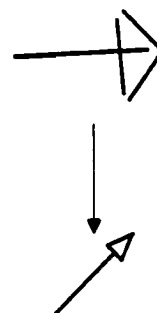


Figure 101 - Bout en flèche



Figure 102 - Signe plus

2.3 Symboles

Les symboles sont obtenus à partir d'une ou plusieurs primitives, composantes spécifiques ou symboles.

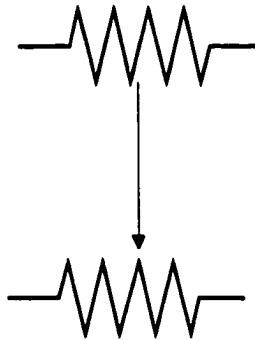


Figure 103 - Résistance

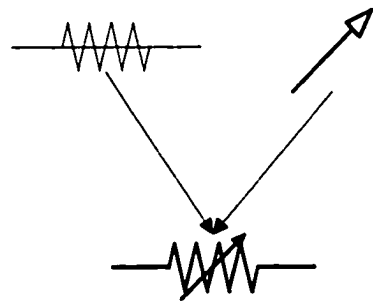


Figure 104 - Résistance variable

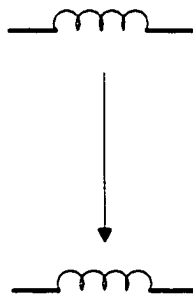


Figure 105 - Inductance

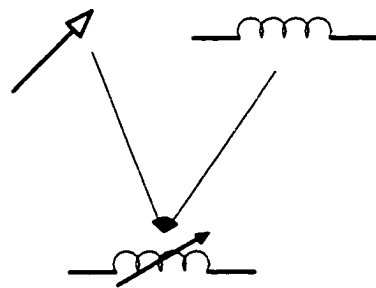


Figure 106 - Inductance variable

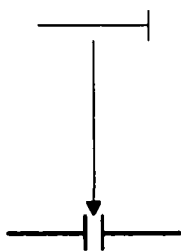


Figure 107 - Condensateur

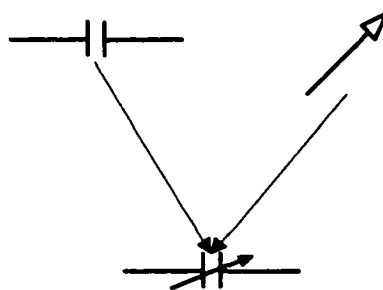


Figure 108 - Condensateur variable

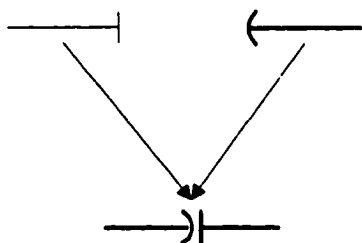


Figure 109 - Condensateur 2

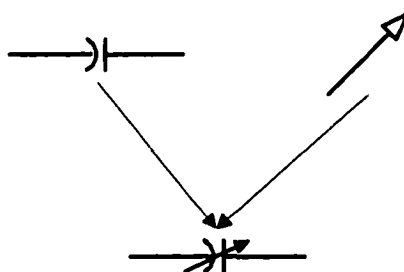


Figure 110 - Condensateur 2 variable

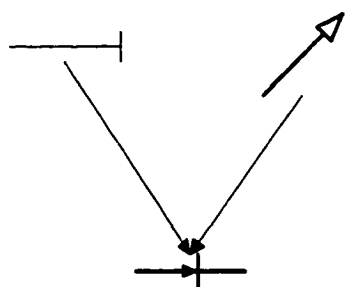


Figure 111 - Condensateur

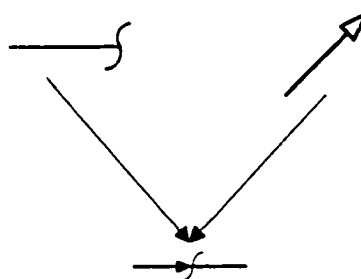


Figure 112 - Condensateur variable

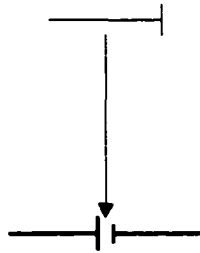


Figure 113 - Condensateur 2

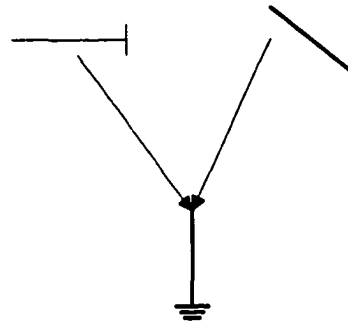


Figure 114 - Condensateur 2 variable

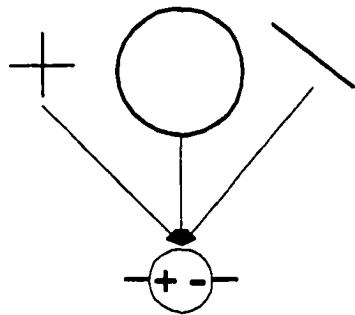


Figure 115 - Source de tension

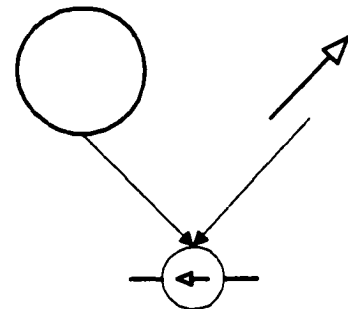


Figure 116 - Source de courant

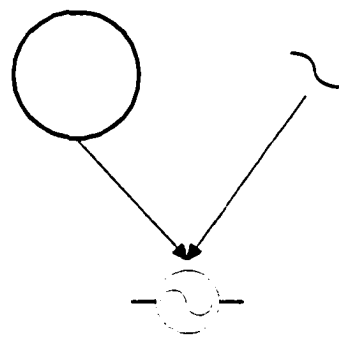


Figure 117 - Source alternative

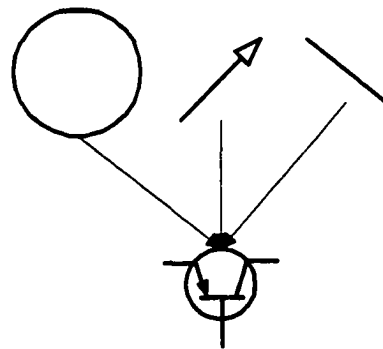
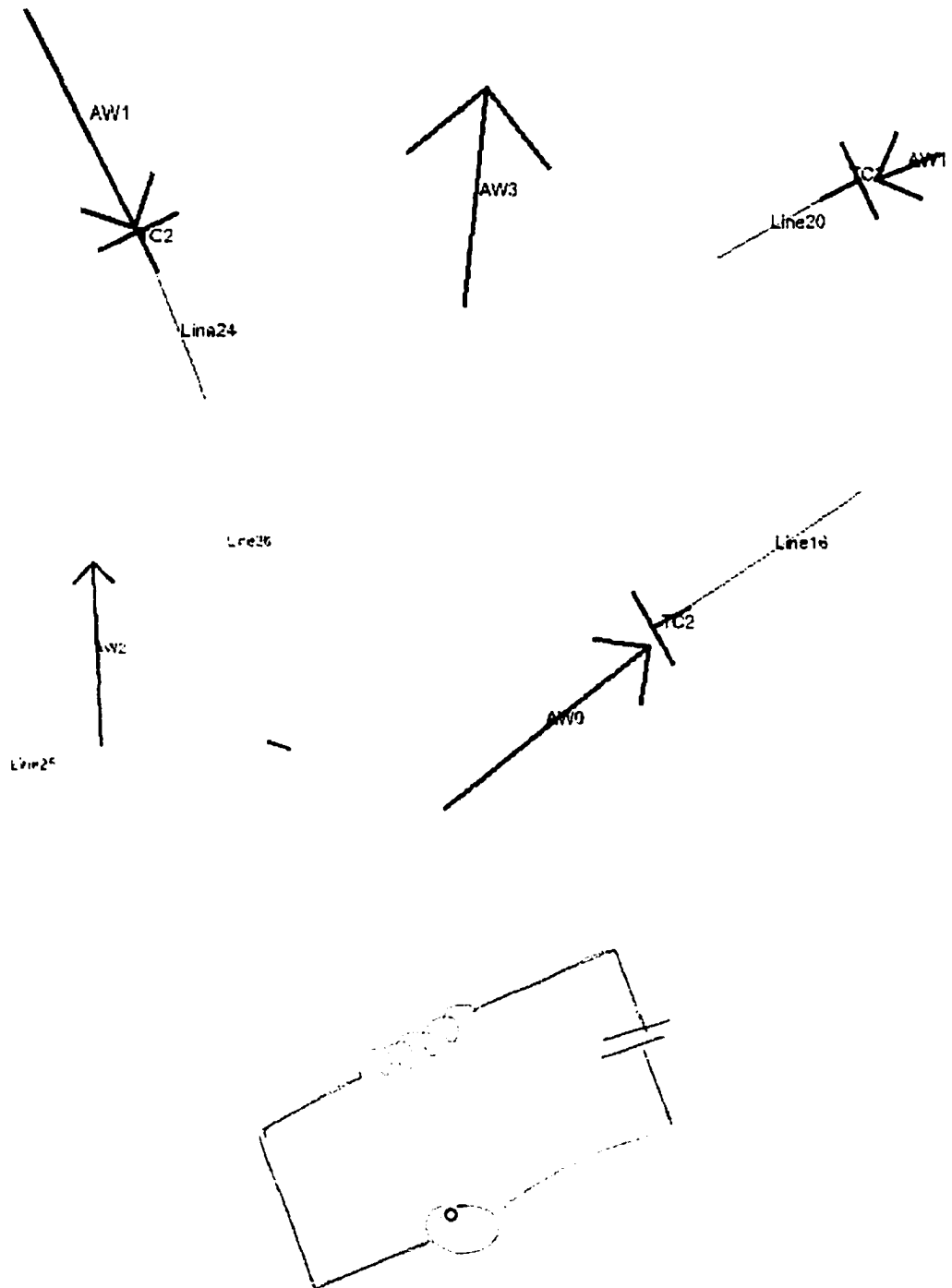


Figure 118 - Transistor

ANNEXE 3

Résultats d'embellissement

Voici quelques-unes des images illustrant les capacités d'embellissement du système :



ANNEXE 4

Références électroniques

Au cours de cette recherche, nous avons accumulé toute une gamme de références électroniques qui à notre avis seraient d'une grande utilité pour quiconque souhaiterait poursuivre des recherches dans cette direction.

Les liens électroniques en question peuvent être consultés à cette adresse :

<http://livia.etsmtl.ca/~jpvalois/references>

On retrouvera parmi ces références des pistes de recherche portant sur les sujets suivants :

- un format de représentation standard de l'encre électronique (*JOT* : <http://hwr.nici.kun.nl/unipen/jot.html>),
- une bibliographie électronique spécifique au domaine de la reconnaissance de formes et de la vision (*Rosenfeld USC Vision Bibliography* : <http://iris.usc.edu/Vision-Notes/rosenfeld/contents.html>),
- divers didacticiels portant sur les thèmes de notre recherche (approche perceptive de la Gestalt : <http://www.unc.edu/courses/psyc21/3-15-99>),
- de multiples outils logiciels, liens vers divers codes source et bibliothèques graphiques (librairie SATIN : <http://quir.berkeley.edu/projects/satin>).

ANNEXE 5

Références thématiques

Nous donnons dans cette annexe une liste de références classées par thème.

5.1 Techniques de reconnaissance

Mankoff, Abowd et Hudson (1999), Blostein (1996), Saund et Moran (1995), Julia et Faure (1995), Mohr (1990), Kato et Inokuchi (1990), Lee et al. (1990), Kojima et Toida (1988), Murase et Wakahara (1985) et Lin (1982).

5.2 Extraction de primitives

Dosch et al. (2000), Rosin et West (1989), O'Gorman (1988), Rosenfeld et Johnston (1973), Rosenfeld et Weszka (1975) et Saund (1993).

Teh et Chin (1989), Wu et Wang (1993), Wuesher et Boyer (1991), Deguchi (1988), Grimson (1989), Saint-Marc, chen et Medioni (1991), Chen, Lee et Pavlidis (1991), Asada et Brady (1986), Freeman et Davis (1977), Fischler et Bolles (1986), Aoyama et Kawagoe (1991) et Mokhtarian et Mackworth (1986).

5.3 Pré-traitement

Brown et Ganapathy (1983), Guerfali et Plamondon (1993), Tappert et al. (1990) et Jenkins et Martin (1992).

5.4 Embellissement

Kato et Inokuchi (1990), Lin (1982), Igarashi et al. (1997b), Igarashi et al. (1997a), Pavlidis et Van Wyk (1985), Murase et Wakahara (1985), Julia et Faure (1995) et Jenkins et Martin (1992) et Sutherland (1963).

BIBLIOGRAPHIE

Aoyama, H., & Kawagoe, M. (1991). A Piecewise Linear Approximation Method Preserving Visual Feature Points of Original Figures. *CVGIP: Graphical Models and Image Processing*, 53(5), 435-446.

Apte, A., Vo, V., & Kimura, T. D. (1993). *Recognizing multi-stroke geometric shapes: an experimental evaluation*. Paper presented at the Symposium on User Interface Software and Technology, Atlanta, GA USA.

Asada, H., & Brady, M. (1986). The Curvature Primal Sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1), 2-14.

Badros, G. J. (1998). *Constraints in Interactive Graphical Applications*. Unpublished Ph.D. General Examination, University of Washington, Washinton.

Blostein, D. (1996). General Diagram-Recognition Methodologies. In K. Tombre & R. Kasturi (Eds.), *Graphics Recognition : methods and applications* (Vol. 1072, pp. 106-122). Berlin: Springer.

Brown, M. K., & Ganapathy, S. (1983). Preprocessing techniques for cursive script word recognition. *Pattern recognition*, 16(5), 447-458.

Bunke, H. (1982). Attributed Programmed Graph Grammars and Their Application to Schematic Diagram Interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(6), 574-582.

Chen, M.-H., Lee, D., & Pavlidis, T. (1991). Residual Analysis for Feature Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1), 30-40.

Chhabra, A. K. (1997). Graphics Symbol Recognition : An Overview. In K. Tombre & A. K. Chhabra (Eds.), *Graphics Recognition : algorithms and systems* (Vol. 1389, pp. 68-79). Berlin: Springer.

Côté, M., Lecolinet, E., Cheriet, M., & Suen, C. Y. (1998). Automatic Reading of Cursive Scripts Using a Reading Model and Perceptual Concepts: the Percepto System. *IJDAR*, 1(1), 3-17.

Deguchi, K. (1988). *Multi-Scale Curvatures for Contour Feature Extraction*. Paper presented at the Proceedings of the 9th International Conference on Pattern Recognition.

Dosch, P., Masini, G., & Tombre, K. (2000, 3-7 September 2000). *Improving Arc Detection in Graphics Recognition*. Paper presented at the International Conference on Pattern Recognition - ICPR 2000, Barcelona, Spain.

- Fahn, C.-S., Wang, J.-F., & Lee, J.-Y. (1988). A Topology-Based Component Extractor for Understanding Electronic Circuit Diagrams. *Computer Vision, Graphics and Image Processing*, 44, 119-138.
- Fischler, M. A., & Bolles, R. C. (1986). Perceptual Organization and Curve Partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1), 100-105.
- Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1990). *Computer Graphics, Principles and Practice*. New York: Addison Wesley.
- Fonseca, M. J., & Jorge, J. A. (2000a). Experimental Evaluation of an Online Scribble Recognizer, *Proceedings of the 11th Portuguese Conference on Pattern Recognition (RecPAD 2000)* . Porto, Portugal.
- Fonseca, M. J., & Jorge, J. A. (2000b). Using Fuzzy Logic to Recognize Geometric Shapes Interactively, *Proceedings of the 9th Int. Conference on Fuzzy Systems (FUZZ-IEEE 2000)* . San Antonio, USA.
- Freeman, H., & Davis, L. S. (1977). A Corner-Finding Algorithm for Chain-Coded Curves. *IEEE Transactions on Computers*, 26, 297-303.
- Grimson, W. E. L. (1989). On the Recognition of Curved Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6), 632-643.
- Gross, M. D. (1996). The Electronic Cocktail Napkin - a computational environment for working with design diagrams. *Design Studies*, 17, 53-69.
- Guerfali, W., & Plamondon, R. (1993). Normalizing and Restoring On-Line Handwriting. *Pattern Recognition*, 26(3), 419-431.
- Harley, J. B., & Woodward, D. (1987). *The History of Cartography*. (Vol. 1). Chicago, IL: University of Chicago Press.
- Hutton, C., Cripps, m., Elliman, D. G., & Higgins, C. A. (1997). A Strategy for On-Line Interpretation of Sketched Engineering Drawings, *4th International Conference on Document Analysis and Recognition (ICDAR '97)* (Vol. 2, pp. 771-775). Ulm, Allemagne: IEEE-CS Press.
- Igarashi, T., Kawachiya, S., Matsuoka, S., & Tanaka, H. (1997a). In Search for an Ideal Computer-Assisted Drawing System, *Proc. of INTERACT '97 (The Sixth IFIP Conference on Human-Computer Interaction)* (pp. 104-111). Sydney, Australia.
- Igarashi, T., Matsuoka, S., Kawachiya, S., & Tanaka, H. (1997b). Interactive Beautification : A Technique for Rapid Geometric Design, *Proc. User Interface Software and Technology (UIST '97)* (pp. 105-114).

- Jenkins, D. L., & Martin, R. R. (1992). Applying Constraints to Enforce User's Intentions in Free-Hand 2-D Sketches. *Intelligent Systems Engineering*, 1(1), 31-49.
- Jenkins, D. L., & Martin, R. R. (1993). The Importance of Free-Hand Sketching in Conceptual Design : Automatic Sketch Input. In T. K. Hight & L. A. Stauffer (Eds.), *Design Theory & Methodology (DTM 93)* (Vol. DE-53, pp. 115-128): ASME.
- Jorge, J. A., & Fonseca, M. J. (1999). A Simple Approach to Recognize Geometric Shapes Interactively, *Proceedings of the Third Int. Workshop on Graphics Recognition (GREC'99)* . Jaipur, India.
- Julia, L., & Faure, C. (1995). Pattern Recognition and Beautification for a Pen-Based Interface, *3rd International Conference on Document Analysis and Recognition (ICDAR '95)* (Vol. 1, pp. 58-63). Montreal, Canada: IEEE-CS Press.
- Kasturi, R., Bow, S. T., El-MASri, W., Shah, J., Gattiker, J. R., & Mokate, U. B. (1990a). A System for Interpretation of Line Drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), 978-992.
- Kasturi, R., Chennubhotla, C., Raman, R., & O'Gorman, L. (1990b). Document Image Analysis : An Overview of Techniques for Graphics Recognition, *Proc. IAPR Workshop on Structural & Syntactic Pattern Recognition* (pp. 192-230).
- Kasturi, R., & Luo, H. (1997). Research Advances in Graphics Recognition : An Update, *Advances in Document Image Analysis, Proceedings of the First Brazilian Symposium (BSDIA '97)* (Vol. 1339, pp. 99-110). Curitiba, Brazil: LNCS.
- Kato, H., & Inokuchi, S. (1990). The Recognition Method for Roughly Hand-Drawn Logical Diagrams Based on Hybrid Utilization of Multi-Layered Knowledge. *ICPR '90*, 2, 578-582.
- Kojima, H., & Toida, T. (1988). Online Hand-Drawn Line-Figure Recognition and its Applications, *ICPR '88* (pp. 1138-1142).
- Lank, E. (2000). *The Diagram Recognition Process*, [PDF]. Available: <http://citeseer.nj.nec.com/did/209606> [2000, .
- Lee, S.-W., Kim, J. H., & Groen, F. C. A. (1990). Translation-, Rotation-, and Scale-Invariant Recognition of Hand-Drawn Electrical Circuit Symbols with Attributed Graph Matching. *SSPR90 Workshop*, 273-292.
- Lin, W. C. (1982). Computer Processing of Hand-Drawn Sketches and Diagrams. In C. Y. Suen & R. De Mori (Eds.), *Computer Analysis and Perception* (Vol. 1, pp. 162). Boca Raton: CRC Press, Inc.
- Loncaric, S. (1998). A Survey of Shape Analysis Techniques. *Pattern Recognition*, 31(8), 983-1001.

- Mankoff, J., Abowd, G. D., & Hudson, S. (1999). Interacting with Multiple Alternatives Generated by Recognition Technologies. *GVU GIT TechReport*, 26, 10.
- Messmer, B. T., & Bunke, H. (1995). Automatic Learning and Recognition of Graphical Symbols in Engineering Drawings. In K. Tombre & R. Kasturi (Eds.), *Graphics Recognition : methods and applications* (Vol. 1072, pp. 123-134). Berlin: Springer.
- Messmer, B. T., & Bunke, H. (1998a). Clustering and error-correcting matching of graphs for learning and recognition of symbols in engineering drawings. In J. Hull & S. Taylor (Eds.), *Document Analysis Systems II* (pp. 102 - 117). Singapore: World Scientific Publ. Co.
- Messmer, B. T., & Bunke, H. (1998b). A New Algorithm for Error-Tolerant Subgraph Isomorphism Detection. *IEEE TRansaction on Pattern Analysis and Machine Intelligence*, 20(5), 493-504.
- Mohr, R. (1990). A General Purpose Line Drawing Analysis System. In W. Scientific (Ed.), *Syntactic and Structural Pattern Recognition, Theory and Applications* (Vol. 7, pp. 479-498). Singapore: World Scientific.
- Mokhtarian, F., & Mackworth, A. (1986). Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1), 34-43.
- Murase, H., & Wakahara, T. (1985). Online Hand-Sketched Figure Recognition. *Pattern Recognition*, 19(2), 147-160.
- Nagy, G. (2000). Twenty Years of Document Image Analysis in PAMI. *IEEE Pattern Analysis and Machine Intelligence*, 22(1), 38-62.
- Nakagawa, M., Kato, N., Machii, K., & Souya, T. (1993). Principles of Pen Interface Design for Creative Work. *Second International Conference on Document Analysis and Recognition (ICDAR '93)*, 718-721.
- Nouboud, F., & Plamondon, R. (1990). On-Line Recognition of Handprinted Characters : Survey and Beta Tests. *Pattern Recognition*, 23(9), 1031-1044.
- O'Gorman, L. (1988). Curvilinear Feature Detection from Curvature Estimation. , 1116-1119.
- O'Gorman, L., & Kasturi, R. (1997). *Executive Briefing, Document Image Analysis*. (IEEE Computer Society ed.). Los Alamitos: IEEE.
- Okazaki, A., Kondo, T., Mori, K., Tsunekawa, S., & Kawamoto, E. (1988). An Automatic Diagram Reader with Loop-Structure-Based Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3), 331-341.

Pasternak, B., & Neumann, B. (1993). *Adaptable Drawing Interpretation Using Object Oriented and Constraint-Based Graphics Recognition*. Paper presented at the Second International Conference on Document Analysis and Recognition (ICDAR '93).

Pavlidis, T. (1980). Algorithms for Shape Analysis of Contours and Waveforms. *IEEE Pattern Analysis and Machine Intelligence*, 2(4), 301-312.

Pavlidis, T., & Van Wyk, C. J. (1985). An Automatic Beautifier for Drawings and Illustrations. *ACM Computer Graphics, Proc. of SIGGRAPH '85*, 225-234.

Petre, M. (1995, June 1995). Why Looking Isn't Always Seeing - Readership Skills and Graphical Programming. *Communications of the ACM*, 38, 33-44.

Plamondon, R., & Srihari, S. N. (2000). On-Line and Off-Line Handwriting Recognition : A Comprehensive Survey. *IEEE Pattern Analysis and Machine Intelligence*, 22(1), 63-84.

Rosenfeld, A., & Johnston, E. (1973). Angle Detection on Digital Curves. *IEEE Transactions on computers*, C-22, 875-878.

Rosenfeld, A., & Weszka, J. S. (1975). An Improved Method of Angle Detection on Digital Curves. *IEEE Transactions on Computers*, C-24, 940-941.

Rosin, P. L., & West, G. A. W. (1989). Segmentation of edges into lines and arcs. *Image and Vision Computing*, 7, 109-114.

Saint-Marc, P., Chen, J.-S., & Medioni, G. (1991). Adaptive Smoothing: A General Tool for Early Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6), 514-529.

Saund, E. (1993). Identifying Salient Circular Arcs On Curves. *CVGIP Image Understanding*, 58(3), 327-337.

Saund, E., & Moran, T. P. (1994). *A Perceptually Supported Sketch Editor*. Paper presented at the UIST'94, Marina del Rey, CA USA.

Saund, E., & Moran, T. P. (1995, June 1995). *Perceptual Organization in an Interactive Sketch Editing Application*. Paper presented at the Fifth International Conference on Computer Vision, Cambridge, Massachusetts.

Sutherland, I. E. (1963). *Sketchpad : A man-machine graphical communications systems*. Unpublished PhD Thesis, Massachusetts Institute of Technology, Cambridge.

Tappert, C. C., Suen, C. Y., & Wakahara, T. (1990). The State of the Art in Online Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8), 787-808.

- Teh, C.-H., & Chin, R. T. (1989). On The Detection of Dominant Points on Digital Curves. *IEEE Pattern Analysis and Machine Intelligence*, 11(8), 859-872.
- Tombre, K. (1996). Structural and syntactic methods in line drawing analysis: To which extend do they work? In P. Perner, P. Wang, & A. Rozenfeld (Eds.), *Advances in Syntactic and Structural Pattern Recognition (SSPR'96)* (pp. 310-321). Germany.
- Tombre, K., Ah-Soon, C., Dosch, P., Habed, A., & Masini, G. (1998). Stable, Robust and Off-the-Shelf Methods for Graphics Recognition. *14th International Conference on Pattern Recognition (ICPR '98)*, 406-408.
- Ullman, D. G., Wood, S., & Craig, D. (1990). The Importance of Drawing in the Mechanical Design Process. *Computer & Graphics*, 14(2), 263-274.
- Valveny, E., & Marti, E. (1999). Application of Deformable Template Matching to Symbol Recognition in Hand-Written Architectural Drawings, *Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR '99)* (pp. 483-486). Piscataway (NJ): IEEE.
- Wu, W.-Y., & Wang, M.-J. J. (1993). Detecting the Dominant Points by the Curvature-Based Polygonal Approximation. *CVGIP : Graphical Models and Images Analysis*, 55(2), 79-88.
- Wuesher, D. M., & Boyer, K. L. (1991). Robust Contour Decomposition Using a Constant Curvature Criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1), 41-51.
- Yu, Y., Samal, A., & Seth, S. C. (1997). A System for Recognizing a Large Class of Engineering Drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8), 868-890.
- Zhao, R. (1993). *Incremental Recognition in Gesture-Based and Syntax-Directed Diagram Editors*. Paper presented at the Conference on Human Factors and Computing Systems, Amsterdam, The Netherlands.
- Zhao, X., Liu, X., Zheng, S., Pan, B., & Yuan, T. Y. (1997). Online Recognition Handwritten Mathematical Symbols, *4th International Conference on Document Analysis and Recognition (ICDAR '97)* (pp. 645-648). Ulm, Germany: IEEE-CS Press.