

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE
EN
MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES

PAR
JEAN-DENIS CARON

APPROCHES GÉOMÉTRIQUES POUR LA DÉTECTION
DE L'ANGLE D'ORIENTATION D'UN TEXTE

JANVIER 2000

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Remerciements

Je tiens à remercier très sincèrement mes deux directeurs, Prosenjit K. Bose et Alain Chalifour, pour leurs idées, leurs suggestions, leur encouragement, leur support, pour avoir su aussi bien me guider et par-dessus tout, pour leur l'amitié. Je veux particulièrement remercier Jit de m'avoir convaincu de poursuivre des études supérieures, pour m'avoir initié et donné le goût à la recherche, pour m'avoir fait découvrir le monde de la géométrie algorithmique, et pour le soutien financier qu'il a apporté. Merci à Alain de m'avoir pris en charge lors du départ de Jit et pour sa rigueur lors de la rédaction de ce mémoire. Celui-ci n'aurait jamais existé sans ces deux personnes.

J'aimerais aussi remercier Kilani Ghoudi pour son aide très bénéfique pour la partie touchant les statistiques et lors de la planification des simulations réalisées dans ce mémoire.

Je tiens à remercier les autres membres du comité de lecture, les professeurs Bruno Rémillard et Fathallah Nouboud, pour leur lecture critique et les commentaires apportés à mon travail de recherche.

Merci, du plus profond de mon coeur, à ma famille, ma mère Monique, mon père Denis, mes soeurs Patricia et Geneviève ainsi qu'à ma copine Nancy, pour leur amour, leur encouragement, leur support et pour avoir cru en moi tout au long de mes études.

Finalement, la bourse obtenue de la Fondation du C.E.U. de l'Université du Québec à Trois-Rivières et la bourse octroyée par le Ministère de l'Industrie du Canada ont été un soutien financier grandement apprécié.

Dédicace

Ce mémoire est dédié à mes parents, Denis et Monique. Sans leur amour, leur support et leur encouragement au fil des années, ce mémoire n'existerait pas.

Jean-Denis Caron

Décembre, 1999

Résumé

Ce mémoire traite du problème de la détection de l'angle d'orientation d'un texte. Nous allons définir l'angle d'orientation d'un texte comme l'angle $\phi \in [-90^\circ, 90^\circ]$, qu'a avec l'horizontale, l'ensemble des droites passant par chacune des lignes du texte. Ce problème constitue une des étapes importantes dans le processus de traitement automatique de documents.

Dans ce mémoire, trois nouvelles approches sont proposées et sont comparées avec deux autres méthodes déjà existantes. Des simulations, effectuées sur des textes de différentes langues (français, anglais, japonais, allemand, russe et espagnol), nous permettent de déterminer la méthode la plus précise et de démontrer que cette dernière est indépendante du langage en autant que celui-ci vérifie la condition suivante : *la distance entre deux caractères consécutifs est plus petite que la distance entre deux caractères sur deux lignes différentes*. Des simulations supplémentaires ont été faites afin de démontrer que la rapidité des méthodes peut être améliorée en ne traitant qu'un pourcentage d'un texte plutôt que le texte entier.

Les méthodes présentées sont basées sur la reconnaissance de formes et la géométrie algorithmique. Une statistique a été développée pour l'estimation de l'angle d'orientation d'un texte.

Table des matières

1	Introduction	1
2	Pré-traitement	5
2.1	Détection des contours du texte	7
2.2	Bounding boxes	15
2.3	Ensemble de points représentatifs du texte	17
3	Ittner et le Bamboo field	20
3.1	Méthode d'Ittner	20
3.2	Bamboo field	23
4	Nouvelles approches	26
4.1	Bamboo field amélioré	26
4.1.1	Estimation de l'AOT	29

4.2	Nearest neighbor	30
5	Comparaison	33
6	Simulations	48
6.1	Indépendance du langage	49
6.2	Amélioration de la rapidité	50
7	Conclusion	59

Table des figures

2.1	Image originale contenant un texte anglais	6
2.2	Détermination du pixel à gauche d'un autre pixel	7
2.3	Exemple d'application de l'algorithme du <i>contour following</i>	9
2.4	Résultat du deuxième algorithme de détection de contour	10
2.5	Sous-domaines binaires d'une image	10
2.6	a) Contour simple, b) Contour faiblement simple	11
2.7	a) Adhérence simple, b) Adhérence composée	13
2.8	Directions de Freeman	13
2.9	a) Paires admissibles, b) Paires non admissibles, c) Paires admissibles	13
2.10	Application du bounding boxes sur l'image originale	16
2.11	Application de l'algorithme du <i>MinDisc</i> sur l'image originale	17
2.12	Représentation d'un boîte	18

2.13	Ensemble de points représentatifs du texte	18
3.1	Arbre sous-tendant minimal du texte original	23
3.2	Exemple d'un segment non-représentatif généré par l' <i>ASM</i>	24
3.3	Exemple d'un sommet de degré 5	24
3.4	Graphe obtenu après l'exécution du <i>bamboo field</i>	25
4.1	Calcul de l'angle entre deux segments	27
4.2	Matrice d'adjacence	28
4.3	Application du Bamboo field	28
4.4	Application du Bamboo field amélioré	29
4.5	Exemple de la non-symétrie de la relation du plus proche voisin . . .	31
4.6	Application du <i>NNG</i> à l'image originale	31
4.7	Application du <i>NNGA</i> à l'image originale	32
5.1	Histogramme de distribution des angles	36
5.2	Texte original anglais ($AOT = 60^\circ$)	42
5.3	Résultat de la méthode d'Ittner sur le texte anglais	43
5.4	Graphe obtenu avec le <i>Bamboo field</i>	44
5.5	Application du <i>Bamboo field amélioré</i>	45

5.6	Ensemble de segments générés par le <i>NNG</i>	46
5.7	Ensemble de segments générés par le <i>NNGA</i>	47
6.1	Texte original en japonais ($AOT = 30^\circ$)	51
6.2	Résultat de la méthode <i>BFA</i> sur le texte en japonais	52
6.3	Texte original en allemand ($AOT = -30^\circ$)	53
6.4	Résultat de la méthode <i>BFA</i> sur le texte en allemand	54
6.5	Texte original en espagnol ($AOT = 60^\circ$)	55
6.6	Résultat de la méthode <i>BFA</i> sur le texte en espagnol	56
6.7	Texte original en français ($AOT = 0^\circ$)	57
6.8	Méthode du <i>BFA</i> appliquée sur 100% du texte en français	58
6.9	Méthode du <i>BFA</i> appliquée sur 25% du texte en français	58

Liste des tableaux

2.1	Paires de déplacements successifs admissibles	14
5.1	Comparaison entre les 5 différentes approches	38
5.2	Tableau comparatif entre les estimateurs	40
5.3	Estimation de faibles AOT	41
6.1	Application du BFA sur des textes de différentes langues	50
6.2	Résultats de la méthode BFA appliquée à 25% et à 100% des textes	57

Chapitre 1

Introduction

Dans le cadre de ce mémoire, nous abordons le problème de la détection de l'angle d'orientation d'un texte (*AOT*) pour lequel nous proposons une approche basée sur la reconnaissance de formes et la géométrie algorithmique. Nous définirons dans un premier temps, l'*AOT* comme étant l'angle $\phi \in [-90^\circ, 90^\circ]$, qu'a avec l'horizontale, l'ensemble des droites passant par chacune des lignes du texte. Nous préciserons ultérieurement le sens attribué à ces droites par le biais d'une statistique développée dans le but d'estimer l'*AOT*.

Le télécopieur est largement utilisé dans nos sociétés et il fait partie du quotidien de plusieurs entreprises ou organismes pour le transfert électronique d'informations diverses. Malgré sa grande utilité, cette technologie présente certaines imperfections, en particulier la présence d'un angle de déviation dans un texte télécopié lequel peut être introduit mécaniquement lors du processus de transfert. Un problème similaire en reprographie est celui de la numérisation des textes. Dans chacun de ces cas, un mauvais alignement ou positionnement du support d'un texte d'orientation connue, induira un biais dans l'orientation du texte après transfert. De plus, la détection et la correction de l'*AOT* constitue une étape importante lors du traitement automatique de documents et en particulier lors de la reconnaissance automatique des caractères. Ces traitements nécessitent plusieurs étapes de post-traitement ap-

pliquées séquentiellement, soit la numérisation du texte, le seuillage de l'image pour éliminer le bruit, le rehaussement de l'image numérisée, la détection et la correction de l'*AOT*, la segmentation facilite l'isolement ou le regroupement des caractères et finalement la classification et la reconnaissance des caractères. Conséquemment, la qualité de la reconnaissance des caractères et le traitement contextuel d'un document sont fortement corrélés à la qualité de l'estimation de l'*AOT*.

Il apparaît donc important de mettre au point des méthodes rapides et efficaces d'estimation de l'*AOT*, afin d'apporter les corrections nécessaires à un document reprographié. Or, il est bien connu, que la difficulté majeure de cette problématique ne réside pas tant dans la correction de l'*AOT* mais bien dans son estimation.

À ce jour, plusieurs méthodes ont été développées afin de détecter ou d'estimer l'*AOT*. Une famille de méthodes ont été publiées [1, 3, 11, 14, 15, 24, 25] lesquelles utilisent la transformée de Hough. Ce type de méthodes peut fournir des résultats très précis comme par exemple la méthode présentée dans [25] qui estime l'*AOT* à 0.1° . Ces méthodes appliquent la transformée de Hough soit sur les pixels ou soit sur les composantes connectées de l'image et associent l'*AOT* au point dominant dans l'espace de Hough. Le principe est de transposer les points (ou les composantes connectées) de l'image d'un plan xy vers un autre plan $\rho\theta$ à l'aide de l'équation

$$\rho_i = x \cos \theta_i + y \sin \theta_i,$$

où $\theta_i \in [-90^\circ, 90^\circ]$, $\rho_i \in [-\rho_{max}, \rho_{max}]$, $\rho_{max} = \sqrt{h^2 + l^2}$ et où h et l sont respectivement la hauteur et la largeur en pixels de l'image. Or, cette approche requiert des hypothèses sur l'intervalle contenant l'angle, plus l'intervalle contenant l'*AOT* est grand, plus le traitement devient complexe et l'espace mémoire requis devient important. Les méthodes utilisant la transformée de Hough sont donc intéressantes dans la mesure où nous anticipons l'intervalle, a priori précis, auquel appartient l'*AOT*.

Une autre approche, présentée dans [22], utilise une méthode d'extraction des lignes d'un texte pour l'approximation de l'*AOT*. Cette approche s'est avérée, après comparaison, plus robuste que l'algorithme de *Baird* [3], un des premiers algo-

rithmes développés qui demeure toujours très populaire de nos jours. Par contre, cette méthode nécessite des conditions quant à la connaissance à priori de l'intervalle contenant l'*AOT*. Malheureusement, cette approche devient inefficace lorsque l'*AOT* est plus grand que $|15^\circ|$.

Une approche basée sur la coopération entre deux réseaux de neurones [19] est aussi utilisée pour calculer l'*AOT*. Le premier réseau est un perceptron à trois couches qui reçoit en entrée la matrice de corrélation d'un signal obtenu à partir de l'image contenant le texte. Ce réseau génère une première approximation qui sert d'entrée au second perceptron multi-couches qui utilise le maximum de vraisemblance pour approximer l'*AOT*. Le désavantage, lorsque l'on utilise un réseau de neurones, est l'obligation d'effectuer une étape d'apprentissage, et donc ce dernier dépend des textes utilisés lors de l'apprentissage.

Une autre méthode, basée sur les fractals et sur les moindres carrés [26], a également été proposée. La théorie des fractals est extrêmement complexe et il n'apparaît pas naturel de l'utiliser pour résoudre le problème de l'*AOT*.

Une autre famille de méthodes est celle utilisant le gradient [21, 23]. Pour ces méthodes, une extraction des lignes du texte est faite à l'aide d'un *blur* et d'un rétrécissement de l'image. Par la suite, différents masques sont appliqués pour calculer l'intensité et la direction du gradient. Généralement, ce type d'approche ne fournit pas des résultats très précis car l'approximation de l'*AOT* est faite avec un pas de 1 degré. Si l'angle d'orientation d'un texte est de 1.5° , l'*AOT* obtenu par une de ces méthodes serait de 1° ou de 2° . Les méthodes que nous présenteront dans ce mémoire ont une précision au centième de degré près.

Une méthode efficace, basée sur la géométrie algorithmique, est celle proposée par Ittner [13]. Cependant, Pateras et al. [17], ont trouvé des failles dans cette approche et ont proposé une amélioration de l'algorithme d'Ittner, le *Bamboo field*. Nous présenterons en détail ces deux méthodes dans les chapitres qui suivront ainsi que deux nouvelles approches dans le but d'estimer l'*AOT*. La première approche développée est une amélioration du *Bamboo field* tandis que la seconde, entièrement

indépendante des méthodes précédentes, utilise tout comme la première, des notions de géométrie algorithmique.

La plupart des méthodes existantes ne sont pas indépendantes du langage utilisé. En contre-partie, les deux nouvelles méthodes présentées dans ce mémoire sont indépendantes du langage (français, anglais, russe, allemand, ...) sous l'hypothèse suivante : *la distance entre deux caractères consécutifs d'une même ligne est plus petite que la distance entre deux caractères appartenant à des lignes différentes*. Des simulations ont été effectuées afin de confirmer l'indépendance des deux nouvelles approches de calcul de l'*AOT* en regard du langage utilisé dans un texte.

Ce mémoire de recherche est constitué de cinq autres chapitres. Au chapitre 2 nous présentons les étapes de pré-traitement communes aux méthodes d'Ittner, du *Bamboo field* et des deux nouvelles approches développées. Au chapitre 3 nous présentons les méthodes d'Ittner et du *Bamboo field* tandis que les trois nouvelles méthodes sont présentées au chapitre 4. Au chapitre 5 nous comparons les performances des quatre méthodes à partir d'une banque de documents, dans le but de déterminer les performances de chacune. Au chapitre 6, nous présentons les tests effectués à l'aide des méthodes développées sur des documents en différentes langues pour valider l'hypothèse d'indépendance du langage. Finalement, nous présentons nos conclusions sur l'ensemble des méthodes utilisées pour le calcul de l'*AOT* et les travaux à envisager pour améliorer ces approches.

Une partie des résultats présentés dans ce mémoire ont fait l'objet d'une conférence arbitrée à la conférence *Canadian Conference on Computational Geometry 98* (Montréal (Qc), Canada, août 1998) [4] et d'une affiche à la conférence *Vision Interface 99* (Trois-Rivières (Qc), Canada, mai 1999) [6].

Chapitre 2

Pré-traitement

Dans ce chapitre nous présentons de façon détaillée les différentes étapes de pré-traitement utilisées par chacune des quatre approches de calcul de l'*AOT* faisant l'objet de notre étude, soit la méthode d'Ittner [13], la méthode du *Bamboo field* [17] et les trois nouvelles méthodes développées dans ce mémoire (Chapitre 3 et Chapitre 4). Chacun de ces algorithmes de pré-traitement fera l'objet d'une section de ce chapitre. Les étapes successives de pré-traitement appliquées à une même image originale pour chacune des méthodes étudiées sont les suivantes :

- la détection du contour du texte,
- la construction des *bounding boxes*,
- le repérage d'un ensemble de points représentatifs du texte.

Considérons une image binaire numérisée, notée \mathcal{I} , composée des caractères d'un texte imprimé. Chaque pixel noir de l'image sera indicé par la valeur 1 et chaque pixel blanc par la valeur 0. Afin d'illustrer chacune des étapes de pré-traitement et aussi dans le but de comparer les méthodes de détection, nous considérerons dans une première étape l'image de départ (originale) présentée à la figure 2.1. Cette image représente un texte anglais numérisé dont l'angle d'orientation est légèrement inférieur à 0° .

ABSTRACT

One of the most powerful and robust algorithms for text-line orientation in document analysis is the recent algorithm of Ittner. The source of power of this algorithm comes from performing a histogram analysis of the slopes of the edges of the *minimal spanning tree* of the set of representative points of the connected components (characters) in a block of text. In this note we propose an improvement of this method that replaces the minimal spanning tree by the *bamboo field*, a subgraph of the minimal spanning tree that increases the robustness of Ittner's approach. Experiments are described that substantiate the improved performance of the proposed algorithm.

FIG. 2.1 – Image originale contenant un texte anglais

Le point de départ de chacune des cinq méthodes de détection qui feront l'objet de notre étude est la construction d'une image transformée suite à l'application successive des trois étapes de pré-traitement. Il résultera de ces étapes de pré-traitement une image unique, constituée d'un ensemble de points représentatifs de la direction du texte, à partir desquels sera estimé l'angle d'orientation du texte. Pour arriver à cette représentation, on doit dans un premier temps détecter le contour de chacune des lettres ou groupe de caractères du texte. L'application d'un algorithme de *bounding boxes* sur l'ensemble des contours produira un ensemble de boîtes contenant chacune une lettre ou un groupe de caractères du texte. À partir d'un seuillage sur les dimensions de ces boîtes, sont épurées les accents et les signes de ponctuation lesquels ne fournissent aucune information pertinente sur l'*AOT*. L'utilisation des boîtes au lieu des pixels permet de diminuer le temps d'exécution des algorithmes puisqu'un nombre réduit d'objets sont ainsi traités. Finalement, l'ensemble des points représentatifs de l'image est construit à partir des centres des boîtes. Ces points pourront par la suite être reliés par un ensemble de segments, ce qui nous permettra d'estimer statistiquement l'*AOT*.

2.1 Détection des contours du texte

L'objectif de la première étape de pré-traitement est idéalement la détection du contour de chacune des lettres de l'image originale, lequel contour sera par la suite inscrit dans un polygone. À défaut d'obtenir un contour de chacun des caractères du texte, certains de ceux-ci seront éventuellement regroupés, en l'occurrence lorsque ces caractères seront connectés. Afin de simplifier l'écriture du texte nous utiliserons l'expression *élément calligraphique* pour signifier une lettre, un caractère calligraphique ou typographique, et un regroupement connexe de tels éléments ou unités d'écriture.

Dans un premier temps nous avons utilisé l'algorithme de *contour following* présenté dans Duda et al. [9]. Avant d'exposer l'algorithme en détail, voyons ce que signifie l'expression *pixel à gauche*. Le pixel à gauche d'un pixel p_j est le pixel p_k à gauche de la direction $\overline{p_i p_j}$, où $\overline{p_i p_j}$ représente un déplacement, p_i est le pixel de départ du dernier déplacement et p_j est le pixel d'arrivée de ce même déplacement. Nous appliquons la même logique pour trouver le pixel à droite d'un autre pixel. À la figure 2.2, si le pixel courant est le pixel p_2 et que le dernier déplacement s'est fait du pixel p_1 au pixel p_2 , alors le pixel à gauche dans ce cas est le pixel p_3 .

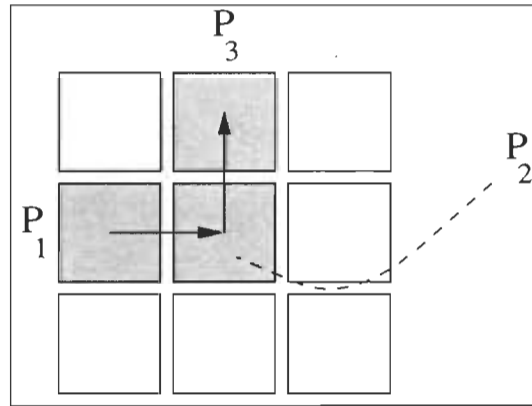


FIG. 2.2 – Détermination du pixel à gauche d'un autre pixel

Voyons maintenant la description de l'algorithme du *contour following* présenté sous la forme d'un pseudo-code.

ContourFollowing (Entrée : I)

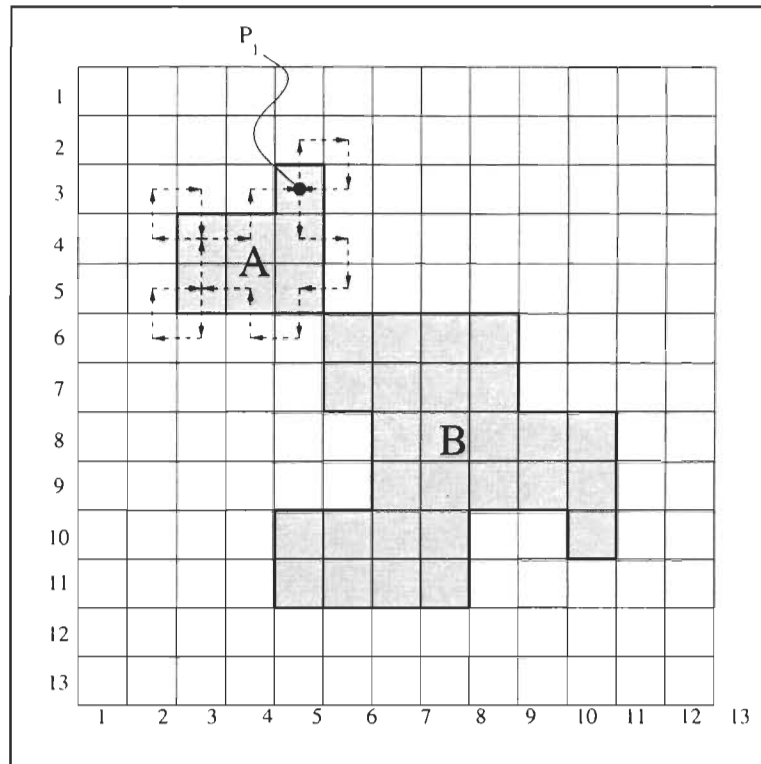
```

Pour chaque pixel  $I_{ij}$  de  $I$ 
  Si  $I_{ij} = 1$ 
    début =  $I_{ij}$ 
    suivant = pixel à gauche de  $I_{ij}$ 
    Tant que suivant  $\neq$  début
      Si suivant = 1
        suivant = pixel à gauche de  $I_{ij}$ 
      Sinon
        suivant = pixel à droite de  $I_{ij}$ 
    Fin Si
  Fin Tant que
Fin Si
Fin Pour

```

Fin ContourFollowing

Afin d'illustrer le fonctionnement de cet algorithme, on présente à la figure 2.3 les étapes de la construction du contour d'une image binaire simple. Il est à noter que cet algorithme construit une trajectoire de contour basé sur la connectivité à 4 voisins, ce qui peut occasionner une détérioration de l'information dans le cadre de notre problématique lors de l'isolement des lettres ou groupe de caractères d'un texte. A titre d'exemple, examinons la figure 2.3. Si l'on applique le *contour following* sur cette image, le pixel P_1 serait le premier pixel de départ rencontré. Le contour qui serait généré à partir de celui-ci est représenté en pointillé sur la figure. Nous remarquons que celui-ci sépare notre objet en deux parties, soit en A et en B alors qu'en réalité, l'objet qui nous intéresse est $A \cup B$. Si cet objet avait été une lettre, alors cette dernière aurait été traitée en deux parties distinctes ce qui aurait introduit un biais dans notre méthode. En regard de cette éventualité, nous avons opté pour une stratégie plus robuste, laquelle permettra d'éviter une telle perte d'information.

FIG. 2.3 – Exemple d'application de l'algorithme du *contour following*

A cet effet, nous avons retenu l'algorithme de détection de contours proposé par Chalifour *et al.* [7]. Afin d'appliquer cet algorithme, on suppose que chaque lettre ou caractère calligraphique est un ensemble de pixels noirs et connexe au sens de la connectivité à 4 voisins dont l'intérieur (s'il existe), constitué de pixels blancs, est ignoré ou mieux sont remplacés par des pixels noirs. Ce qui nous intéresse à cette étape, c'est plus l'enveloppe d'un caractère que sa reconnaissance en soi. Chaque contour, d'une lettre, d'un caractère calligraphique ou d'un groupe de caractères, qui sera généré par cet algorithme est un chemin simple, ou faiblement simple (voir plus loin), fermé, orienté positivement (antihoraire) et d'épaisseur égale à un pixel. Les pixels d'un contour seront blancs et pour des fins de présentation nous inverserons les couleurs dans la présentation des résultats. Finalement, chaque contour correspondra à une chaîne de longueur minimale au sens de la connectivité

à 8 voisins. On trouve à la figure 2.4 les contours obtenus après le pré-traitement du texte original.

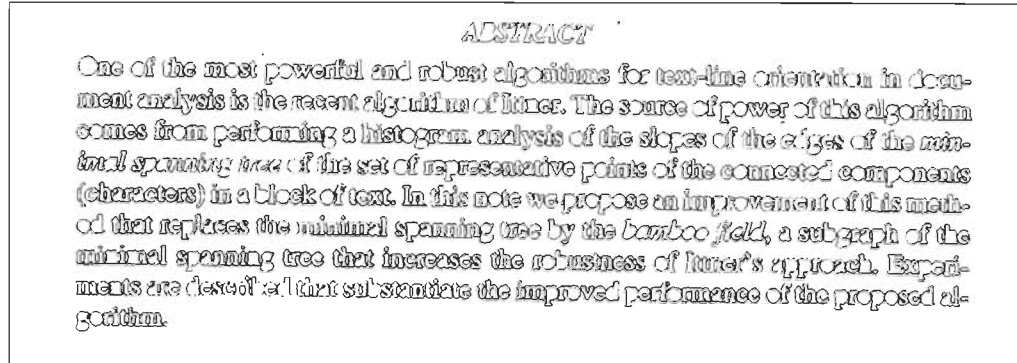


FIG. 2.4 – Résultat du deuxième algorithme de détection de contour

De façon générale, l'algorithme de contour proposé se décrit comme suit ; soit une image binaire où chaque pixel noir est indicé par la valeur 1 et chaque pixel blanc par la valeur 0. L'objectif visé est la construction pour chaque sous-domaine connexe de l'image, constitué de pixels noirs, d'un contour formé de pixels blancs. Le *contour* d'un sous-domaine connexe, est un chemin simple ou faiblement simple (voir plus loin), fermé et orienté positivement (antihoraire) et d'épaisseur égale à un pixel. De plus, le contour d'un sous-domaine de l'image correspondra à une chaîne de longueur minimale au sens de la connectivité à 8 voisins. L'orientation privilégiée, la représentation matricielle de l'image, est celle généralement utilisée, soit un axe des x positif orienté de haut en bas et un axe des y orienté positivement de gauche à droite de l'image.

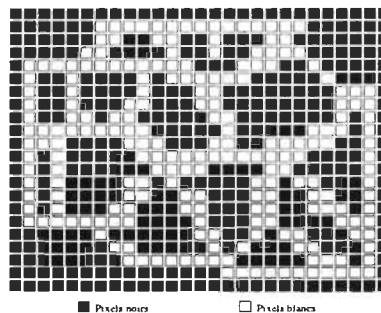


FIG. 2.5 – Sous-domaines binaires d'une image

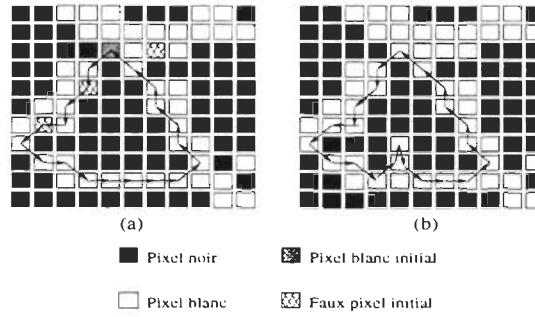


FIG. 2.6 – a) Contour simple, b) Contour faiblement simple

Pour chaque sous-domaine connexe D_K de l'image binaire on définit son contour C_K à l'aide d'une chaîne de pixels blancs,

$$[P_1, P_2, \dots, P_i, P_{i+1}, \dots, P_{L(K)}, P_{L(K)+1} = P_1],$$

où $L(K)$ est la longueur de la chaîne et P_i est le i -ième pixel de coordonnées entières (x_i, y_i) . Le pixel P_1 est appelé le pixel de départ du contour C_K . Un contour est dit *simple* si tous les pixels de la chaîne sont différents à l'exception des deux extrémités et un contour est dit *faiblement simple* s'il possède une ou plusieurs *adhérences simples* ou *composées*. On définit une *adhérence simple* comme étant un sous-chemin du non simple du contour, de largeur de un pixel, et tel que ce sous-chemin ou sous-chaîne n'entoure aucun domaine connexe constitué de pixels blancs. De plus, chaque pixel d'une adhérence simple est parcouru deux fois dans la chaîne (aller-retour), sauf un pixel que l'on appellera l'extrémité terminale de l'adhérence. Par conséquent, la longueur d'une adhérence simple est nécessairement un entier impair. On a la représentation suivante d'une adhérence simple ;

$$[P_k, P_{k+1}, \dots, P_t, \dots, P_{m-1}, P_m], \quad P_{t-i} = P_{t+i}, \quad 1 \leq i \leq m - t$$

avec $t - k = m - t$ et P_t est le pixel terminal de l'adhérence simple. On définira une *adhérence composée* comme étant la juxtaposition d'adhérences simples connectées entre elles à des pixels terminaux intérieurs à la chaîne décrivant l'adhérence composée.

On définira comme étant l'ensemble des *extrémités de départ potentielles* des

contours des sous-domaines $\{D_K, 1 \leq K \leq N\}$ d'une image binaire, les pixels blancs de coordonnées (i, j) tel que le pixel de coordonnées $(i+1, j-1)$ est un pixel blanc et le pixel $(i+1, j)$ est noir. Le déplacement de (i, j) à $(i+1, j)$ est donc obligatoirement d'indice de Freeman égal à 5, ce qui est capital dans les développements qui suivront. Notons R_1 l'ensemble initial des extrémités de départ potentiels et R_K l'ensemble des pixels potentiels de départ à l'étape de la construction du contour du K -ième domaine de l'image. Afin de rendre efficace l'algorithme de construction des contours, les coordonnées (i_K, j_K) du pixel de départ du contour du K -ième sous-domaine doivent vérifier l'assertion suivante ;

$$\min_j \left\{ \min_i \{ (i, j) \in R_K \} \right\}.$$

Lors de la construction d'une chaîne définissant le contour d'une cellule d'un sous-domaine D_K , les extrémités potentiels de départ rencontrés lors de la construction du contour de D_K seront éliminés de la liste. On repère facilement ces pixels en cours de construction puisque qu'un tel pixel et son successeur dans la chaîne de contour forme un couple dont le déplacement de Freeman est de valeur égale à 5. Enonçons maintenant les principes de base de la détection du contour d'un sous-domaine d'une image binaire.

Soit D_K un sous-domaine quelconque de l'image et notons $P_1 = (x_1, y_1)$ les coordonnées discrètes du premier pixel de départ potentiel du contour C_K , alors par construction le second pixel du contour orienté est le pixel blanc de coordonnées $P_2 = (x_2, y_2) = (x_1 + 1, y_1 - 1)$. On peut associer au premier arc du contour, que l'on note $a_1 = [P_1 P_2]$, l'indice de Freeman $\alpha_1 = 5$.

À partir de ce premier déplacement, il est possible de construire successivement et de façon univoque les arcs du contour de longueur minimale puisque chaque arc $a_k = [P_k, P_{k+1}]$, ($k \geq 2$) construit à partir de l'arc a_{k-1} doit respecter les deux principes ou hypothèses suivants :

- H_1 - L'arc $a_k = [P_k, P_{k+1}]$ d'indice de Freeman α_k est tel, et le pixel P_{k+1} est sélectionné, que le chemin minimise la longueur du contour au sens de la connectivité à 8 voisins ;

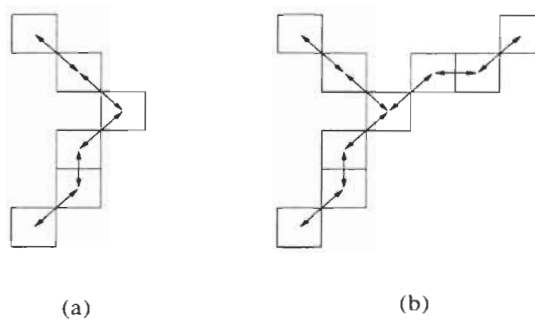


FIG. 2.7 – a) Adhérence simple, b) Adhérence composée

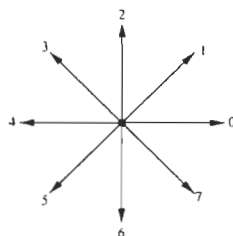


FIG. 2.8 – Directions de Freeman

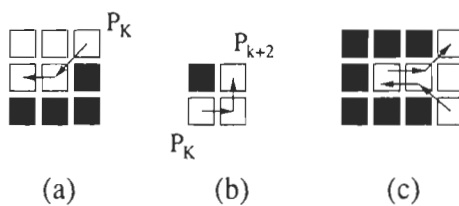


FIG. 2.9 – a) Paires admissibles, b) Paires non admissibles, c) Paires admissibles

- H_2 - La chaîne construite est antihoraire ou de façon équivalente, les pixels du contour sont des pixels blancs et les pixels à gauche de l'arc, par rapport au déplacement orienté, sont noirs.

Avec ces hypothèses on a, qu'étant donné un arc a_{k-1} , $k \geq 2$, il existe un et un seul pixel P_{k+1} , ou un seul arc orienté, réalisant simultanément les hypothèses H_1 et H_2 et le choix du pixel P_{k+1} s'effectue de façon univoque en fonction des indices des pixels voisins du pixel P_k . En effet, si on utilise les indices de Freeman pour identifier les déplacements (ou directions) définis par deux pixels successifs du contour alors les seules paires (i, j) de déplacements successifs admissibles sont celles présentées au tableau 2.1. Le premier indice i (ligne) indique la direction d'un arc a_k et l'indice j (colonne) indique la direction du déplacement admissible suivant, soit l'arc a_{k+1} qui minimise la longueur de la chaîne de contour.

$\begin{smallmatrix} j \\ i \end{smallmatrix}$	0	1	2	3	4	5	6	7
0	X	X			X	X	X	X
1	X	X	X	X		X	X	X
2	X	X	X	X			X	X
3	X	X	X	X	X	X		X
4	X	X	X	X	X	X		
5		X	X	X	X	X	X	X
6			X	X	X	X	X	X
7	X	X		X	X	X	X	X

TAB. 2.1 – Paires de déplacements successifs admissibles

Pour une direction d'arc a_k , on remarque qu'il existe au plus 7 directions adjacentes possibles de l'arc a_{k+1} , ce qui rend l'algorithme efficace. On note que certaines configurations représentent deux directions successives opposées lesquelles apparaissent naturellement lorsque des adhérences simples ou composées sont présentes dans un contour.

Comme conséquence de cette construction nous obtenons pour chaque sous-domaine

$D(K)$ un contour C_K de longueur $L(K)$ défini par une suite de pixels ;

$$P_1, P_2, \dots, P_k, P_{k+1}, \dots, P_{L(K)}, P_{L(K)+1} = P_1,$$

où l'arc $[P_1, P_2]$ est obligatoirement d'indice égal à 5 et P_1 est le premier pixel de départ potentiel rencontré dans la liste R_K . Le pixel P_1 et ceux qui correspondaient à des déplacements d'indice 5 dans le contour, sont de toute évidence retirés de l'ensemble des pixels de départ à l'étape suivante de construction d'un autre contour dans l'image. On peut associer au contour ainsi construit une suite de déplacements successifs, lesquels sont représentés dans la chaîne à l'aide de leur indice de Freeman,

$$\alpha_1, \alpha_2, \dots, \alpha_k, \dots, \alpha_{L(K)},$$

où α_k est la direction de l'arc $[P_k, P_{k+1}]$ pour $k = 1, \dots, L(K)$. Il a été démontré par Chalifour *et al* [7] qu'à partir de cette suite de pixels et de déplacements successifs admissibles, il était possible de calculer l'aire exacte d'un domaine discret.

2.2 Bounding boxes

L'application de la détection de contour sur une image originale \mathcal{I} génère une seconde image ne contenant que le contour de chacun des éléments calligraphiques du texte. L'étape suivante consiste à traiter cette seconde image à l'aide du *bounding boxes*. Pour chacun des objets repérés, les éléments calligraphiques, une boîte rectangulaire contenant tous les pixels du contour est construite. Les côtés des boîtes calculées sont orientés selon l'axe des x et selon l'axe des y .

Puisque l'orientation d'un texte est principalement dû aux caractères significatifs tels que les lettres du texte, nous voulons seulement conserver les boîtes contenant ces lettres. Il faut donc rejeter les boîtes qui contiennent les accents, les signes de ponctuation ainsi que les accentuations. Ces boîtes à rejeter ont de toute évidence une aire inférieure aux boîtes contenant les lettres. Un seuillage sur les aires des boîtes nous permettra de rejeter les boîtes non significatives et réduira à nouveau

la taille du problème.

Posons $B = \{b_1, b_2, \dots, b_n\}$, l'ensemble de toutes les boîtes obtenues à partir de l'algorithme du *bounding boxes*, $A = \{a_1, a_2, \dots, a_n\}$ les aires correspondantes et ψ un seuil sur la fonction d'aire. On pose comme seuil, la valeur ψ donnée par

$$\psi = \frac{1}{n\phi} \sum_{i=1}^n a_i ,$$

où ϕ est une valeur d'ajustement du seuil.

À partir de simulations effectuées dans [17], il est suggéré de poser $\phi = 2$. Si $\phi > 2$, nous n'éliminons pas suffisamment de boîtes et lorsque $\phi < 2$, nous perdons un trop grand nombre de boîtes. Le résultat du *bounding boxes* appliqué à l'image originale est présenté à la figure ci-dessous.

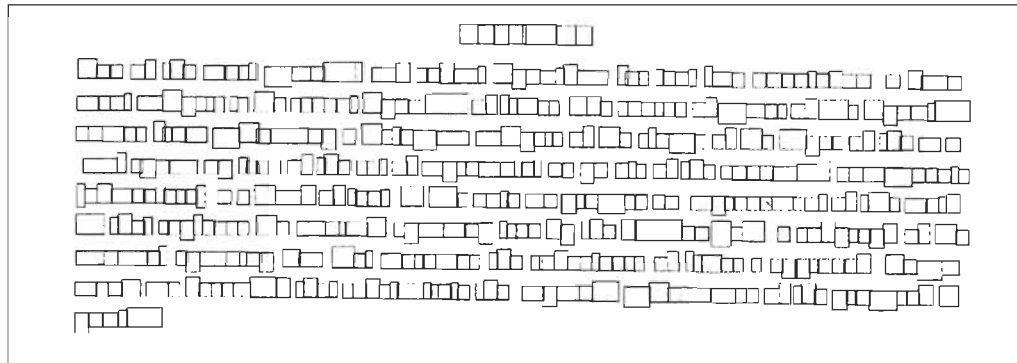
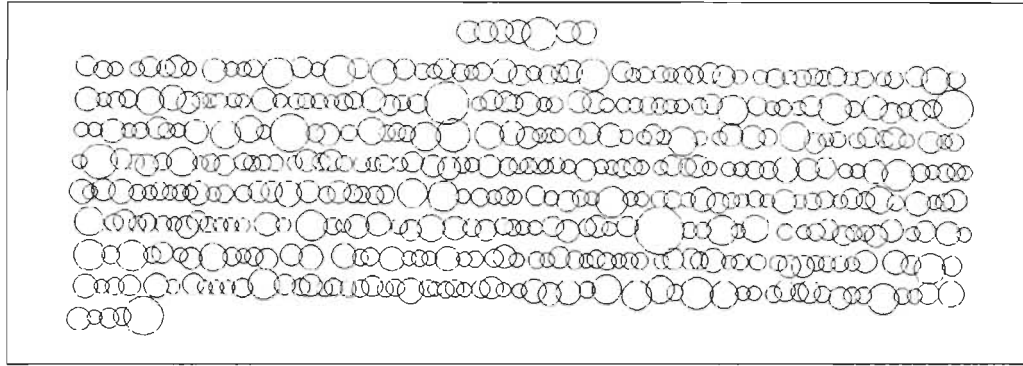


FIG. 2.10 – Application du bounding boxes sur l'image originale

Au lieu d'appliquer le *bounding boxes*, il est aussi possible d'utiliser des cercles de rayon minimal, lesquels contiennent tous les pixels du contour de chacune des lettres du texte. L'algorithme permettant de générer cet ensemble de cercles de rayon minimal contenant tous les points d'un certain ensemble P est l'algorithme du *MiniDisc* [8]. Nous avons appliqué l'algorithme du *MiniDisc* sur le texte de l'image originale et nous obtenons le résultat présenté à la figure 2.11.

FIG. 2.11 – Application de l'algorithme du *MinDisc* sur l'image originale

Cependant, si une lettre possède un contour de n pixels, le *bounding boxes* s'exécute toujours en temps $O(n)$ tandis que l'algorithme de *MiniDisc* est de l'ordre de $O(n)$ en moyenne seulement. On conclue que le *bounding boxes* exige, en moyenne, moins d'opérations élémentaires que le *MiniDisc*, donc le traitement sera plus rapide. Pour cette raison et puisque les résultats des simulations comparant ces deux approches n'ont pas montré de différences significatives concernant l'approximation de l'angle d'orientation du texte, le *bounding boxes* a été retenu dans notre méthodologie. Considérons maintenant l'étape de construction des points représentatifs après le seuillage sur les valeurs des aires des boîtes.

2.3 Ensemble de points représentatifs du texte

Soit $B = \{b_1, b_2, \dots, b_n\}$ l'ensemble des n boîtes obtenues à l'étape précédente, où chacune des b_k est représentée par ses quatre sommets (figure 2.12). Soit une boîte $b_i \in B$ et $i \in \{1, \dots, n\}$, alors le centre géométrique de la boîte b_i est le point p_i représentatif de la boîte b_i , lequel est calculé de la façon suivante :

$$p_{i_x} = \frac{b_{i_{x1}} + b_{i_{x2}}}{2}$$

$$p_{i_y} = \frac{b_{i_{y1}} + b_{i_{y3}}}{2}$$

$$p_i = (p_{i_x}, p_{i_y})$$

où, p_{i_x} est la coordonnée en x du point et p_i et p_{i_y} sa coordonnée en y .

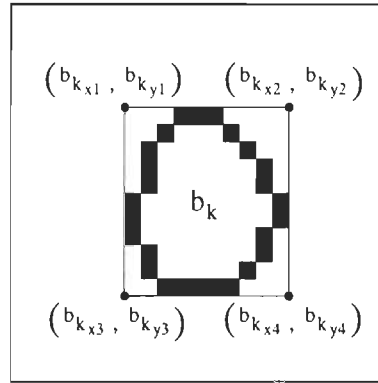


FIG. 2.12 – Représentation d'un boîte

L'ensemble P des centres géométriques constitue l'ensemble des points représentatifs du texte et sert de point de départ pour la *méthode d'Ittner*, le *Bamboo field* et les deux nouvelles approches géométriques. La figure 2.13 montre l'ensemble de points obtenus à partir de l'image originale en ne retenant que les centres géométriques

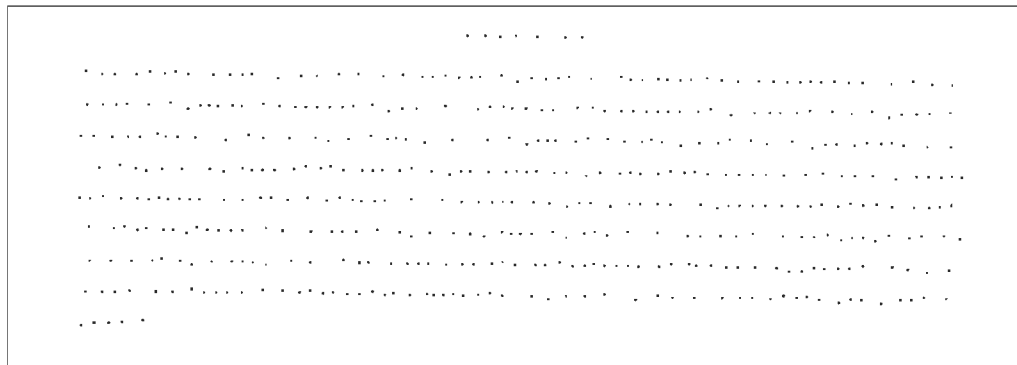


FIG. 2.13 – Ensemble de points représentatifs du texte

L'ensemble P des points représentatifs du texte suggèrent dans leur alignement et leur distribution dans une fenêtre de texte une orientation de l' AOT . Il reste donc à développer une méthode efficace pour regrouper en segments les points représentatifs, lesquels segments nous permettront d'estimer l' AOT . Dans les prochains chapitres, nous analyserons deux méthodes connues pour ce problème et nous proposerons deux nouvelles approches.

Chapitre 3

Ittner et le Bamboo field

Nous avons vu au chapitre précédent les trois étapes de pré-traitement de données textuelles numérisées en prévision de l'application de quatre méthodes géométriques pour l'estimation de l'orientation d'un texte. Il s'agit des méthodes d'Ittner, du *Bamboo field*, d'une nouvelle approche que nous appellerons *Bamboo field amélioré* (*BFA*) et de deux méthodes basées sur le *nearest neighbor graph* (*NNG* et *NNGA*). Ce chapitre présente les deux premières approches et les trois autres méthodes seront exposées au chapitre 4.

Ces deux approches sont similaires dans le sens qu'elles sont toutes deux basées sur un même graphe de proximité, soit l'arbre sous-tendant minimal. Nous verrons aussi en quoi le *Bamboo field* constitue une amélioration de la méthode d'Ittner. Décrivons ces deux méthodes.

3.1 Méthode d'Ittner

Tel que mentionné précédemment, la méthode d'Ittner [13] utilise les trois étapes de pré-traitement présentées au chapitre précédent. À ce stade, nous avons à traiter

un ensemble de n points $P = \{p_1, p_2, \dots, p_n\}$ représentatifs du texte original et dont l'alignement suggère une orientation générale du texte.

Pour estimer l'angle d'orientation du texte, Ittner propose de relier les points de P à l'aide d'un graphe de proximité. Il s'agit de l'arbre sous-tendant minimum (*ASM*), c'est-à-dire l'arbre de longueur minimale reliant tous les points de P .

L'algorithme que nous avons utilisé pour calculer l'*ASM* a été proposé par R. Prim [18] et porte d'ailleurs le nom de son inventeur. Pour calculer l'*ASM*, nous désirons connecter chacun des $P_i \in P$ entre eux de telle sorte que la somme des segments soit de longueur minimale. Dans un premier temps, nous sélectionnons le segment de longueur minimum et nous l'ajoutons dans l'*ASM*. Puis, successivement, nous ajoutons les segments de longueur minimum qui sont adjacents aux sommets déjà présents dans l'arbre, tout en évitant de créer des cycles avec les segments appartenant à l'arbre. Définissons quelques notions avant de présenter l'algorithme de Prim plus en détail.

Soit $P = \{p_1, \dots, p_n\}$ l'ensemble des points représentatifs du texte original et $S = \{s_1, \dots, s_m\}$ l'ensemble des segments, triés par ordre croissant, obtenus par la construction du graphe complet de P . Nous définissons un graphe complet d'un ensemble P comme un graphe où pour chaque $p_i \in P$, $k \in \{1, \dots, n\}$ et $i \neq k$, il existe un segment $\overline{p_i p_k}$. Voici maintenant l'algorithme de Prim sous forme de pseudo-code :

Prim (Entrée : P , Sortie : ASM)

$S =$ graphe complet de P où les s_i sont triés

$ASM = \phi$

$P_{in} = \phi$

$i = 1$

Tant que $P_{in} \neq P$

$s_c = s_i = \overline{s_{i_1} s_{i_2}}$

Si $s_{i_1} \notin P_{in}$ ou $s_{i_2} \notin P_{in}$

on ajoute s_c dans ASM

$P_{in} = P_{in} \cup \{s_{i_1}, s_{i_2}\}$

Fin Si

$i++$

Fin Tant que

Fin Prim

Cet algorithme permet de construire un arbre, de longueur minimal, reliant par un chemin simple, toutes paires de points de P . Cet arbre est l'arbre sous-tendant minimal, un graphe de proximité. Pour plus de détails sur cet algorithme, on peut consulter les références [5, 20].

Dans la méthode d'Ittner, tous les segments de l' ASM sont pris en compte pour approximer l'angle d'orientation du texte. La figure 3.1 représente l' ASM correspondant à l'image originale. Comme on le constate, la méthode d'Ittner fait apparaître des segments verticaux, qui plus est, ces segments ne fournissent pas une information pertinente sur l'orientation du texte. Ces segments non-représentatifs de l'AOT correspondent généralement à des segments qui relient deux lignes consécutives du texte. La présence de ces liens est normale car le calcul de l' ASM oblige le graphe à être connexe. De plus, ces segments indésirables peuvent également relier une lettre à un accent, à un signe de ponctuation ou à tout autre caractère n'ayant pas été supprimé lors de l'étape de seuillage appliquée aux boîtes (*bounding boxes*).

Pour approximer l'angle d'orientation du texte, la moyenne ou la médiane sur les segments est utilisée. Donc, puisque tous les segments de l'*ASM* sont utilisés pour estimer l'*AOT*, un biais sera introduit dans le calcul de l'*AOT* par la présence de certains segments non-représentatifs générés par l'*ASM*. Le nombre de segments non-représentatifs peut être plus ou moins élevé, tout dépendant de la langue dans laquelle est écrit le texte. Nous verrons dans la section 4.1 du chapitre suivant, la méthodologie utilisée pour estimer l'angle d'orientation du texte.

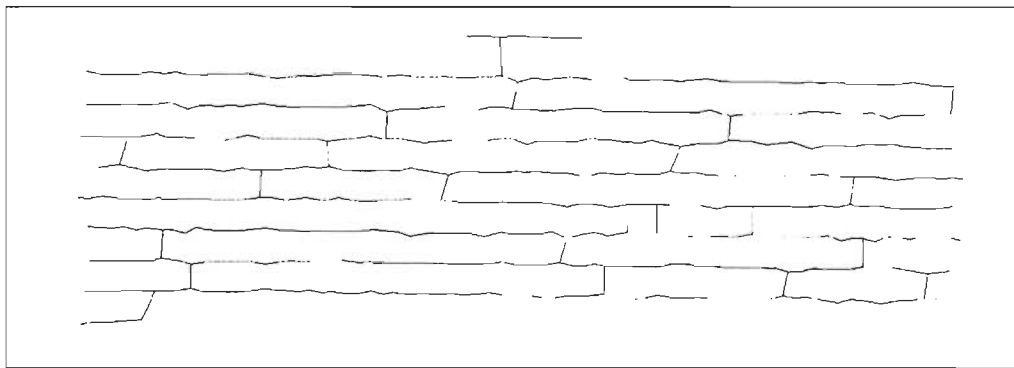


FIG. 3.1 – Arbre sous-tendant minimal du texte original

3.2 Bamboo field

La méthode du *Bamboo field* [17] constitue une amélioration de la méthode d'Ittner. Le *Bamboo field* permet de générer un graphe de proximité pour lequel certains segments non-représentatifs de l'*AOT* ont été éliminés. La figure 3.2 montre un segment $\overline{p_1 p_2}$ non-représentatif de l'*AOT* obtenu lors du calcul de l'*ASM* où L_1 et L_2 représentent deux lignes successives d'un texte.

Il est connu [17] que la majorité de ces segments non-représentatifs de l'*AOT* correspondent à des segments adjacents à des sommets (points représentatifs de l'*ASM*) ayant un degré supérieur ou égale à 3, le degré d'un sommet p_i étant égal au nombre de segments pour lesquels p_i est l'une des deux extrémités. À la figure 3.3, le sommet p_1 est de degré 5 tandis que les autres sommets sont de degré 1. Dans la méthode

du *Bamboo field*, on tient compte de cette observation afin d'éliminer de l'*ASM* de tels segments.

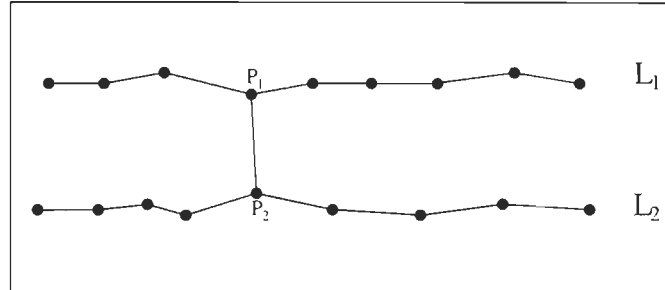


FIG. 3.2 – Exemple d'un segment non-représentatif généré par l'*ASM*

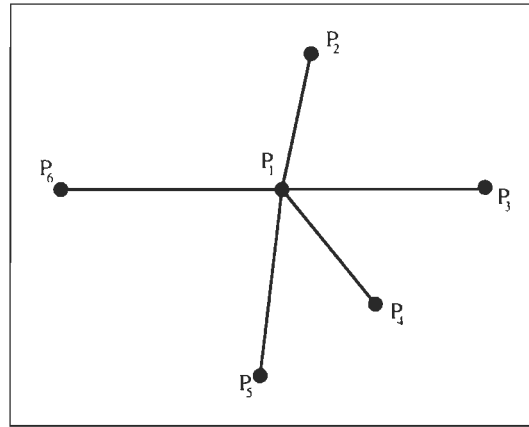
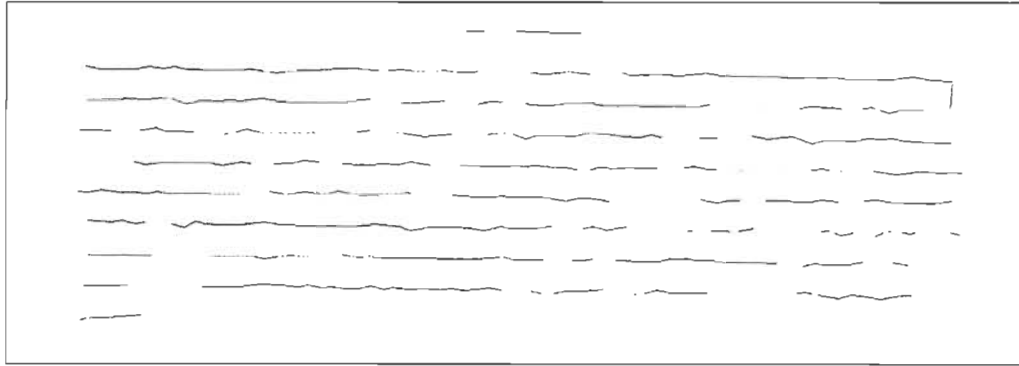


FIG. 3.3 – Exemple d'un sommet de degré 5

Supposons l'*ASM* construit à partir de l'algorithme d'Ittner, alors l'objectif visé est d'appliquer un nouveau critère permettant de rejeter les segments non-représentatifs de l'*AOT*. Un premier critère serait d'éliminer tous les segments adjacents à un sommet de degré supérieur ou égale à trois. Le résultat obtenu est un ensemble de sous-arbres correspondant à des chaînes polygonales, quasi-parallèles. Cette approche produit de bons résultats, quoiqu'elle ne permet pas l'élimination de tous les segments indésirables. Nous avons appliqué la méthode du *Bamboo field* au texte original et le résultat est présenté à la figure 3.4.

Après épuration des segments par l'application de ce critère, l'estimation de l'*AOT*

FIG. 3.4 – Graphe obtenu après l'exécution du *bamboo field*

est faite à partir des segments restants de l'arbre. Nous préciserons le calcul d'angle au chapitre suivant lors de la présentation des deux nouvelles approches algorithmiques du calcul de l'*AOT*. Une première méthode, le *Bamboo field amélioré*, sera basée sur une modification du *Bamboo field* et les deux autres seront basées sur la construction d'un autre graphe de proximité : le *nearest neighbor graph*.

Chapitre 4

Nouvelles approches

Au chapitre précédent, nous avons présenté deux méthodes basées sur l'arbre sous-tendant minimal (*ASM*), soit la méthode d'Ittner et la méthode du *Bamboo field*. Dans ce chapitre, nous proposons une nouvelle méthode, aussi basée sur l'*ASM*, laquelle constituera une amélioration de la méthode du *Bamboo field*, et par le fait même, de la méthode d'Ittner. À la deuxième section, deux autres approches basées sur un autre graphe de proximité, le *nearest neighbor graph*, seront proposées pour l'approximation de l'angle d'orientation d'un texte.

4.1 Bamboo field amélioré

Soit l'ensemble des n segments $S = \{s_1, s_2, \dots, s_n\}$ obtenus suite à l'application du *Bamboo field*, c'est-à-dire après l'extraction de l'arbre sous-tendant minimal et l'application du critère de rejet des segments adjacents aux sommets de degré supérieur ou égale à trois.

Comme nous avons pu le constater, des segments non significatifs sont encore présents lors du calcul de l'*ASM* et de l'application de la méthode du *Bamboo field*. Idéalement,

nous désirons conserver un ensemble de chaîne de segments, où chacune des chaînes, composée de deux ou de plusieurs segments, forment une ligne la plus droite possible, c'est-à-dire tel que l'angle entre deux segments adjacents est le plus près possible de 180° . L'objectif est d'introduire un second critère qui permettra l'élimination de toute paire de segments adjacents dont l'angle est plus grand qu'un certain angle critique α , lequel est à déterminer (seuillage).

Le critère de rejet s'énonce comme suit :

Pour chaque paire de segments adjacents s_i et s_j de S
 $\theta = \text{angle formé entre } s_i \text{ et } s_j$
 Si $\theta < \alpha$
 s_i et s_j sont supprimés de S
 Fin Si
 Fin Pour

Évidemment, l'angle calculé entre deux segments est l'angle, inférieur à 180° , calculé à partir du produit scalaire dans R^2 (voir figure 4.1). Plus l'angle critique α est près de 180° , plus un grand nombre de segments seront éliminés. À partir de simulations nous avons estimé l'angle α à 170° .

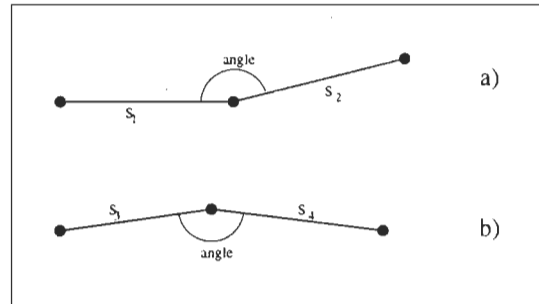


FIG. 4.1 – Calcul de l'angle entre deux segments

Une *matrice d'adjacence* est utilisée comme structure de donnée pour conserver les segments d'un graphe. La matrice d'adjacence d'un arbre de n points, est une matrice $A = [a_{ij}]$ de dimension $n \times n$, tel que $a_{ij} = 1$, s'il existe un segment reliant

les points P_i et P_j , sinon $a_{ij} = 0$. À la figure 4.2, nous avons à gauche, un arbre et à droite sa représentation sous forme d'une matrice d'adjacence. Si on applique le critère proposé sur l'angle entre deux segments consécutifs à cet exemple, les segments qui seront conservés sont $\overline{P_2P_3}$ et $\overline{P_3P_4}$. Cependant, si nous considérons un ordre quelconque des 4 points et que nous appliquons directement ce critère sans utiliser de structure particulière, il se pourrait que seulement le segment $\overline{P_3P_4}$ soit retenu, ce qui nous fait perdre de l'information sur l'orientation du texte. Le fait d'utiliser ce type de structure permet d'éviter ce genre de situation.

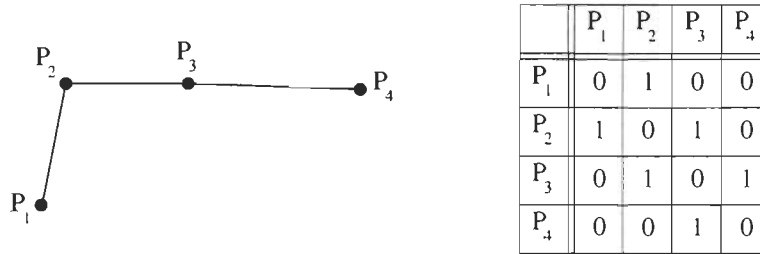


FIG. 4.2 – Matrice d'adjacence

À la figure 4.4 on retrouve le résultat de l'application de ce critère à l'image originale avec un angle critique α de 170° . Nous pouvons comparer ce résultat avec celui de la méthode du *Bamboo field* que l'on trouve à la figure 4.3.

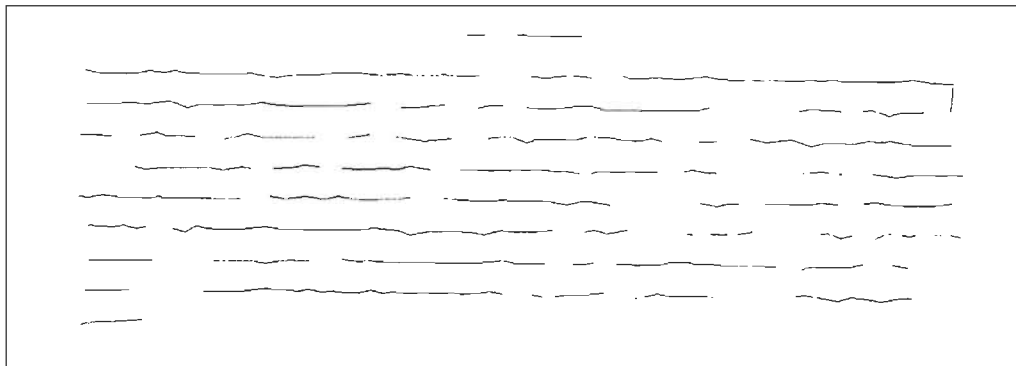


FIG. 4.3 – Application du Bamboo field

L'application de ce nouveau critère permet donc de créer un nouvel ensemble de segments à partir desquels nous estimerons l'angle d'orientation du texte. La section qui suit traite de l'estimation de l'AOT.

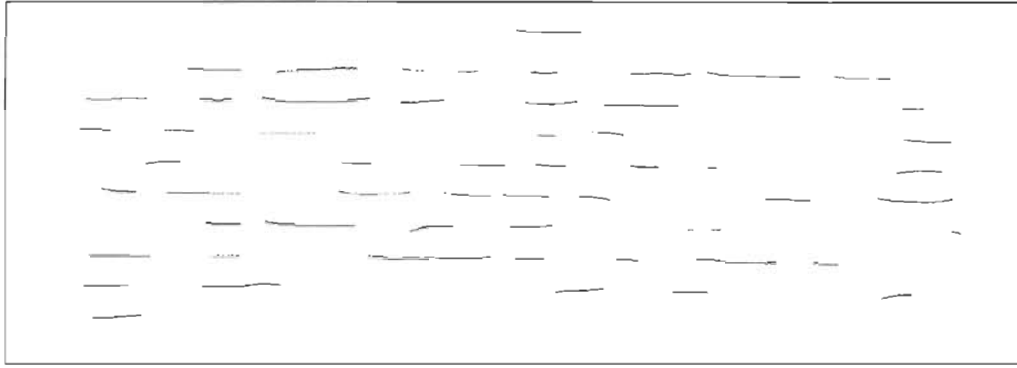


FIG. 4.4 – Application du Bamboo field amélioré

4.1.1 Estimation de l'AOT

Soit $S = \{s_1, s_2, \dots, s_n\}$ l'ensemble des segments résultant des méthodes du *Bamboo field amélioré*, d'Ittner, du *Bamboo field* ou du *Nearest neighbor graph*. Pour chacun des segments s_k de S , notons a_k l'angle que s_k forme avec l'horizontal en orientant s_k de telle façon que son origine est le sommet à gauche du segment.

Notons $\vec{a} = \{a_1, a_2, \dots, a_n\}$. le vecteur ordonné des angles des segments, lequel correspond à un vecteur d'échantillonnage de l'AOT et qui servira à son estimation.

Deux méthodes différentes peuvent être utilisées pour l'approximation de l'AOT. Une première consiste à estimer l'AOT par la moyenne des a_k :

$$\bar{a} = \frac{\sum_{i=1}^n a_i}{n},$$

et la seconde alternative est d'utiliser la médiane de \vec{a} comme estimateur. Dans le cas de la médiane, on doit trier les s_k et on pose la médiane $\tilde{a} = a_{n/2}$. Ces deux approches ont été utilisées pour l'estimation de l'angle d'orientation d'un texte, pour les quatre méthodes présentées dans ce mémoire. Dans les chapitres suivants, nous analyserons à partir de simulations la précision de ces deux estimateurs afin de déterminer lequel est le plus efficace.

4.2 Nearest neighbor

Les trois approches exposées jusqu'à présent utilisent le même graphe de proximité, soit l'arbre sous-tendant minimal. Cependant, il existe un autre graphe de proximité qui pourrait être utilisé pour l'estimation de l'angle d'orientation d'un texte, il s'agit du *nearest neighbor graph* (*NNG*). Comme son nom l'indique, le *NNG* [16] d'un ensemble de points P est le graphe du plus proche voisin de chacun des points de P .

Avant de définir plus en détail ce qu'est le *NNG* d'un ensemble de points, définissons tout d'abord ce qu'est le plus proche voisin d'un point. Soit un ensemble de n points $P = \{p_1, p_2, \dots, p_n\}$, on dira que le point p_b est le plus proche voisin de p_a , si et seulement si

$$|p_a - p_b| \leq \min_{a \neq c} \{|p_a - p_c|\},$$

où $p_c \in P$ et $|p_a - p_c|$ représente la distance euclidienne entre les points p_a et p_b .

Notons que la relation *être le plus proche voisin* n'est pas symétrique, c'est-à-dire, que si p_b est le plus proche voisin de p_a , alors p_a n'est pas nécessairement le plus proche voisin de p_b . La figure 4.5 montre un exemple illustrant cette situation. Dans ce cas, p_2 qui est le plus proche voisin de p_1 tandis que le plus proche voisin de p_2 n'est pas p_1 , mais plutôt p_3 .

Le *NNG* d'un ensemble P est donc, un graphe non-orienté, tel que tous les points de P sont connectés et tel que pour tout arc $\overline{p_i p_j}$ du graphe, p_i est le plus proche voisin de p_j ou p_j est le plus proche voisin de p_i . De toute évidence, le *NNG* est non-connecté, étant donné qu'*être le plus proche voisin* est une relation non nécessairement symétrique pour un ensemble de points quelconque. On montre à la figure 4.6 le *nearest neighbor graph* obtenu à partir de l'ensemble des points représentatifs de l'image originale du chapitre 2. Tous les segments résultant de l'application du *nearest neighbor graph* sont utilisés pour l'approximation de l'*AOT*.

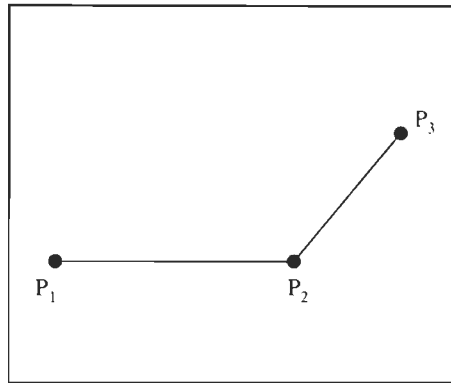
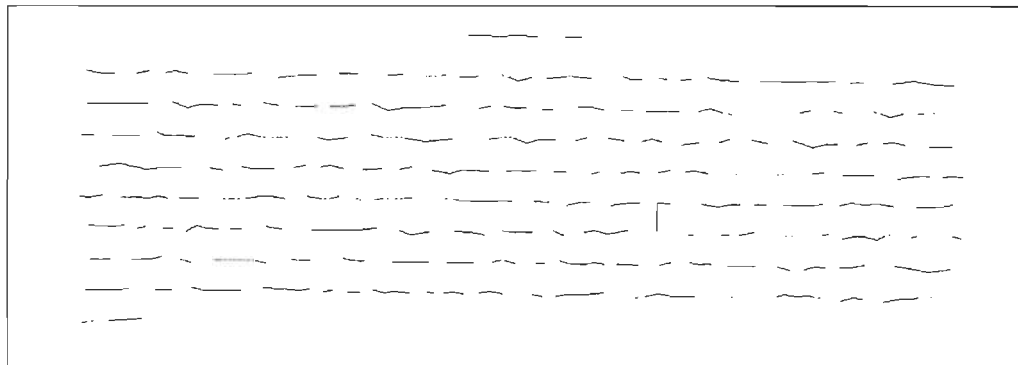


FIG. 4.5 – Exemple de la non-symétrie de la relation du plus proche voisin

FIG. 4.6 – Application du *NNG* à l'image originale

Nous pouvons améliorer les résultats obtenus avec le *NNG* en appliquant les deux critères de seuillage utilisés dans le cas de la méthode *BFA*. La figure 4.7 montre le résultat de ces critères sur le texte original. Nous appellerons cette méthode *NNGA* pour signifier la méthode du *nearest neighbor améliorée*.

À partir des approches exposées jusqu'à maintenant, qui à première vue semblent équivalentes, nous procéderons à une série de simulations afin de comparer ces méthodes, ce qui fera l'objet des deux prochains chapitres de ce mémoire.

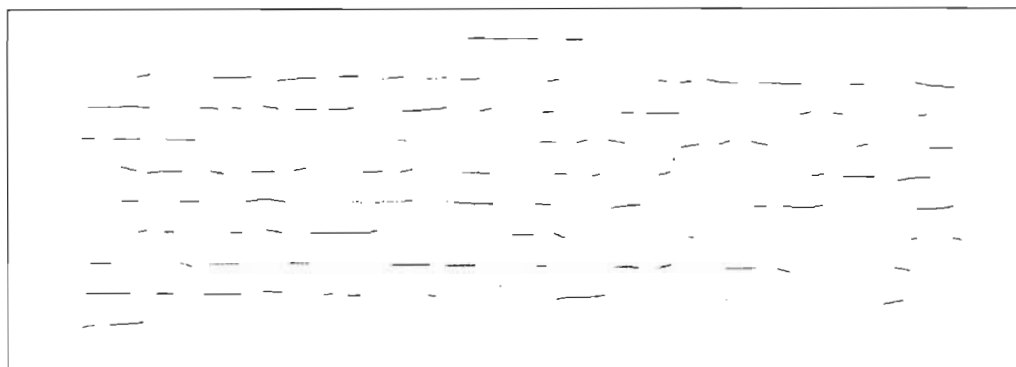


FIG. 4.7 – Application du *NNGA* à l'image originale

Chapitre 5

Comparaison

Nous avons vu dans les chapitres précédents cinq approches servant à la détection de l'angle d'orientation d'un texte, mais aucune comparaison entre ces méthodes n'a été faite en regard de la qualité de l'estimation de l'*AOT*. Lorsque l'on observe les images obtenues après traitement, nous pouvons porter un jugement subjectif a priori sur l'efficacité des méthodes. De prime abord, la méthode du *Bamboo field amélioré* devrait conduire à de meilleurs résultats si l'on tient compte de l'orientation qu'ont les segments avant l'approximation de *AOT*. Évidemment, nous ne pouvons affirmer formellement que tel est le cas sans que des simulations sur plusieurs textes différents n'aient été effectuées. Ce chapitre présente un ensemble de simulations qui ont été faites pour chacune des méthodes et l'analyse comparative des résultats.

Lorsque que nous voulons comparer des méthodes entre elles afin de déterminer laquelle est la plus efficace, dans notre cas celle qui estime le mieux l'*AOT*, il faut tenir compte des deux critères suivants :

- la biais,
- la précision.

La qualité de ces deux critères est garante de la qualité d'un estimateur pouvant fournir une bonne approximation d'un paramètre.

Notons θ , la valeur théorique de l'*AOT* et par $\hat{\theta}$, l'estimateur de θ . Un estimateur $\hat{\theta}$ est sans biais si la moyenne de sa distribution d'échantillonnage est égale à la valeur θ , c'est-à-dire si

$$E(\hat{\theta}) = \theta.$$

Si l'estimateur $\hat{\theta}$ est biaisé alors son biais est mesuré par l'écart suivant :

$$biais = E(\hat{\theta}) - \theta.$$

En plus d'avoir la propriété d'être sans biais, un estimateur se doit d'être précis, c'est-à-dire il faut que sa variance se rapproche le plus possible de 0. Ainsi, si $\hat{\theta}_1$ et $\hat{\theta}_2$ sont deux estimateurs sans biais de θ , l'estimateur $\hat{\theta}_1$ est le plus efficace si

$$Var(\hat{\theta}_1) < Var(\hat{\theta}_2)$$

ou encore, si

$$\sigma_{\hat{\theta}_1} < \sigma_{\hat{\theta}_2}.$$

Ceci nous indique tout simplement que l'estimateur $\hat{\theta}_1$ est plus concentré autour de θ que $\hat{\theta}_2$.

Afin de comparer les méthodes proposées dans ce mémoire, nous avons utilisé des estimations par intervalles de confiance. Ces derniers nous permettent de vérifier à la fois le biais et la précision des estimateurs. Étant donné que pour approximer l'*AOT* nous désirons utiliser la moyenne et la médiane, il a donc fallu employer deux types d'intervalles de confiance, un premier défini pour la moyenne et un autre pour la médiane.

Avant de définir les intervalles de confiance pour chacun de ces deux paramètres et afin de bien saisir les simulations qui ont été faites, posons les notations suivantes. Premièrement considérons un échantillon de n textes notés par

$$\vec{T} = \{T_1, T_2, \dots, T_n\}$$

dont nous connaissons les valeurs réelles d' AOT :

$$\vec{A} = \{A_1, A_2, \dots, A_n\}.$$

Pour chacune des méthodes, lorsque nous désirons approximer l' AOT d'un texte donné T_k , nous utilisons un ensemble de m segments formant un angle a_{k_i} avec l'horizontal. On représente les angles des segments d'un texte T_k par le vecteur :

$$\vec{a}_k = \{a_{k_1}, a_{k_2}, \dots, a_{k_m}\}.$$

Notons \bar{a}_k et \tilde{a}_k la moyenne et la médiane des a_{k_i} d'un texte T_k , respectivement donné par :

$$\bar{a}_k = \frac{1}{m} \sum_{i=1}^m a_{k_i}$$

et

$$\tilde{a}_k = a_{k_{\frac{m}{2}}}.$$

Pour chacun des textes T_k de \vec{T} , nous calculons la moyenne, la médiane ainsi que les intervalles de confiance sur la moyenne et sur la médiane.

La définition d'un intervalle de confiance consiste à calculer un intervalle pour lequel il est vraisemblable que la valeur théorique ou réelle de l' AOT s'y retrouve.

L'intervalle de confiance pour l' AOT est défini par deux limites auxquelles est associée une certaine probabilité, fixée à l'avance et aussi élevée qu'on le désire, de contenir la vraie valeur de l' AOT :

$$P(L_i \leq A \leq L_s) = P(\hat{A} - k \leq A \leq \hat{A} + k) = 1 - \alpha$$

où

- A : valeur réelle de l' AOT ,
- \hat{A} : estimateur de l' AOT ,
- L_i : limite inférieure de l'intervalle,
- L_s : limite supérieure de l'intervalle,
- $1 - \alpha$: la probabilité associée à l'intervalle de contenir la valeur réelle de l' AOT ,
- k : la marge d'erreur qui tient compte des fluctuations de \hat{A} et de la probabilité $1 - \alpha$.

Nous avons utilisé un seuil de 0.05. Nous pouvons utiliser ces intervalles de confiance en supposant que la distribution des angles d'un texte T_k suit une loi normale. La figure 5.1 montre l'histogramme de distribution des angles du texte contenu dans la figure 5.2.

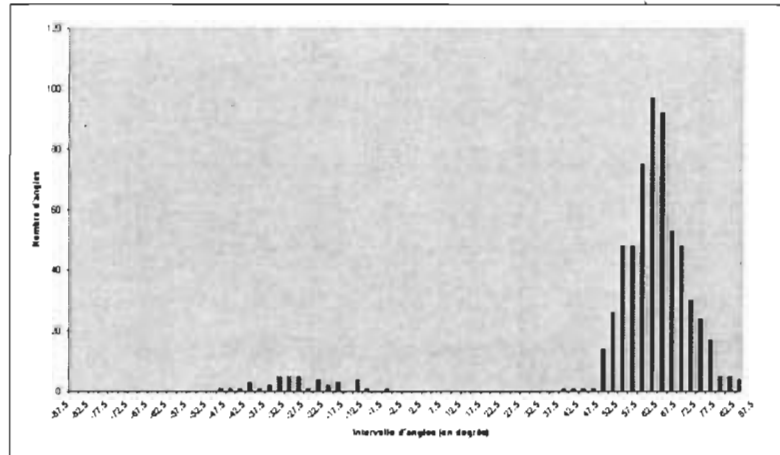


FIG. 5.1 – Histogramme de distribution des angles

Considérons le cas d'un texte T_k , alors l'intervalle de confiance sur la moyenne se calcule de la façon suivante :

$$[L_{ki}, L_{ks}] = [\bar{a}_k - M_k, \bar{a}_k + M_k] = [\bar{a}_k - z_{\alpha/2} \frac{\sigma_k}{\sqrt{n}}, \bar{a}_k + z_{\alpha/2} \frac{\sigma_k}{\sqrt{n}}],$$

où $z_{\alpha/2} = 1.96$ est valeur prise dans la table de la loi normale pour un seuil $\alpha = 0.05$, M_k est à la marge d'erreur, et

$$\sigma_k = \frac{1}{m-1} \sum_{i=1}^m (\bar{a}_k - a_{ki})^2.$$

Si \vec{a}_k est un vecteur trié, alors l'intervalle de confiance sur la médiane est le suivant :

$$[L_{ki}, L_{ks}] = [a_{kv}, a_{kw}],$$

où

$$v = \lfloor \frac{m}{2} - 1.96 \frac{\sqrt{n}}{2} \rfloor$$

et

$$w = \lceil \frac{m}{2} + 1.96 \frac{\sqrt{n}}{2} \rceil.$$

Dans ce cas, la marge d'erreur M_k est égale à $\frac{L_s - L_i}{2}$. Ces paramètres statistiques sont décrit plus en détails dans [2].

Pour les simulations, nous avons utilisé un échantillon de 100 textes, soit 50 textes français et 50 anglais. Les 50 textes, pour chacune des deux langues, correspondent à 10 textes différents auxquels 5 rotations ont été appliquées, soit -60° , -30° , 0° , 30° et 60° . Ces rotations ont été choisies arbitrairement puisque la qualité des résultats doit être indépendante de l'angle de rotation.

Cet échantillon a donc été soumis à chacune des cinq approches, et ce avec l'AOT estimé avec les deux estimateurs, la moyenne et la médiane. Le tableau ci-dessous

	Moyenne		Médiane	
	biais	marge d'erreur	biais	marge d'erreur
Ittner	4.07°	1.75°	0.56°	0.62°
Bamboo field	0.46°	0.87°	0.19°	0.56°
BFA	0.24°	0.48°	0.20°	0.45°
NNG	0.82°	1.05°	0.40°	0.52°
NNGA	0.66°	0.85°	0.43°	0.64°

TAB. 5.1 – Comparaison entre les 5 différentes approches

illustre les résultats obtenus pour chacune des méthodes et pour l'ensemble des 100 textes.

Le biais ϵ correspond au biais absolu moyen :

$$\epsilon = \frac{1}{100} \sum_{i=1}^{100} |A_i - \hat{A}_i|$$

où \hat{A}_i est l'estimateur de l'*AOT* à partir de la moyenne ou de la médiane et A_i est la valeur réelle de l'*AOT*. Les marges d'erreur correspondent à la marge d'erreur moyenne sur les 100 textes.

Ces résultats montrent clairement que dans le cas de la moyenne, la méthode du *Bamboo field amélioré* est la plus précise. Le biais et la marge d'erreur sont environ deux fois plus faibles que la méthode du *Bamboo field*, qui occupe la deuxième place. Les trois autres approches, soit *Ittner*, *NNG* et *NNGA* sont comparativement moins performantes. Ceci est prévisible car la moyenne utilise tous les segments pour le calcul l'intervalle de confiance et les méthodes d'*Ittner*, du *Bamboo field* et du *NNG* produisent des valeurs d'angle aberrantes pouvant détériorer la qualité des résultats.

On remarque que dans le cas de la médiane, les résultats générés par les méthodes du *Bamboo field* et du *Bamboo field amélioré* sont similaires, soit une différence du biais de l'ordre de 0.01°. De plus, la différence entre les marges d'erreur est négligeable. Contrairement à la moyenne, la médiane ne tient pas compte des valeurs

aberrantes, ce qui explique les résultats similaires entre le *Bamboo field* et le *Bamboo field amélioré*, car les segments non significatifs du *Bamboo field* ne sont pas utilisés pour le calcul de l'*AOT*.

D'après les données de ce tableau, il semble donc préférable d'utiliser la médiane plutôt que la moyenne pour approximer l'*AOT* car les biais et les marges d'erreur sont nettement plus faibles, et ce pour les quatre approches.

Les figures 5.3, 5.4, 5.5, 5.6 et 5.7 montrent les résultats (intervalles de confiance sur la moyenne et sur la médiane) obtenus à l'aide des quatres méthodes appliquées au texte anglais présenté à la figure 5.2.

Autre estimateur de l'*AOT*

Les estimateurs de l'*AOT* présentés à la section 4.1.1 donnent de bons résultats, mais ces derniers ne sont pas optimaux. En effet, dans le cas d'une distribution normale, le meilleur estimateur est la moyenne des angles pondérée par la longueur de chaque segment. Le modèle statistique correspondant est le suivant ; pour chaque segment s_i , d'extrémités (x_{i1}, y_{i1}) et (x_{i2}, y_{i2}) , on suppose que la pente, correspondant à l'*AOT*, est égale à β , d'où

$$Y_{ij} = \alpha_i + \beta X_{ij} + \epsilon_{ij},$$

où α_i est l'intercept avec l'axe des ordonnées et ϵ_{ij} est le terme d'erreur. Les erreurs sont considérées indépendantes et de même variance σ^2 . L'angle d'estimation de l'*AOT* n'est autre que la tangente inverse de β . Le meilleur estimateur b de β , si l'on suppose que les erreurs sont distribués selon une loi normale, est donné par

$$b = \frac{\sum_{i=1}^n (y_{i2} - y_{i1})(x_{i2} - x_{i1})}{\sum_{i=1}^n (x_{i2} - x_{i1})^2} = \frac{\sum_{i=1}^n b_i (x_{i2} - x_{i1})^2}{\sum_{i=1}^n (x_{i2} - x_{i1})^2},$$

où $b_i = (y_{i2} - y_{i1})/(x_{i2} - x_{i1})$ est la pente du segment s_i . Nous pouvons également

donner un intervalle de confiance pour β . La variance de l'estimateur est donnée par

$$\sigma^2(b) = \frac{2\sigma^2}{(x_{i2} - x_{i1})^2},$$

et cette variance peut être estimée par

$$s^2(b) = \frac{\sum_{i=1}^n (b_i - b)(x_{i2} - x_{i1})^2}{\sum_{i=1}^n (x_{i2} - x_{i1})^2}.$$

Finalement, l'intervalle de confiance de b pour un niveau $100(1 - \alpha)\%$ est donné par $\beta = b \pm s(b)t_{\alpha/2, n-1}$. On peut ainsi construire un intervalle de confiance pour l'angle d'orientation.

Nous avons testé cet estimateur sur deux textes, à partir des segments résultants de la méthode *BFA*, afin de vérifier si les résultats sont similaires aux résultats obtenus avec les deux estimateurs définis antérieurement.

	Réel	Moyenne	Médiane	Moy. pondérée
Texte 1	0°	-0.14°	0.00°	-0.17°
Texte 2	30°	29.72°	30.07°	29.51°

TAB. 5.2 – Tableau comparatif entre les estimateurs

Selon le tableau ci-dessus, pour les deux textes utilisés, les trois estimateurs fournissent des résultats similaires d'estimation de l'*AOT*. Afin de valider les performances du dernier estimateur proposé, des tests devront être fait sur un nombre comparables de textes.

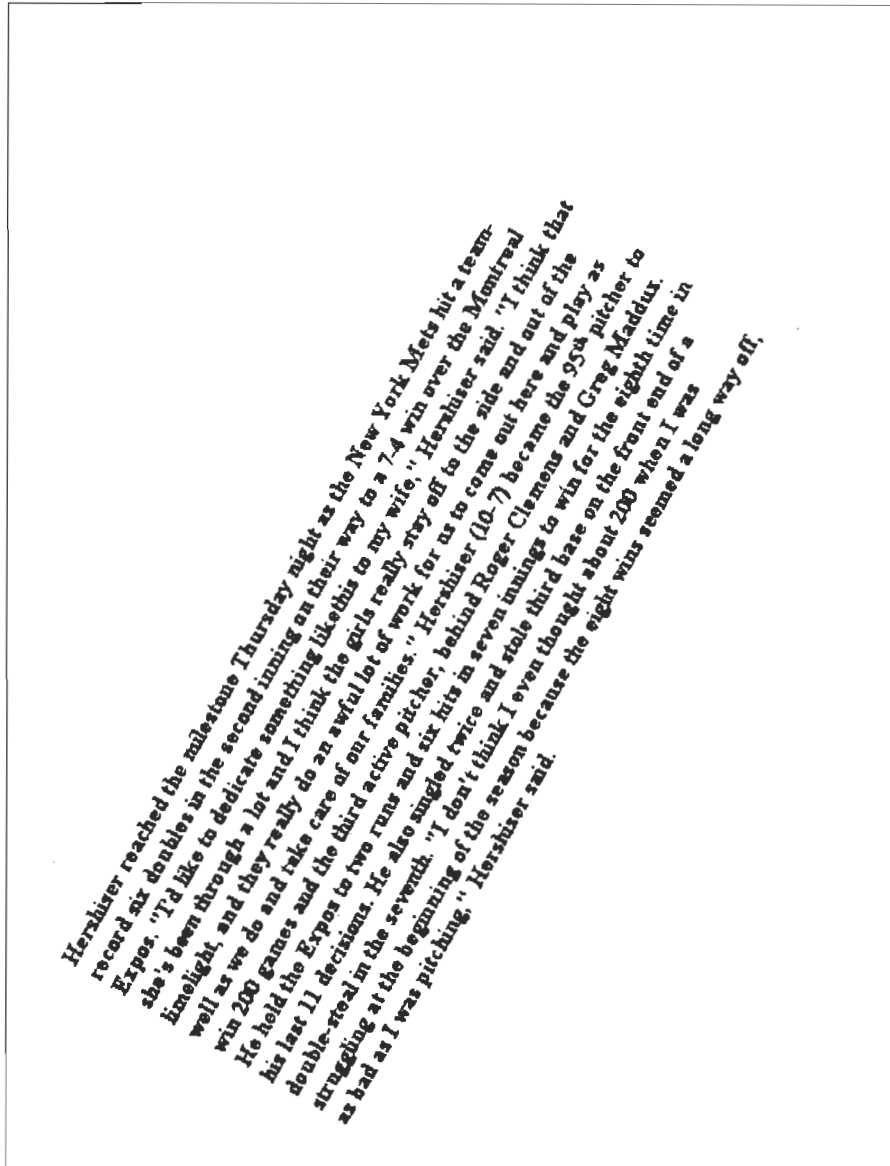
Tests supplémentaires

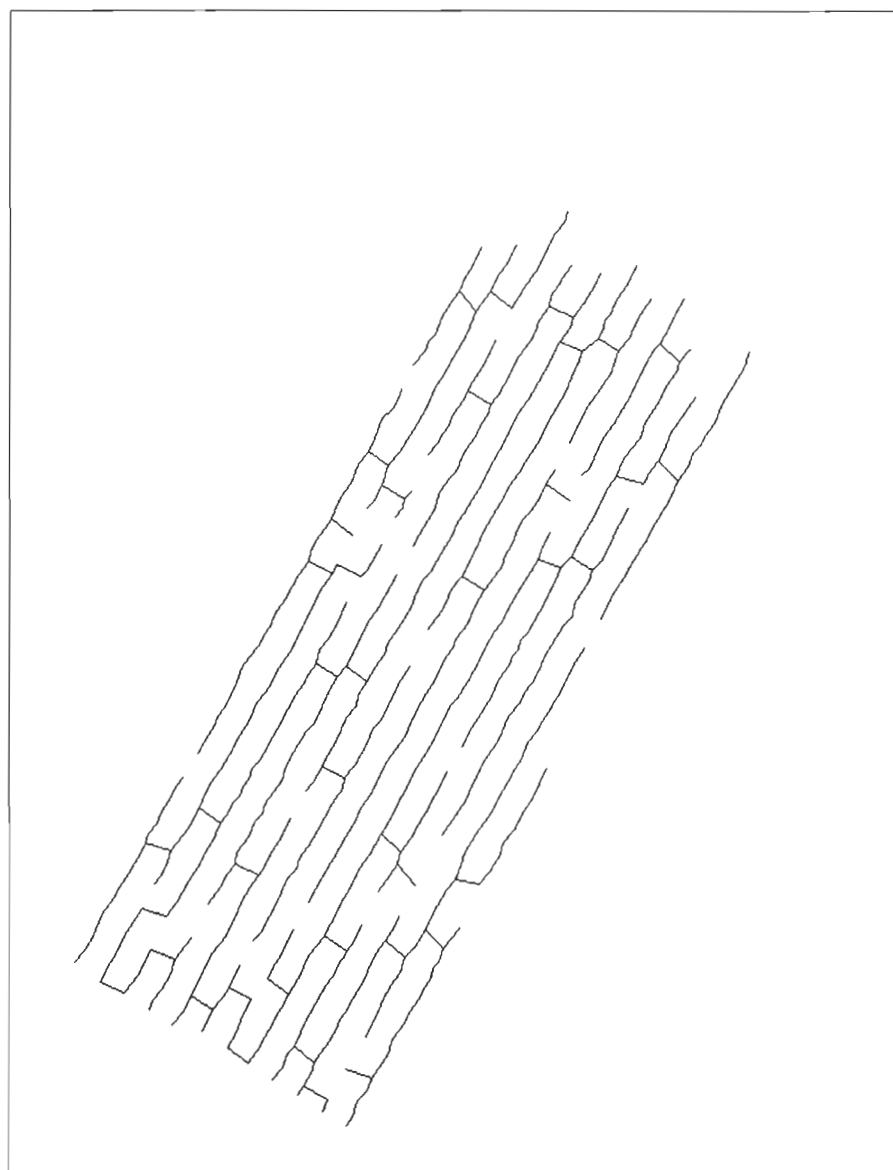
Les estimateurs ont été testés sur des textes ayant une grande déformation, soit -60° , -30° , 30° et 60° . Dans la réalité, il est plus probable que les angles d'orientation des textes soient plus faibles, c'est-à-dire plus près de 0° . Il est donc nécessaire de tester les estimateurs sur des textes de plus faibles angles pour vérifier si l'erreur d'estimation est aussi précise que dans le cas de grands AOT .

Nous avons utilisé un échantillon de 25 textes anglais et français ayant des AOT de 1° , 2° , 5° et 10° . Les simulations ont menées à des résultats similaires au cas de grands AOT , soit une erreur moyenne de 0.38° sur l'ensemble des textes comparativement à 0.24° pour les angles supérieurs ou égaux à $|30^\circ|$. Les résultats des simulations pour l'ensemble des 25 textes sont présentés dans le tableau ci-dessous, où l'erreur est celle obtenue en utilisant la moyenne comme estimateur à partir des segments calculés avec la méthode *BFA*.

Angle	Erreur moyenne
0°	0.50°
1°	0.33°
2°	0.42°
5°	0.32°
10°	0.33°
Total	0.38°

TAB. 5.3 – Estimation de faibles AOT

FIG. 5.2 – Texte original anglais ($AOT = 60^\circ$)

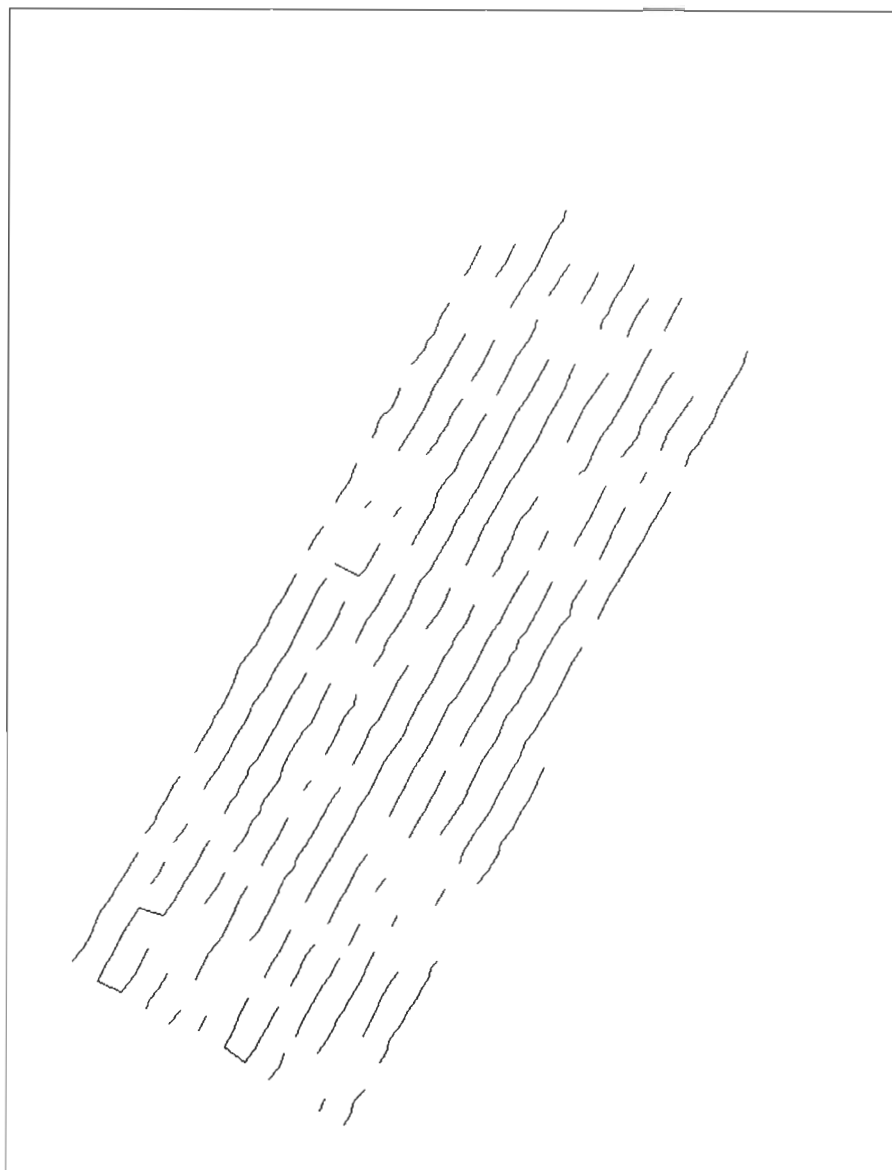


Nombre de segments : 631

Intervalle sur la moyenne : 54.14 ± 1.82

Intervalle sur la médiane : 59.04 ± 0.68

FIG. 5.3 – Résultat de la méthode d'Ittner sur le texte anglais

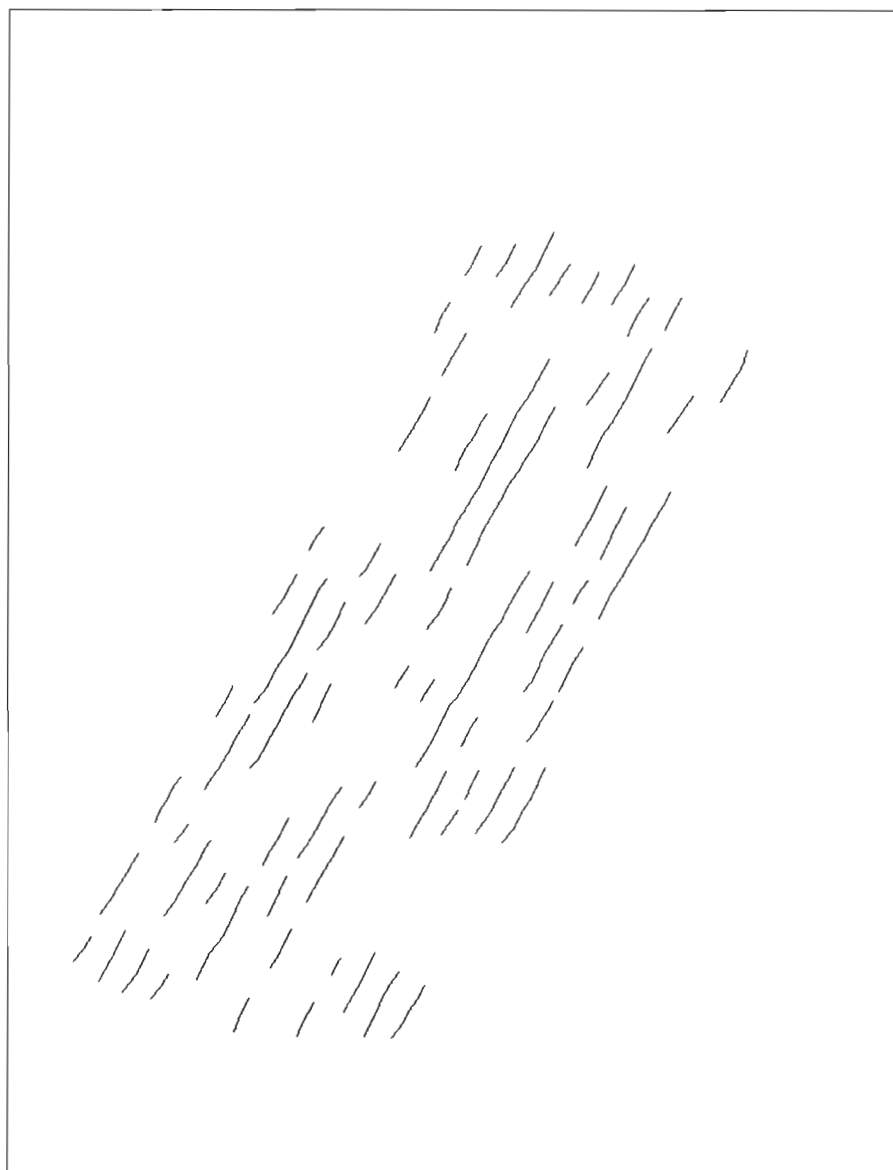


Nombre de segments : 479

Intervalle sur la moyenne : 59.06 ± 0.91

Intervalle sur la médiane : 59.53 ± 0.76

FIG. 5.4 – Graphe obtenu avec le *Bamboo field*

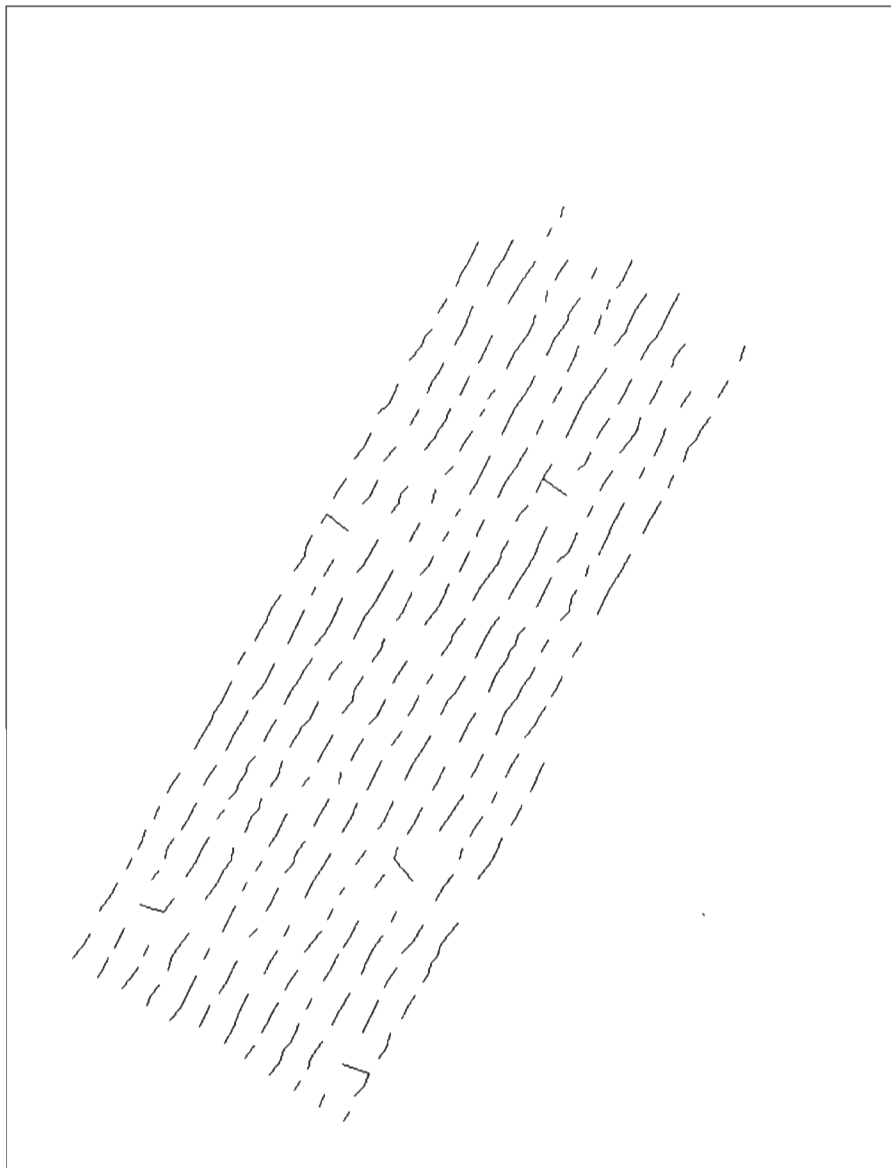


Nombre de segments : 276

Intervalle sur la moyenne : 59.46 ± 0.51

Intervalle sur la médiane : 59.62 ± 0.61

FIG. 5.5 – Application du *Bamboo field amélioré*

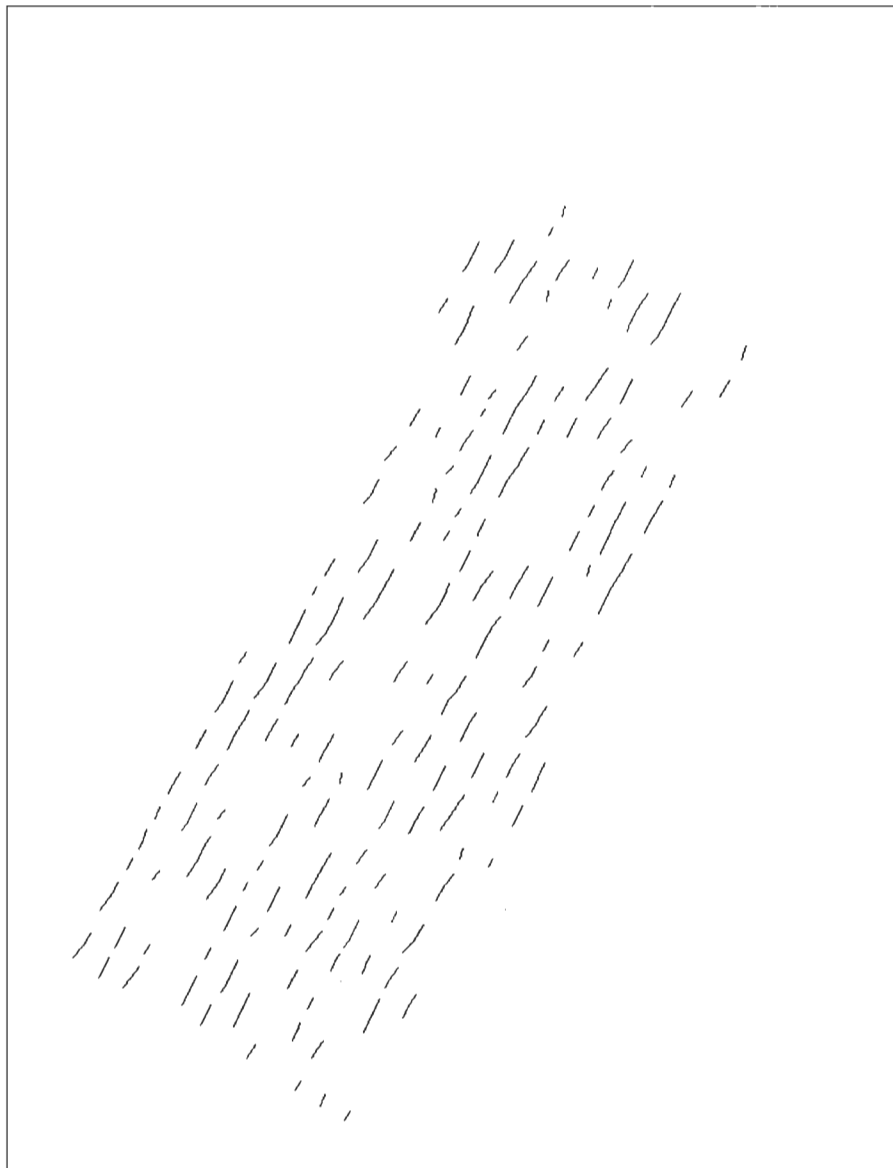


Nombre de segments : 437

Intervalle sur la moyenne : 58.76 ± 1.16

Intervalle sur la médiane : 59.53 ± 0.93

FIG. 5.6 – Ensemble de segments générés par le *NNG*



Nombre de segments : 270

Intervalle sur la moyenne : 58.97 ± 0.95

Intervalle sur la médiane : 59.57 ± 0.79

FIG. 5.7 – Ensemble de segments générés par le *NNGA*

Chapitre 6

Simulations

Au chapitre 5, nous avons comparé cinq approches géométriques pour le calcul de l'*AOT* en utilisant deux estimateurs, soit la moyenne et la médiane, et ce à partir d'un échantillon constitué de 50 textes français et de 50 textes anglais. Les résultats de cette simulation nous ont permis de conclure que la méthode du *Bamboo field amélioré* est la plus efficace lorsque la moyenne est utilisé comme estimateur et que dans le cas de la médiane, le *Bamboo field amélioré* et le *Bamboo field* génèrent des résultats similaires et précis.

Dans ce chapitre, nous allons considérer seulement la méthode du *Bamboo field amélioré* car elle s'avère la plus précise, afin de démontrer que l'approche est indépendante du langage utilisé si celui-ci respecte la condition C_1 suivante :

*la distance entre deux caractères consécutifs sur une
même ligne est plus petite que la distance entre deux
caractères sur deux lignes différentes .* (C_1)

Dans la deuxième partie de ce chapitre, un raffinement de la méthode est présenté, lequel améliorera la rapidité du traitement. Les résultats de ces simulations démontreront que cette amélioration, en plus de diminuer le temps d'exécution, pro-

duit des résultats aussi précis.

6.1 Indépendance du langage

Au chapitre 5, seulement des simulations sur des textes français et anglais ont été effectuées. Ces deux langages respectent la conditions C_1 , mais ils ne sont pas les seuls langages à s'y conformer. Nous n'avons qu'à penser au chinois, au japonais, à l'allemand, à l'espagnol, au russe, à l'italien, etc.

Afin de démontrer l'indépendance de la nouvelle méthode par rapport aux langages respectant la condition C_1 , nous avons utilisé la même méthodologie qu'au chapitre 4, c'est-à-dire, nous nous sommes basés sur les mêmes paramètres statistiques, l'erreur calculé à partir du biais et de la marge d'erreur.

Nous avons constitué un échantillon de 150 textes écrits en français, en anglais, en japonais, en espagnol, en russe et en allemand. Chaque langue comportait le même nombre de textes, soit 25 (5 textes différents ayant subi des rotations de -60° , -30° , 0° , 30° et 60°).

On retrouve au tableau 6.1 les résultats obtenus pour chacune des six langues testées. Nous retrouvons également dans ce tableau le biais absolu et la marge d'erreur obtenus pour la moyenne et la médiane.

Nous remarquons que les estimations des angles d'orientation sont très précises, sauf dans le cas du japonais lorsque la médiane est utilisée comme estimateur. Ceci est dû au fait qu'en japonais, il y a moins de caractères au pouce que dans les autres langues, ce qui produit une réduction du nombre de segments servant à l'estimation de l' AOT , par conséquent la médiane devient moins efficace. De plus la marge d'erreur est plus élevée pour les textes japonais étant donné le nombre réduit de segments servant à l'approximation de l' AOT . Les figures qui suivent montrent des exemples de l'application de la méthode du *Bamboo field amélioré* sur des textes

	Moyenne		Médiane		n
	biais	marge d'erreur	biais	marge d'erreur	
Français	0.16°	0.42°	0.18°	0.42°	25
Anglais	0.26°	0.54°	0.24°	0.63°	25
Japonais	0.48°	0.93°	1.03°	0.89°	25
Espagnol	0.35°	0.42°	0.17°	0.43°	25
Russe	0.19°	0.42°	0.16°	0.29°	25
Allemand	0.65°	0.60°	0.47°	0.68°	25
Total	0.35°	0.56°	0.38°	0.56°	150

TAB. 6.1 – Application du *BFA* sur des textes de différentes langues

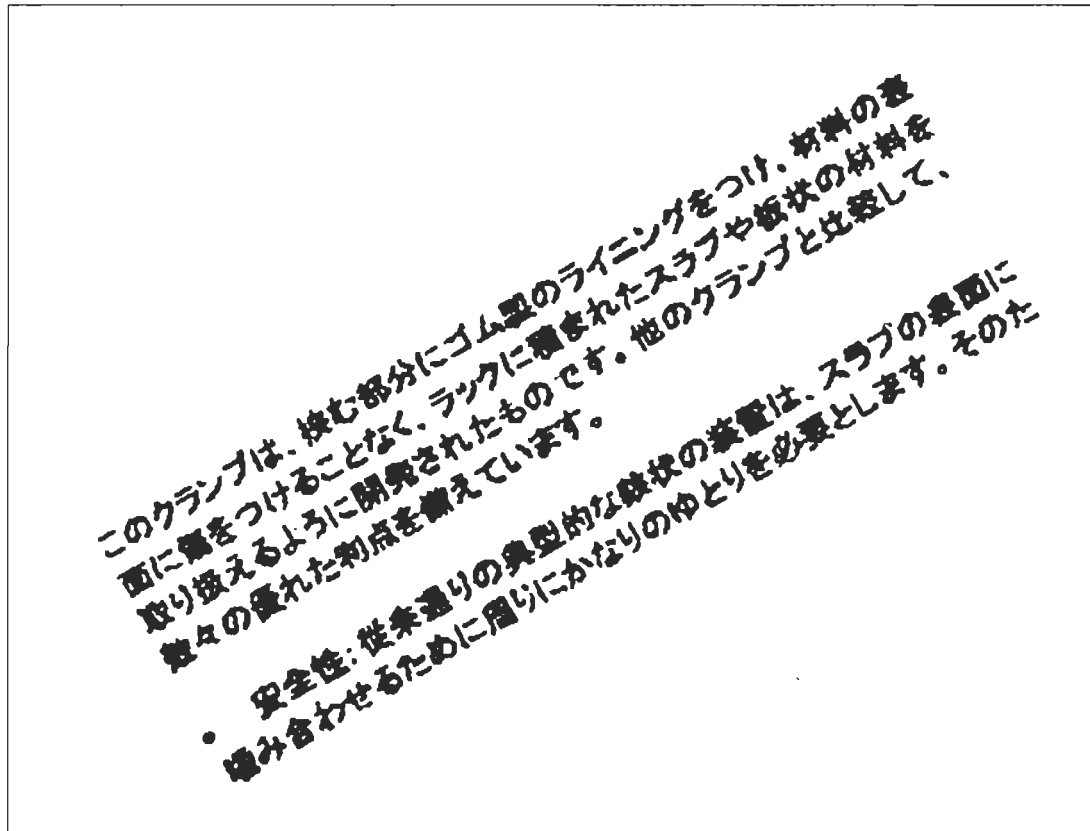
de différentes langues, soit les figures 6.1 et 6.2 en japonais, les figures 6.3 et 6.4 en allemand et les figures 6.5 et 6.6 en espagnol.

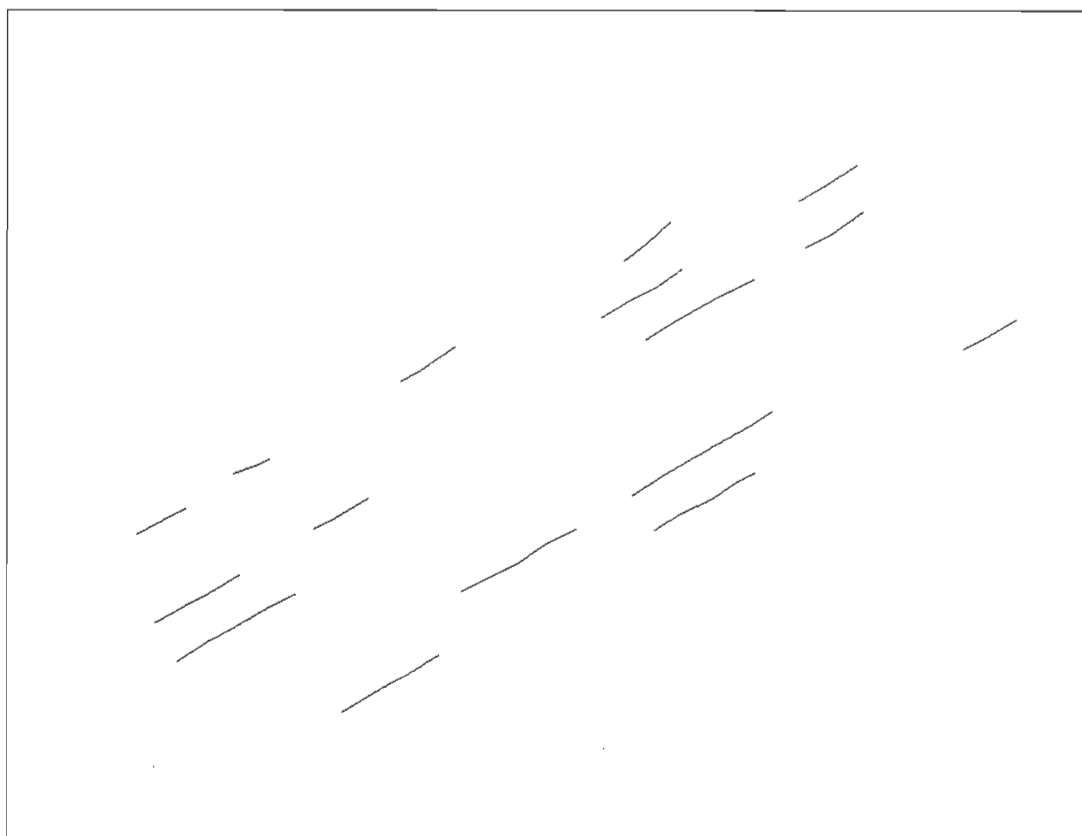
6.2 Amélioration de la rapidité

Sachant que la méthode du *Bamboo field amélioré* fournit des résultats très satisfaisants, nous pouvons nous interroger sur l'amélioration du temps d'exécution de la méthode.

Intuitivement, le traitement d'une partie ou d'un pourcentage de l'image devrait fournir une approximation assez juste de l'*AOT*. Par pourcentage d'une image, nous entendons un pourcentage de la surface de l'image et non un pourcentage de nombre de segments traités. D'autre part, si une portion seulement de l'image est utilisée, le nombre de segments servant à l'estimation devient plus petit, ce qui a pour effet d'augmenter la marge d'erreur.

Afin d'analyser l'erreur encourue en ne traitant qu'une partie de l'information, nous avons utilisé les 100 textes originaux du chapitre 5, c'est-à-dire, les 50 textes anglais

FIG. 6.1 – Texte original en japonais ($AOT = 30^\circ$)

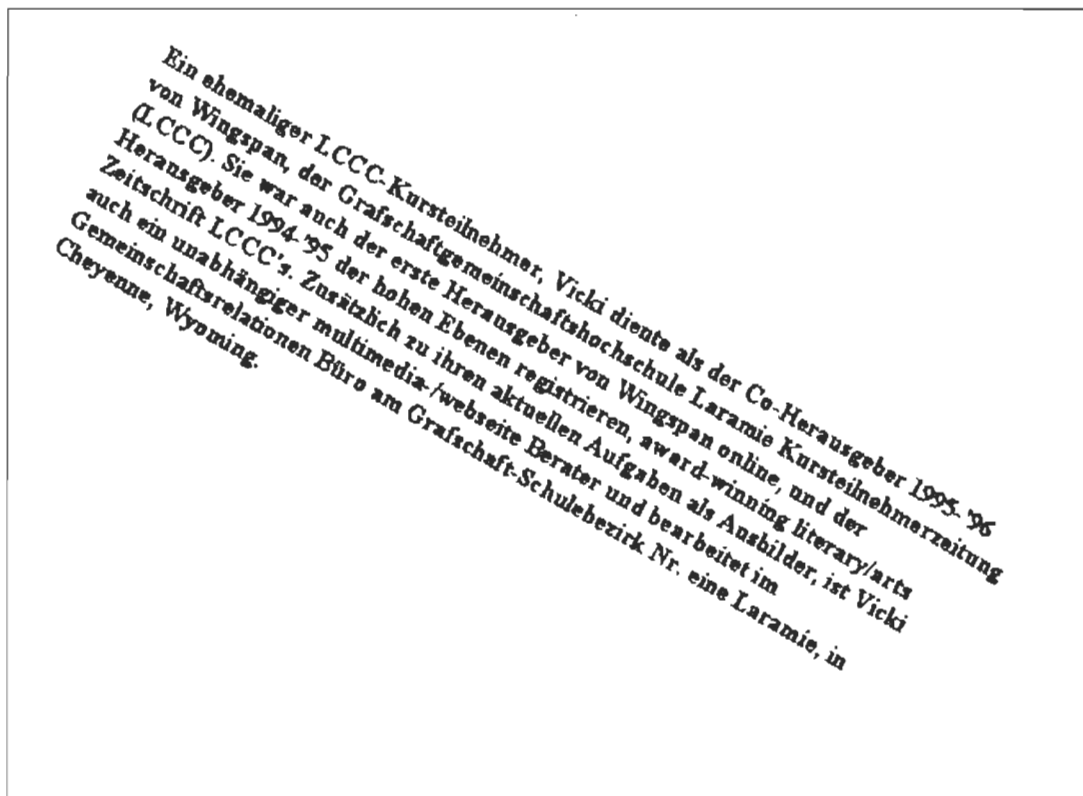


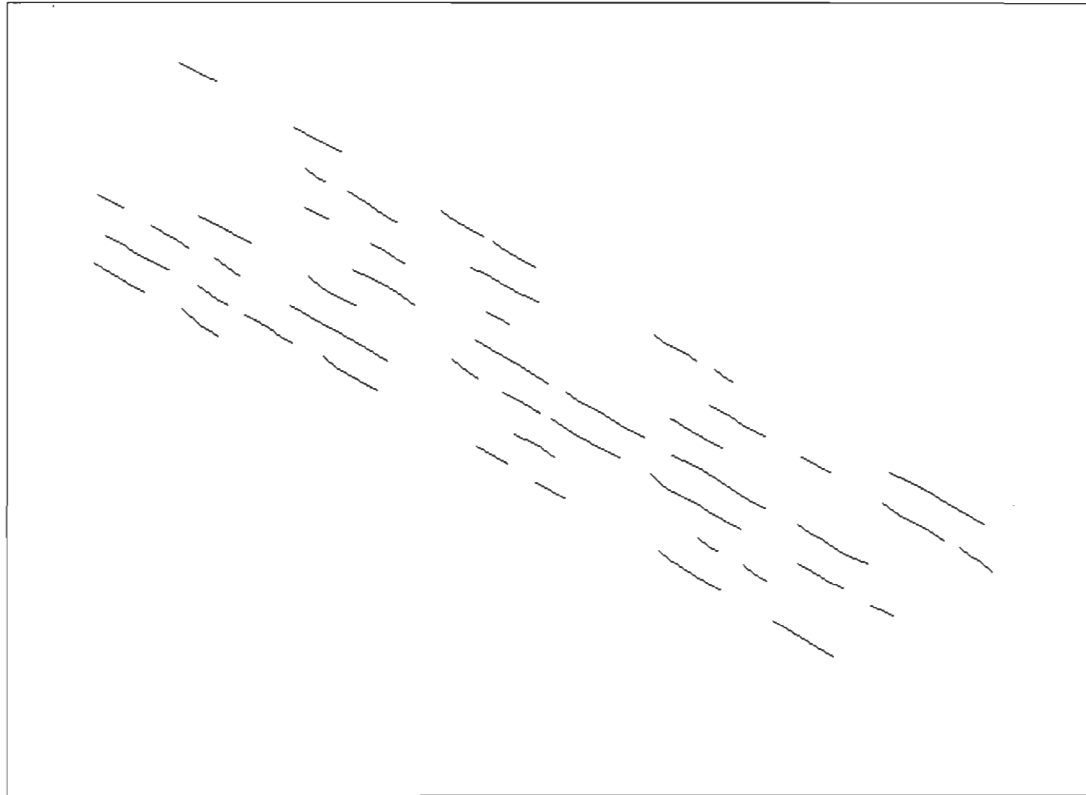
Nombre de segments : 46

Intervalle sur la moyenne : 30.03 ± 1.12

Intervalle sur la médiane : 30.26 ± 1.24

FIG. 6.2 – Résultat de la méthode *BFA* sur le texte en japonais

FIG. 6.3 – Texte original en allemand ($AOT = -30^\circ$)

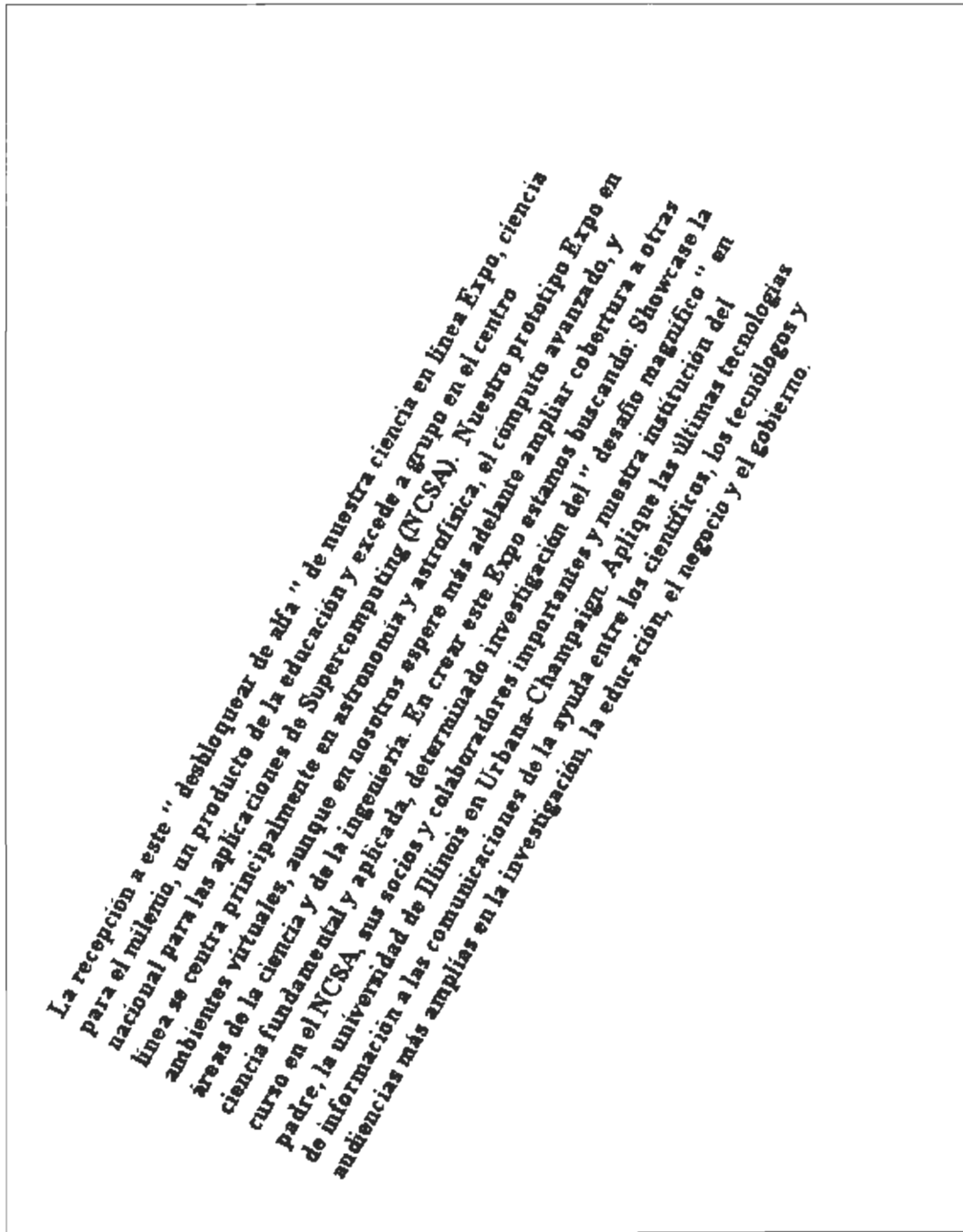


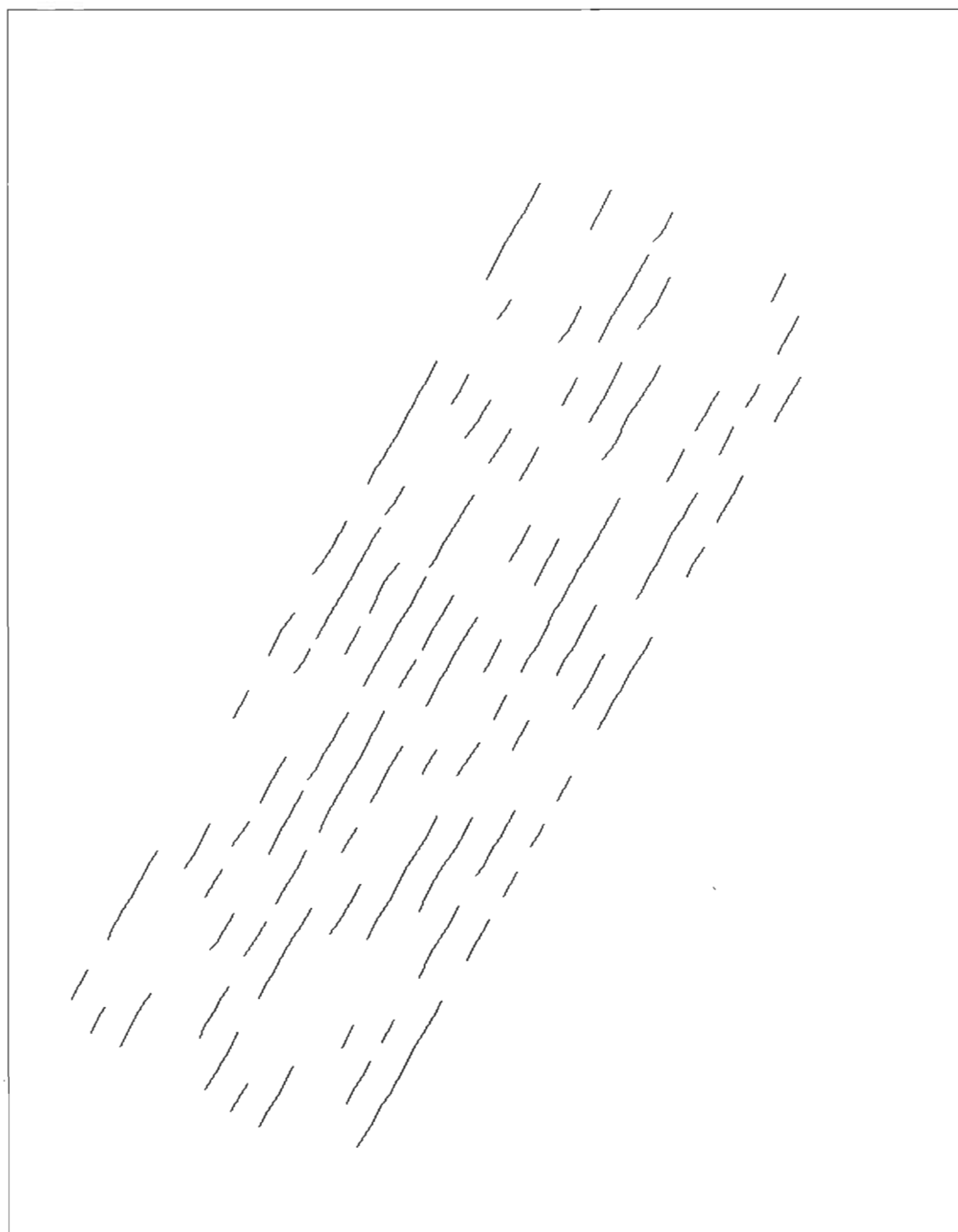
Nombre de segments : 161

Intervalle sur la moyenne : -30.26 ± 0.71

Intervalle sur la médiane : -29.74 ± 1.08

FIG. 6.4 – Résultat de la méthode *BFA* sur le texte en allemand

FIG. 6.5 – Texte original en espagnol ($AOT = 60^\circ$)



Nombre de segments : 309

Intervalle sur la moyenne : 59.93 ± 0.38

Intervalle sur la médiane : 60.26 ± 0.36

FIG. 6.6 – Résultat de la méthode *BFA* sur le texte en espagnol

et les 50 textes français. Nous avons appliqué la méthode du *Bamboo field amélioré* sur cet échantillon de deux façons différentes. Dans un premier temps sur les textes en entier et dans un second temps sur 25% du texte extrait à partir du milieu de l'image. On trouve les résultats obtenus ci-dessous.

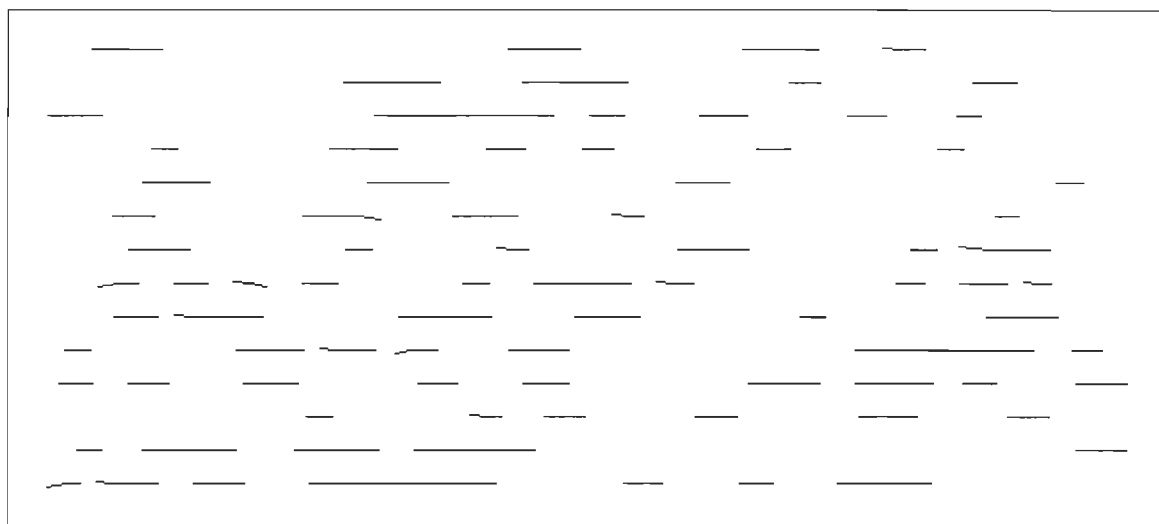
	Moyenne		Médiane	
	biais	marge d'erreur	biais	marge d'erreur
25%	0.30°	0.70°	0.30°	0.61°
100%	0.24°	0.48°	0.20°	0.45°

TAB. 6.2 – Résultats de la méthode *BFA* appliquée à 25% et à 100% des textes

Dans ce tableau, nous remarquons que les erreurs d'approximation sont sensiblement les mêmes, légèrement plus élevées dans le cas de 25% des textes, mais négligeables, et comme prévu, la marge d'erreur est plus élevée. Nous avons obtenu avec la moyenne comme estimateur des marges d'erreur de 0.70 et 0.48 pour le quart et la totalité du texte respectivement et 0.61 et 0.45 avec la médiane. On trouve aux figures 6.8 et 6.9 les résultats obtenus en traitant 100% et 25% du texte de la figure 6.7.

4e journée pédagogique. Tous les professeurs de l'école qui enseignent aux 2e et 3e cycles bénéficient d'un perfectionnement en grammaire. Ce perfectionnement est donné par le conseiller pédagogique de la commission scolaire. Au menu : la phrase de base, les groupes syntaxiques, les opérations sur la phrase et la répartition des objets d'apprentissage par cycle. Réunion de tout le personnel de l'école. Et là je comprends pourquoi le directeur de l'école me posait toutes ces questions sur la lecture. C'est que le projet particulier de l'école cette année est la promotion de la lecture et, particulièrement, de la lecture littéraire. Les parents sont également impliqués. Ils recevront un document qui leur permettra de réaliser certaines activités à la maison. Ce document a été élaboré par les professeurs de l'école, le directeur, certains parents, le conseiller pédagogique de la commission scolaire ainsi qu'un professeur de l'université du Québec. Par cycle, nous devons nous entendre sur un choix d'activités : cercles de lecture, heure du conte, critique littéraire...le choix est infini. Pour nous aider, nous pouvons consulter la bibliothèque pédagogique de l'école : on y retrouve Jocelyne Gasson, une série de brochures proposant plusieurs activités.

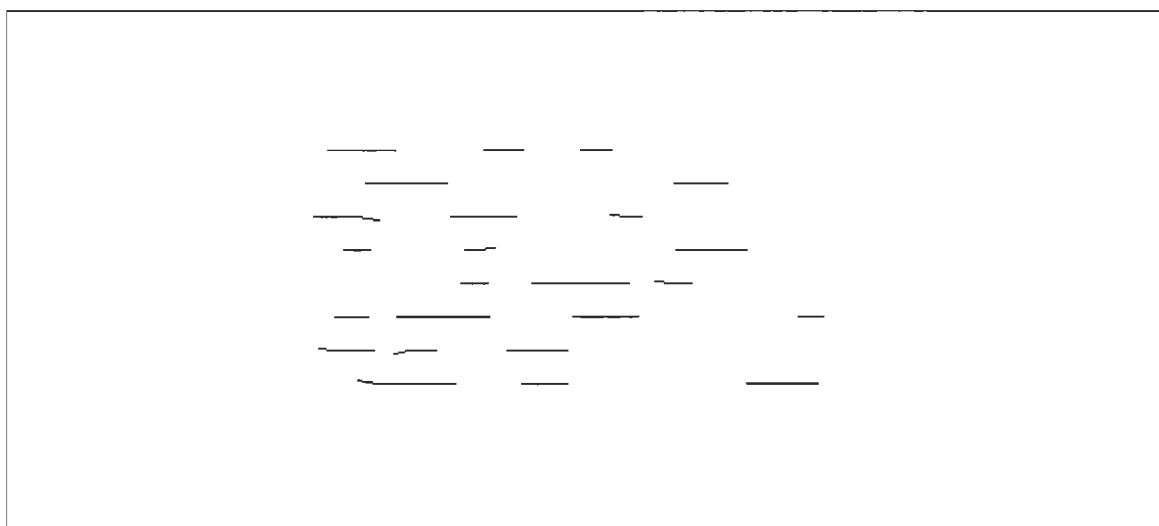
FIG. 6.7 – Texte original en français ($AOT = 0^\circ$)



Nombre de segments : 303

Intervalle sur la moyenne : -0.28 ± 0.22

FIG. 6.8 – Méthode du *BFA* appliquée sur 100% du texte en français



Nombre de segments : 83

Intervalle sur la moyenne : -0.28 ± 0.53

FIG. 6.9 – Méthode du *BFA* appliquée sur 25% du texte en français

Chapitre 7

Conclusion

Dans ce mémoire, nous avons résolu un problème important en traitement automatique de documents : la détection de l'angle d'orientation d'un texte. Nous avons exposé cinq différentes approches permettant d'estimer l'angle d'orientation d'un texte, il s'agit de la méthode d'Ittner, du *Bamboo Field* et de trois nouvelles approches. Parmi celles-ci, trois d'entre elles utilisent le même graphe de proximité, l'arbre sous-tendant minimal tandis que les deux autres approches utilisent le *nearest neighbor graph*.

Des comparaisons ont été faites afin de déterminer laquelle de ces approches est la plus efficace. Le biais et la marge d'erreur sont les paramètres statistiques qui ont été utilisés pour comparer les différentes méthodes. À partir des simulations effectuées, la méthode du *Bamboo field amélioré* s'est avérée comme étant la plus précise avec une erreur moyenne de 0.24° et une marge d'erreur de 0.48° dans le cas où la moyenne est employée comme estimateur et de 0.20° comme erreur moyenne et 0.45° pour la marge d'erreur dans le cas de la médiane.

De plus, des simulations supplémentaires ont été faites afin de montrer que la nouvelle approche est indépendante du langage en autant que celui-ci respecte la condition suivante : *la distance entre deux caractères consécutifs est plus petite que la*

distance entre deux lignes du texte. L'échantillon de textes était constitué de textes français, anglais, japonais, allemand, russe et espagnol.

Nous avons également montré qu'en traitant seulement un pourcentage du texte (25%), les résultats obtenus sont similaires à ceux obtenus en traitant l'image en entier, ce qui nous permet d'accélérer le calcul de l'*AOT*. De plus, un nouvel estimateur a fourni des résultats similaires à ceux qui ont été obtenus avec la moyenne et la médiane. Des tests supplémentaires confirmeraient le potentiel de ce meilleur estimateur. Finalement, la méthode *BFA*, qui s'est avérée la plus précise, produit un bon estimé de l'*AOT* tant pour les fortes que pour les faibles déformations de l'orientation d'un texte.

Un autre problème relié au sujet de ce mémoire, lequel pourrait être abordé à l'aide d'une approche géométrique, est celui de la direction de l'écriture d'un texte, par exemple, de gauche à droite, de haut en bas et réciproquement. Évidemment, le traitement de textes ne respectant pas le critère C_1 demeure entier. Dans ce dernier cas, des transformations gardant invariant l'*AOT* pourraient être appliquées à un texte lors des étapes de pré-traitement afin d'obliger le respect de la contrainte C_1 . Le traitement des textes manuscrits et des textes contenant des équations constitue de toute évidence un défi à relever.

Bibliographie

- [1] A. AMIN et R. SHIU (1996), *New Skew Detection and Correction Algorithms*, Proc. of 5th International Workshop on Frontiers in Handwritten Recognition, Essex, Angleterre, p. 251.
- [2] G. BAILLARGEON, (1990), *Méthode statistiques de l'ingénieur*, Édition SMG, Trois-Rivières (Qc), Canada.
- [3] H.S. BAIRD, (1987), *The skew angle of a printed documents*, Proc. of 1987 Conf. Society of Photographic Scientist and Engineers, Rochester (NY), États-Unis, May 20-21.
- [4] P.K. BOSE, J.D. CARON et K. GHOUDI, (1998), *Detection of Text-line Orientation*, Proc. of Tenth Canadian Conference on Computational Geometry, Montréal (Qc), Canada, August 1998.
- [5] G. BRASSARD et P. BRATLEY (1987), *Algorithmique, conception et analyse*, Masson, Montréal (Qc), Canada.
- [6] J.D. CARON (1999), *Détection de l'angle d'orientation d'un texte*, Proc. of 12th Vision Interface Conference, Trois-Rivières (Qc), Canada, May 1999.
- [7] A. CHALIFOUR, S. BEAUCHEMIN, P. BOSE, F. NOUBOUD et L.B.-M. VOUGHT (1998), *Analyse morphologique des filets de tricoptères : un problème d'écotoxicologie et de géométrie discrète*, Proc. of 11th Vision Interfage Conference, Vancouver (BC), Canada, June 1998, p. 479-486.
- [8] M. DE BERG, M. VAN KREVELD, M. OVERMARS et O. SCHWARZKOPF (1997), *Computational Geometry - Algorithms and Applications*, Springer-Verlag, Berlin, Allemagne.

- [9] R.O. DUDA et P.E. HART (1972), *Pattern classification and scene analysis*, John-Wiley and Sons, New York (NY), États-Unis.
- [10] A. HENNIG, G. RAZA, N. SHERKAT et R.J. D'WHITROW (1996), *Detecting a Document's : A simple Stochastic Approach*, Department of Computing, Nottingham, Angleterre, to be submitted.
- [11] S. HINDS, J. FISHER et D. D'AMATO (1990), *A document skew detection method using runlength encoding and the Hough transform*, Proc. of 10th Int. Conf. on Pattern Recognition, Atlantic City, États-Unis, p. 464-468.
- [12] P.V.C. HOUGH, (1962), *Method and means for recognizing complex patterns*, United States Patent, États-Unis.
- [13] D.J. ITTNER, (1993), *Automatic inference of textline orientation*, Proc. 2nd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, États-Unis, p. 123-133.
- [14] D.X. LE et G. THOMA (1993), *Document Skew Angle Detection Algorithm*, Proc. 1993 SPIE Symposium on Aerospace and Remote Sensing - Visual Information Processing II, Orlando, États-Unis, vol. 1961, p. 251-262.
- [15] D.X. LE, G. THOMA et H. WESCHLER (1994), *Automated Page Orientation and Skew Angle Detection for Binary Document Images*, Pattern Recognition, vol. 27, no 10, p. 1325-1344.
- [16] J. O'ROURKE (1998), *Computational geometry in C*, Cambridge Press, Cambridge, États-Unis.
- [17] C. PATERAS et G. TOUSSAINT, (1995), *An improved algorithm for text-line orientation estimation in document*, School of Computer Science, McGill University, Montréal (Qc), Canada, to be submitted.
- [18] R.C. PRIM (1957), *Shortest Connection Networks and Some Generalizations*, Bell System Technical Journal, p. 1389-1401.
- [19] N. RONDEL et G. BUREL (1995), *Cooperation of multi-layer perceptrons for the estimation of the skew angle in text document images*, Proc. of 3rd Int. Conf. on Pattern Document Analysis and Recognition, Montreal (Qc), Canada, p. 1141-1144.

- [20] K.H. ROSEN (1991), *Discrete Mathematics and its Applications*, McGraw-Hill, Princeton, États-Unis.
- [21] J. SAUVOLA et M. PIETIKAINEN, (1994), *Skew angle detection using texture direction analysis*, Department of Electrical Engineering, University of Oulu, Finland, to be submitted.
- [22] R. SMITH (1995), *A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation*, 3rd International Conference Document Analysis and Recognition, Montréal (Qc), Canada.
- [23] C. SUN et D. SI (1997), *Skew and Slant Correction for Document Images Using Gradient Direction*, 4rd International Conference Document Analysis and Recognition, Montréal (Qc), Canada.
- [24] B. WAKED, S. BERGLER, S. KHOURY et C.Y. SUEN (1998), *Skew Detection, Page Segmentation, and Script Classification of Printed Document Images*, Center for Pattern Recognition and Machine Intelligence, GM-606, Concordia University, Montréal (Qc), Canada.
- [25] B. YU et A.K. JAIN (1996), *A Robust and Fast Skew Detection Algorithm for Generic Documents*, Pattern Recognition, vol. 29, p. 1599-1629.
- [26] C.L. YU, Y.Y. TANG et C.Y. SUEN (1995), *Document skew detection based on fractal and least squares method*, Proc. of the 3rd Int. Conf. on Pattern Document Analysis and Recognition, Montreal (Qc), Canada, p. 1149-1152.