Institut d'Informatique Université de Fribourg (Suisse)

Une approche uniforme pour la reconnaissance de la structure physique de documents composites fondée sur l'analyse des espaces

THESE

présentée à la faculté des Sciences de l'Université de Fribourg (Suisse)
pour l'obtention du grade de Doctor scientiarum informaticarum

Antoine Sourou AZOKLY
de
Porto-Novo, République du Bénin

Thèse No. 1105

Imprimeries Centrales Neuchâtel S.A., 1995

Acceptée par la Faculté des Sciences de l'Université de Fribourg (Suisse) sur la proposition des Professeurs Rolf INGOLD (Université de Fribourg, Suisse), Georges STAMON (Université de Paris V, France) et Karl TOMBRE (INRIA Lorraine, Nancy France).

Fribourg, le 13 octobre 1995

Le Doyen:

Prof. Jean-Paul BERRUT

à

mon père,

tous ceux que j'aime

et aux

enfants orphelins de la bêtise humaine

au Rwanda

et

en ex-Yougoslavie

Table des matières

1 Introduction Générale **1**
- 1.1 La problématique . . . 1
  - 1.1.1 Le document du point de vue informatique . . . 1
  - 1.1.2 L'intérêt de la reconnaissance des documents . . . 1
- 1.2 L'objet de cette thèse . . . 2
  - 1.2.1 Motivations . . . 2
  - 1.2.2 Objectifs . . . 2
  - 1.2.3 Plan . . . 3

2 Structures et normes régissant les documents **5**
- 2.1 Structures de documents . . . 5
  - 2.1.1 Structure logique . . . 5
  - 2.1.2 Structure physique . . . 6
  - 2.1.3 Représentation . . . 6
  - 2.1.4 Spécialisation . . . 7
  - 2.1.5 Granularité . . . 7
- 2.2 Normes et architectures régissant les documents . . . 8
  - 2.2.1 Normes régissant les documents . . . 9
  - 2.2.2 Systèmes de manipulation de documents . . . 11

3 Un apperçu des techniques de reconnaissance **13**
- 3.1 Objectif . . . 13
- 3.2 Prétraitement . . . 14
  - 3.2.1 Estimation de l'inclinaison . . . 15
  - 3.2.2 Redressement . . . 16
- 3.3 Reconnaissance de structures physiques . . . 16
  - 3.3.1 Segmentation . . . 16
  - 3.3.2 Etiquetage . . . 19
- 3.4 Reconnaissance de structures logiques . . . 21
  - 3.4.1 Approche syntaxique . . . 21
  - 3.4.2 Approche IA . . . 22
- 3.5 Les limites . . . 23

4 Modèle des documents composites **25**
- 4.1 Introduction et définitions . . . 25
- 4.2 Formatage conventionel . . . 25
  - 4.2.1 Attributs usuels de formatage . . . 26
  - 4.2.2 Règles de présentation . . . 27
  - 4.2.3 Modèle de pages . . . 29
- 4.3 Structure physique générique . . . 31
  - 4.3.1 Macrostructures . . . 31

4.3.2 Microstructures . . . 33
4.3.3 Structures élémentaires . . . 34
4.3.4 Modélisation des entités physiques . . . 34
4.4 Délimiteurs d'entités physiques . . . 35
4.4.1 Importance des espaces . . . 35
4.4.2 Rectangles structurants . . . 35
4.4.3 Séparateurs . . . 36
4.5 Architecture du système de reconnaissance . . . 38

5 Règles de production des microstructures 41
5.1 Fondement . . . 41
5.1.1 Attributs métriques . . . 42
5.1.2 Topologie locale entre entités physiques . . . 42
5.1.3 Seuils métriques . . . 44
5.1.4 Structures terminales . . . 45
5.2 Structure graphique usuelle des blocs textuels . . . 45
5.2.1 Texte . . . 45
5.2.2 Ligne . . . 46
5.2.3 Mot . . . 47
5.3 Structure graphique usuelle des formules . . . 48
5.4 Structure graphique usuelle des tableaux . . . 50
5.5 Structure graphique usuelle des illustrations . . . 51
5.6 Aspects graphiques élémentaires . . . 53

6 Reconnaissance de microstructures 57
6.1 Classification des composantes connexes . . . 57
6.1.1 Définition et objectifs . . . 57
6.1.2 Extraction de composantes connexes . . . 58
6.1.3 Classification par regroupement des plus proches voisins . . . 58
6.1.4 Initialisation des classes . . . 59
6.1.5 Raffinement de la classification . . . 61
6.2 Etiquetage suivant une approche de découpe hiérarchique . . . 61
6.2.1 Extraction des rectangles structurants . . . 61
6.2.2 Découpe hiérarchique des microstructures . . . 63
6.2.3 Etiquetage de microstructures . . . 64
6.2.4 Limites . . . 64
6.3 Etiquetage suivant une approche mixte . . . 65
6.3.1 Algorithme générique d'étiquetage par fusion . . . 65
6.3.2 Blocs textuels . . . 66
6.3.3 Expressions mathématiques . . . 68
6.3.4 Tableaux . . . 71
6.3.5 Blocs graphiques et photographiques . . . 72
6.4 Estimation des lignes de base . . . 73
6.4.1 Définition et motivation . . . 73
6.4.2 Estimation à partir des signes . . . 74
6.5 Ordre de parcours dans les structures physiques . . . 75
6.5.1 Objectif . . . 75
6.5.2 Ordres de parcours . . . 75

7 Estimation des seuils métriques **79**
7.1 Un estimateur paramétrique des seuils métriques 80
7.1.1 Fondement 80
7.1.2 L'estimateur Σ 80
7.2 Seuil des espaces entre signes diacritiques et lettres 82
7.3 Seuil des espaces inter-caractères 82
7.4 Seuil des espaces inter-mots 83
7.5 Seuil des espaces inter-lignes 84
7.5.1 L'interligne 84
7.5.2 Espace inter-lignes effectif 85

8 Langage de description des macrostructures **87**
8.1 Motivation et originalité 87
8.2 Description des classes de documents 88
8.3 Description des classes de pages 90
8.4 Description des séparateurs génériques 91
8.4.1 Séparateur 91
8.4.2 Séparateur de substitution 91
8.5 Description des régions génériques 92
8.6 Description de documents spécifiques 94

9 Reconnaissance des macrostructures **97**
9.1 Analyseurs des descriptions de macrostructures 97
9.1.1 Analyseur de descriptions génériques 97
9.1.2 Analyseur de descriptions spécifiques 97
9.2 Segmentation en régions 98
9.2.1 Détermination des régions 98
9.2.2 Extraction des séparateurs 99
9.3 Segmentation en blocs et identification des tableaux 101
9.3.1 Découpe hiérarchique 102
9.3.2 Découpe en colonnes 104
9.3.3 Découpe en rangées 104
9.3.4 Découpe des blocs mosaïques 105
9.3.5 Analyse des tableaux 106

10 Evaluation **107**
10.1 Distance dans l'espace des structures arborescentes 107
10.1.1 Rappel 107
10.1.2 Coût de la transformation d'un arbre en un autre 107
10.2 Distance dans l'espace des structures physiques 108
10.3 Evaluation des techniques de reconnaissance 110
10.3.1 Une architecture à la carte 110
10.3.2 Environnement d'évaluation 111
10.3.3 Classification des composantes connexes 111
10.3.4 Estimation des seuils métriques 112
10.3.5 Fiabilité 112
10.3.6 Efficacité 114

11 Conclusion générale **117**
11.1 Bilan . . . 117
11.1.1 Classification des composantes connexes . . . 117
11.1.2 Reconnaissance des microstructures . . . 118
11.1.3 Reconnaissance des macrostructures . . . 118
11.1.4 Estimation des seuils métriques . . . 119
11.2 Perspectives . . . 119

A Langage de description des macrostructures physiques **127**
A.1 Grammaire : description des classes de documents . . . 127
A.2 Grammaire : description des documents spécifiques . . . 128
A.3 Exemple : description de la classe LRD_articles . . . 128
A.4 Exemple : un document spécifique de LRD_articles . . . 129

B Autres résultats de reconnaissance **131**

Liste des figures

2.1 Tableaux spécifiques conformes à la description génŕique 8
2.2 Description de documents en ODA. 11

3.1 De la production à la reconnaissance des documents imprimés. 13
3.2 Partition des angles dans la méthode des moindres carrés. 15
3.3 Segmentation par RLSA. 18
3.4 Exemple de profil de projection vertical. 19
3.5 Représentation d'une expression mathématique. 20
3.6 Modèle générique de documents 22
3.7 Une architecture de reconnaissance guidée par une decription. 22
3.8 Une architecture de reconnaissance orientée *Blackboard*. 23

4.1 Métrique des caractères. 26
4.2 Attributs des lignes. 28
4.3 Modèles de page usuels pour les revues. 30
4.4 Modèles de page usuels pour les magazines. 30
4.5 Modèles de page usuels pour les livres. 30
4.6 Modèle de page usuel pour la première page d'un journal. 30
4.7 Structure physique générique des documents composites. 31
4.8 Un exemple de macrostructure de page. 32
4.9 Un exemple de microstructure de texte. 33
4.10 Un exemple de microstructure d'un tableau. 34
4.11 Schéma de représentation d'une entité physique. 35
4.12 Représentation au moyen d'un arbre de la structure physique d'un tableau. 36
4.13 Exemples de rectangles structurants extraits d'un tableau. 37
4.14 Un exemple de séparateur horizontal. 37
4.15 Architecture du système de reconnaissance. 38

5.1 Topologies locales possibles entre entités physiques. 43
5.2 Schéma d'alignement horizontal. 44

6.1 Processus d'extraction des composantes connexes. 58
6.2 Partition initiale de l'espace des vecteurs caractéristiques. 60
6.3 Raffinement de la classification des lettres. 61
6.4 Rectangles structurants verticaux et horizontaux du document (Doc. 1). 62
6.5 Structure hiérarchique d'une expression mathématique. 64
6.6 Reconnaissance de documents textuels (Doc. 1). 67
6.7 Reconnaissance de documents textuels (Doc. 9). 68
6.8 Reconnaissance d'expressions mathématiques (Doc. 3). 69
6.9 Reconnaissance d'expressions mathématiques (Doc. 4). 70
6.10 Reconnaissance de tableaux (Doc. 6). 71
6.11 Reconnaissance de tableaux (Doc. 5). 72

6.12 Reconnaissance de blocs graphiques (Doc. 7). 73
6.13 Reconnaissance de blocs graphiques (Doc. 8). 74
6.14 Estimation des lignes de base à partir de composantes connexes. 74
6.15 Ordre de parcours des signes dans un bloc textuel. 75
6.16 Ordre de parcours des blocs dans une page. 76
6.17 La comparaison de profils n'est pas toujours une relation d'ordre. 77

7.1 Documents servant à illustrer l'estimation des seuils métriques. 79
7.2 Distribution des espaces inter-caractères. 81
7.3 Distribution des espaces entre les signes diacritiques et les lettres. 82
7.4 Distribution des espaces inter-mots. 83
7.5 Distribution des espaces entre lignes de base consécutives. 84
7.6 Distribution de la largeur des rectangles structurants horizontaux (Doc. 1). 85

8.1 Pages typiques des articles publiés dans la revue *LRD*. 88
8.2 Raffinement des documents composites. 88
8.3 Macrostructures génériques des pages typiques de la figure 8.1. 89
8.4 Décomposition en couches de la macrostructure 8.3.b. 90
8.5 Description de la macrostructure 8.4 au moyen des séparateurs génériques. 92
8.6 Schéma de substitution d'un séparateur flottant. 93
8.7 Macrostructure de la classe de pages 8.3.a. 94

9.1 Priorités initiales d'analyse et directions de recherche des séparateurs flottants. . . 100
9.2 Attributs d'un séparateur horizontal. 100
9.3 Illustration de la détermination des séparateurs (Doc. 1). 101
9.4 Illustration de la détermination des séparateurs (Doc. 2). 102
9.5 Types de structures mosaïques. 102
9.6 Catégories de rectangles structurants verticaux. 103
9.7 Rectangles structurants vérifiant une structure matricielle. 105

10.1 Distance entre deux arbres 108
10.2 Illustration de l'éloignement d'un bloc calculé par rapport au bloc de référence. . . 114

B.1 Autre résultat de reconnaissance (Doc. 2). 131
B.2 Reconnaissance de documents textuels (Doc. 1). 132
B.3 Autre résultat de reconnaissance (Doc. 10). 132
B.4 Reconnaissance de tableaux (Doc. 6). 133
B.5 Reconnaissance de blocs graphiques (Doc. 8). 133
B.6 Autre résultat de reconnaissance (Doc. 11). 134
B.7 Autre résultat de reconnaissance (Doc. 12). 134
B.8 Autre résultat de reconnaissance (Doc. 13). 135
B.9 Autre résultat de reconnaissance (Doc. 14). 135
B.10 Reconnaissance de documents textuels (Doc. 9). 136
B.11 Autre résultat de reconnaissance (Doc. 15). 136
B.12 Reconnaissance de tableaux (Doc. 5). 137
B.13 Reconnaissance de blocs graphiques (Doc. 7). 137

Liste des Tableaux

4.1 Unités typographiques usuelles en TEX. 26

10.1 Rapport en % des différentes classes de CCX calculées. 111
10.2 Seuils métriques estimés en *pixels*. 112
10.3 Distance entre les structures physiques calculées et celles escomptées. 113
10.4 Rapports en % du temps passé dans les primitives de reconnaissance. 115

Abstract

We present in this thesis a uniform approach to recognize the physical structure of printed documents that may contain various kinds of blocks: we call such documents *composite documents*. After an introduction to the subject of this thesis, we present first the state of the art in the field of document recognition. In this part, we present models and standards used to represent document structures. The remaining part of this report is devoted to our own research.

We start our study by modelling the *composite documents*. In this part, we describe the generic physical structure of composite documents as well as a modelling of white spaces (*background*) in such documents. Our systematical study of white spaces in documents leads, on one hand, to a set of rules that we use to guide microstructure recognition and, on the other hand, to a new language to describe a common layout of documents which are members of a same class. Such descriptions have been used to guide macrostructure recognition.

Based on the white spaces modelling, we have developed a uniform approach enabling us to recognize the physical structure of composite documents and to estimate automatically all parameters used. Our approach, combining both ascending and descending strategies, consists of two levels:

1. The microstructure recognition: this process is guided by rules we have developed to model the usual graphical aspect of microstructures. The recognition is based on a connected component classification. The analysis of textual blocks, as well as mathematical expressions, is realized by a hierarchically merging method, while tables are analysed by a hierarchically splitting method.

2. The macrostructure recognition: this process starts by segmenting a document into regions and finishes by segmenting the regions into blocks. The segmentation into regions is guided by a description of the document class. The region segmentation into blocks is realized by a hierarchically splitting method.

All metric parameters used for segmentation have been automatically estimated by a general method we have defined; this method is based on a statistical study of white spaces. The last part of this thesis presents a qualitative as well as a quantitative evaluation of the developed recognition methods. This report is concluded by a synthesis.

KEYWORDS: Document formatting Document Modelling Document representation standards Segmentation Document recognition Page Definition Language Structural analysis Statistical classification Tree metric

Résumé

Nous présentons dans cette thèse une approche d'analyse uniforme pour la reconnaissance de la structure physique des documents à contenu varié. Après une introduction à la problématique traitée dans cette thèse, nous présentons une synthèse des traitements usuels dans le domaine de la reconnaissance des documents. Dans cette synthèse, nous présentons aussi les structures usuelles de représentation et les normes régissant les documents. La suite est consacrée à la recherche que nous avons menée.

Nous commençons notre analyse par une modélisation des *documents composites* auxquels nous nous sommes intéressés. Dans cette partie, nous décrivons la structure physique générique des documents composites ainsi qu'une modélisation des espaces inoccupés (*le fond*) dans les documents. Notre étude systématique des espaces débouche d'une part, sur un ensemble de règles servant à guider la reconnaissance des microstructures et, d'autre part, sur un nouveau langage de description des classes de documents que nous utilisons pour guider la reconnaissance des macrostructures.

Pour la reconnaissance de la structure physique des documents composites, nous avons developpé une approche d'analyse qui est à la fois uniforme et capable d'estimer de façon automatique tous les paramètres utiles à son bon fonctionnement. Cette approche, fondée sur une modélisation des espaces et combinant une stratégie d'analyse ascendante avec une autre descendante, se compose de deux niveaux d'analyse :

1. La *reconnaissance des microstructures* est guidée par des règles que nous avons établies lors de notre modélisation des aspects graphiques usuels des microstructures. Cette reconnaissance commence par une *classification des composantes connexes*. L'analyse des blocs de texte et des expressions mathématiques est réalisée par une méthode de fusion hiérarchique suivant des règles. L'analyse des tableaux est réalisée par une méthode mixte, combinant une approche de fusion hiérarchique et une autre de découpe hiérarchique.

2. La *reconnaissance des macrostructures* complète celle des microstructures. Elle débute par une segmentation des documents en régions et finit par une segmentation des régions en blocs. La segmentation en régions est guidée par une description de la classe du document traité, alors que la segmentation d'une région en blocs est réalisée par une méthode de découpe hiérarchique.

Les seuils métriques de segmentation utlisés ont été estimés de façon automatique au moyen d'une méthode générale basée sur une étude statistique des espaces inoccupés. La dernière partie du mémoire est consacrée à l'évaluation aussi bien qualitative que quantitative des méthodes de reconnaissance développées ainsi qu'à une synthèse des travaux réalisés.

MOTS CLÉS : Formatage de documents Modélisation de documents Normes de représentation des documents Segmentation Reconnaissance de documents Language de définition de pages Analyse structurelle Classification statistique Métrique des structures arborescentes

Chapitre 1

Introduction Générale

1.1 La problématique

La communication, procédé d'échange d'informations et de connaissances, est fondamentale dans toute société. Malgré les poussées technologiques de ces dernières années en matière des télécommunications et de la production de documents électroniques, le document papier n'en reste pas moins un support primordial. En effet, les journaux, les lettres administratives, les magazines de tous genres, les revues scientifiques, les livres, sont autant de documents imprimés dont la production ne cesse d'augmenter.

1.1.1 Le document du point de vue informatique

Si l'on se réfère à la définition du dictionnaire LE PETIT ROBERT, un document est tout ce qui sert à instruire, à savoir : tout écrit servant de preuve ou de renseignement, tout ce qui sert de preuve de témoignage, toute pièce qui permet d'identifier une marchandise en cours de transport, etc. Du point de vue informatique, un *document* est généralement défini par tout ce que l'on produit, distribue, utilise ou garde lors d'un processus de communication écrite ou électronique; par conséquent, un document peut être qualifié de physique ou électronique. On distingue, en fonction de la nature du support appelé *médium* et de l'instrument avec lequel on écrit, plusieurs types de documents : les documents manuscrits, les documents imprimés, les documents électroniques, les documents audio, les documents vidéos, les documents multimédia combinant dans un seul document tous les autres types de média. Dans cette thèse, nous traitons le problème de la reconnaissance des documents imprimés. Cette reconnaissance a pour but de convertir des documents papiers dans une forme électronique.

1.1.2 L'intérêt de la reconnaissance des documents

Les innovations technologiques, au rang desquelles l'ordinateur et l'informatique, ont contribué à abaisser le coût de production des documents tout en améliorant l'efficacité du matériel ainsi que celle des systèmes informatiques utilisés pour la production. Souvent, ces innovations entraînent des changements dans la manière de représenter les documents, créant ainsi des incompatibilités avec les documents existants. Alors, dans le soucis, d'une part, d'uniformiser les vieux documents (papiers ou électroniques) avec les nouveaux et, d'autre part, de faire face à l'augmentation croissante du volume des documents, des systèmes informatiques de plus en plus performants sont requis. Il s'agit, par exemple :

- dans le domaine bancaire, de la lecture automatique des bulletins de versément,
- dans le domaine postal, du tri automatique des courriers,

- dans le domaine bibliothécaire, de la classification automatique de documents,
- dans le domaine éditorial, de la rétroconversion automatique de documents papiers à une forme électronique; c'est la récupération de l'existant,
- etc.

Ces traitements procurent des avantages considérables aussi bien sur les plans économique, organisationnel qu'écologique : partage de l'information, accès plus rapide à l'information, réduction des coûts de saisie de nouvelles informations, réduction des espaces de stockage, réduction des frais liés à l'infrastructure nécessaire pour l'archivage des documents papiers.

1.2 L'objet de cette thèse

1.2.1 Motivations

Les premiers travaux réalisés dans notre équipe de recherche ont traité le problème de la reconnaissance de la structure logique des documents imprimés. Cette reconnaissance consiste à déterminer l'organisation hiérarchique du contenu d'un document en partant de l'image digitalisée, au moyen d'un scanner, des pages de ce dernier. La reconnaissance est basée sur une *segmentation* des images qui consiste à partitionner ces dernières en blocs homogènes. De nos expérimentations, nous avons relevé la nécessité de disposer d'une méthode de segmentation plus robuste et plus fiable, pour pouvoir traiter des documents de contenu varié (textes, expressions mathématiques, tableaux, graphiques et photographies). Cette observation a été le point de départ de nos investigations dans ce domaine. En outre, notre intérêt pour la segmentation a été renforcé par le fait que pour un bon nombre d'applications, par exemple dans le domaine bibliothécaire, il n'est pas nécessaire d'avoir recours, ni à un système de reconnaissance optique des caractères, ni à un système de reconnaissance de structures logiques. C'est le cas, par exemple, dans la compression, l'archivage ou la classification des documents.

1.2.2 Objectifs

Nous désignons par *structure physique* la découpe hiérarchique en blocs homogènes perçue par l'œil lorsque l'on regarde une page de document. La reconnaissance de la structure physique d'un document est réalisée à partir de la segmentation de ses pages. L'étude des techniques courantes de segmentation que nous avons réalisée a révélé deux limites importantes :

1. S'il était possible de trouver des techniques spécialisées pour traiter des documents textuels, des expressions mathématiques, des tableaux ou encore des blocs graphiques, il était en revanche plus difficile d'intégrer celles-ci dans une approche d'analyse uniforme pour la reconnaissance des documents de contenu varié. En effet, l'utilisation de ces différentes techniques exigeait un filtrage manuel des blocs pouvant perturber le bon fonctionnement de la technique utilisée.

2. La fiabilité des techniques dépendait avant tout d'une bonne estimation des seuils métriques servant, soit à la découpe d'une entité en sous-entités plus homogènes, soit à la fusion des entités en une entité de niveau hiérarchique plus élevé. Le problème posé par le choix automatique de ces seuils n'était pas traité.

Dans cette thèse, nous contribuons à la résolution du problème posé par ces deux limites par le développement d'une approche d'analyse uniforme fondée sur une exploitation des espaces, n'impliquant aucun système de reconnaissance optique des caractères et dans laquelle les seuil métriques de segmentation sont estimés de façon automatique.

1.2.3 Plan

Les deux premiers chapitres de cette thèse sont consacrés à l'étude de l'état de l'art dans le domaine de la reconnaissance des documents imprimés :

- Dans le chapitre 2, nous présentons les structures ainsi que les normes régissant les documents.
- Dans le chapitre 3, nous présentons un appperçu des techniques de reconnaissance. D'une part, nous situons le problème de la reconnaissance de structures physiques par rapport au problème général de reconnaissance de documents. D'autre part, nous présentons une synthèse des techniques de segmentation et d'étiquetage présentées dans la littérature. Nous concluons le chapitre sur les limites de ces techniques, lesquelles limites seront à l'origine de cette thèse.

Notre travail de recherche à proprement dit, motivé par les lacunes montrées par les techniques courantes de segmentation et d'étiquetage, s'étend du chapitre 4 au chapitre 10.

- Dans le chapitre 4, nous présentons une modélisation des documents, dits documents composites, auxquels nous nous sommes intéressés. Nous concluons le chapitre par une description de notre architecture de reconnaissance qui se compose globalement de deux parties.

La première partie de notre architecture de reconnaissance, décrite du chapitre 5 au chapitre 7, est consacrée à la reconnaissance des microstructures.

- Dans le chapitre 5, nous établissons, après une étude exhaustive sur la topologie locale entre entités physiques, les règles de production gouvernant l'aspect graphique usuel des microstructures; il s'agit là, de la première originalité de notre travail.
- Dans le chapitre 6, nous présentons une approche mixte de reconnaissance fondée, d'une part, sur les règles de production établies au chapitre 5 et, d'autre part, sur une analyse des espaces inoccupés.
- Dans le chapitre 7, nous abordons de front un problème souvent passé sous silence dans les techniques de segmentation classiques : celui de la sélection automatisée des seuils métriques de segmentation; il s'agit, ici, de la deuxième originalité de notre travail.

La seconde partie de notre architecture de reconnaissance, décrite aux chapitres 8 et 9, est consacrée à la reconnaissance des macrostructures.

- Dans le chapitre 8, nous définissons un nouveau langage qui permet de décrire partiellement la macrostructure générique des documents à traiter puisqu'il n'existe, pour la reconnaissance des macrostructures, aucune méthode universelle. Ce nouveau langage tient son originalité du fait que les documents sont décrits par rapport aux espaces inoccupés (c.-à-d. le *fond* du document) et non pas par rapport aux objets eux-mêmes.
- Dans le chapitre 9, nous présentons notre approche de reconnaissance des macrostructures. Elle est guidée par une description générique de la classe du document à traiter; la description est donnée dans le nouveau langage que nous présentons dans le chapitre 8.

Dans le chapitre 10, nous présentons une évaluation aussi bien qualitative que quantitative de l'ensemble des méthodes que nous avons mises au point dans cette thèse. A cet effet, nous avons défini dans l'espace des structures physiques une distance métrique servant à comparer entre-elles deux structures physiques. Dans le chapitre 11, nous concluons ce mémoire par une synthèse des travaux réalisés et par les perspectives. Les annexes sont organisées comme suit : (A) grammaire complète du langage de description des macrostructures génériques suivie d'un exemple de description, (B) illustration d'autres résultats de reconnaissance suivis de nos remerciements et de notre curriculum vitae.

Chapitre 2

Structures et normes régissant les documents

Introduction

Lors de l'élaboration d'un document, l'auteur rédige son texte en terme d'enchaînement logique d'idées tandis que le typographe en réalise la présentation graphique qui aide le lecteur à mieux saisir le message de l'auteur. Généralement, une mauvaise mise en page conduira à une mauvaise interprétation du message délivré par l'auteur.

Comme tout traitement informatique, l'analyse de documents reposent sur un modèle des données à traiter, en l'occurrence celui des documents. En général, ce modèle doit être assez riche pour permettre des traitements variés. Les traitements les plus courants dans le domaine de l'analyse de documents sont la création (production), la modification, l'impression, la consultation, la transmission, le stockage, la recherche, la reconnaissance, etc. Le modèle doit être également adapté selon le point de vue de l'opérateur : (1) logique pour l'auteur, (2) physique pour le typographe et (3) syntaxique, sémantique ou pragmatique pour le linguiste. Dans la section 2.1, nous présentons une synthèse des structures usuelles pour la modélisation des documents et, dans la section 2.2, les normes régissant ces structures.

2.1 Structures de documents

On différencie, en général, l'aspect logique d'un document de son aspect physique. Cette dissociation permet à chaque intervenant de travailler sur le document à un niveau d'abstraction qui correspond à ses préoccupations. Il s'agit, par exemple, pour l'auteur, de la structure logique décrite à la section 2.1.1 et, pour le compositeur, de la structure physique décrite à la section 2.1.2. Dans la section 2.1.3, nous présentons les structures de représentation et, dans la section 2.1.4, la spécialisation des structures de documents. La section 2.1.5 est consacrée aux différents niveaux de granularité structurelle présents dans un document. Tous les modèles présentés dans cette section sont valables aussi bien pour la structure logique que pour la structure physique.

2.1.1 Structure logique

La *structure logique* décrit l'organisation hiérarchique du texte contenu dans un document au moyen d'*entités logiques* telles que les chapitres, les sections, les titres, les paragraphes, les notes, les citations, les formules, les tableaux, les cellules ou les graphiques. Les entités logiques sont des concepts servant à structurer le message de l'auteur; en retour, elles servent de repères au lecteur.

Cette abstraction offre l'avantage de rendre la description du texte contenu dans un document indépendant de tout support physique.

2.1.2 Structure physique

La *structure physique*[1] décrit au moyen d'*entités physiques* l'organisation hiérarchique des blocs typographiques composant les pages d'un document. Les entités physiques sont des concepts servant à structurer l'aspect graphique d'un document; elles décrivent la présentation graphique des entités logiques. Notons que plus d'une structure physique peuvent être dérivées d'une même structure logique; inversément, l'interprétation que l'on fait d'une structure physique peut également conduire à des structures logiques différentes.

2.1.3 Représentation

On distingue trois types de relations, aussi bien au niveau logique que physique, entre les entités d'un document :

1. la linéarité entre entités qui se suivent,
2. la hiérachie entre entités imbriquées l'une dans l'autre,
3. la collatéralité entre entités indépendantes, mais pouvant se référencer mutuellement.

Le choix d'un type abstrait, permettant de décrire l'organisation interne d'un document, est guidé par la nature des relations existant entre ses entités.

Structure de liste

Une structure de liste est utilisée pour traduire les relations de linéarité entre les entités d'un document. Elle est appropriée pour la représentation des documents conçus pour être lus de manière séquentielle; par exemple, un roman et, dans une certaine mesure, une liste bibliographique, un catalogue ou un dictionnaire.

Structure d'arbre

Une structure d'arbre est utilisée pour traduire les relations de hiérarchie (inclusion) existant entre les entités d'un document; c'est la structure de représentation la plus usuelle. Cette structure est aussi utilisée pour représenter les données dans certains éditeurs syntaxiques (Mentor, Centaur) ainsi que dans certains éditeurs de documents et de programmes (Tioga de Xerox) [1].

Structure de forêt

Lorsqu'il s'agit de représenter un document contenant des objets flottants, caractérisés par une position, a priori, inconnue dans le document, une structure d'arbre devient insuffisante. Une structure de forêt sert à modéliser, en plus du contenu principal, les objets flottants (tableaux, graphiques, notes) qui peuvent être référencés à plusieurs endroits du document. Dans la pratique, les objets flottants d'un document ainsi que le contenu textuel principal sont représentés au moyen de structures d'arbres indépendantes, mais connectées entre-elles. La structure de forêt est utilisée dans le système Grif [2] pour représenter les documents.

Structure de graphe

Une structure de graphe permet de gérer les renvois, les références croisées et les documents non séquentiels qui sont faits pour être lus, par exemple, à partir des index ou des tableaux. C'est le modèle des hypertextes et hypermédias [2].

[1] Dans certains modèles, la *structure physique* est aussi désignée par *structure graphique*.

2.1.4 Spécialisation

L'immense variété des documents rend illusoire, à notre avis, toute tentative de décrire au moyen d'un seul modèle la structure de tous les documents possibles. Par conséquent, il convient de regrouper, dans des *classes de documents*, les documents ayant une même structure. On désigne par *structure générique* les règles structurelles communes aux documents appartenant à une même classe et par *structure spécifique* la structure d'une instance de document conformément à une classe.

Structure générique

Une *structure générique* décrit la structure régissant l'ensemble des documents appartenant à une même classe. On parle de structure logique générique lorsque la description se rapporte au contenu des documents et de structure physique générique lorsqu'elle décrit la présentation graphique des documents. Par exemple, les numéros d'une même revue sont régis par une même structure générique, ce qui confère à ces derniers un aspect graphique similaire. La description 2.1 est un exemple de structure physique générique décrivant l'ensemble des règles de présentation devant régir l'aspect graphique des tableaux contenus dans les rapports publiés dans un institut donné. La structure générique peut servir à valider la structure d'un document particulier.

Un tableau est formé
I : d'une légende optionnelle, centrée et constituée
d'une référence suivie,
d'une ligne de texte en italique :
fonte : helvética,
II : d'une séquence d'au moins deux colonnes séparées par des distances horizontales régulières.
Chaque colonne est formée de cellules séparées par des distances verticales régulières.
Une cellule est
soit une séquence de colonnes (cf. II)
soit un bloc.

Description 2.1: Exemple de structure physique générique des tableaux.

Structure spécifique

Une *structure spécifique* décrit la structure régissant une instance de document appelée *document spécifique*. On parle de structure logique spécifique lorsque la description se rapporte au contenu du document et de structure physique spécifique lorsqu'elle décrit la présentation graphique du document. Les illustrations de la figure 2.1 sont deux exemples de tableaux spécifiques dont l'aspect graphique, quand bien même différent, est conforme à la structure générique de la description 2.1. Généralement, la présentation graphique d'un document spécifique est fondée sur sa structure physique générique.

2.1.5 Granularité

Généralement, on distingue deux niveaux de granularité complémentaires dans la structure des documents; il s'agit de la macrostructure et de la microstructure.

Macrostructure

La macrostructure décrit l'organisation hiérarchique d'un document à un niveau élevé. Dans le cas d'une structure logique, l'organisation décrite par une macrostructure part de l'entité logique représentant un document tout entier et s'arrête au niveau des paragraphes, des formules, des

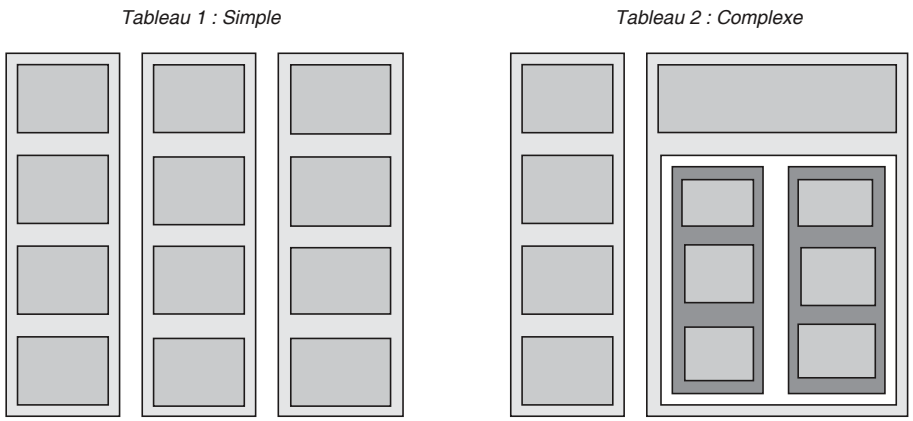


Figure 2.1: Tableaux spécifiques conformes à la description 2.1.

graphiques, des tableaux et des photographies. Ces entitées constituent les feuilles d'une macrostructure logique qui se compose d'entités logiques intermédiaires telles que : les chapitres, les références bibliographiques, les tables des matières, les sections, les sous-sections, etc. Dans le cas d'une structure physique, l'organisation décrite par une macrostructure part de l'entité physique représentant l'ensemble des pages du document et s'arrête au niveau des blocs représentant des paragraphes, des bouts de paragraphe, des formules, des graphiques, des tableaux ou des photographies. Ces blocs constituent les feuilles d'une macrostructure physique qui se compose d'entités physiques intermédiaires telles que : les pages, les régions, les blocs, etc.

Microstructure

La microstructure décrit l'organisation hiérarchique interne des feuilles d'une macrostructure. En effet, les systèmes de manipulation de documents peuvent s'intéresser à des structures plus fines. Par exemple, dans le cas de la structure logique, il s'agit des mots, des caractères, des changement de fontes, des emphases; dans le cas de la structure physique, il s'agit des lignes de texte, des mots, des signes, des composantes connexes. Les feuilles d'une microstructure logique représentent les caractères et celles d'une microstructure physique, les composantes connexes.

Pour certaines classes de documents, la microstructure est plus complexe que la macrostructure alors que pour d'autres, c'est plutôt l'inverse. Par exemple, pour un dictionnaire, les microstructures logiques, décrivant ses entrées, sont plus complexes que la macrostructure logique qui, elle, est constituée d'une liste d'entrées dans le dictionnaire. Inversément, pour un journal, la macrostructure physique, décrivant une mosaïque de blocs, est plus complexe que ses microstructures physiques qui, elles, sont formés de blocs textuels homogènes, constitués d'une séquence de lignes homogènes, et d'illustrations constituées de symboles graphiques.

2.2 Normes et architectures régissant les documents

La multiplication des systèmes de production de documents ainsi que l'évolution des imprimantes et autres périphériques d'affichage ont rendu difficile l'échange de documents sous leur forme électronique puisque, chaque système avait son propre format de représentation et chaque imprimante son propre jeu de commandes. Pour éviter la prolifération des formats de représentation des documents, des groupes indépendants ainsi que des organismes de normalisation tels que l'ISO[2] ont défini des formats en vue de faciliter les échanges entres systèmes et plateformes différents.

[2] International Standards Organization

Dans la section 2.2.1, nous présentons une synthèse des standards régissant le contenu ainsi que la présentation des documents. Dans la section 2.2.2, nous présentons un survol des systèmes permettant une manipulation plus riche de documents encore plus complexes.

2.2.1 Normes régissant les documents

SGML

La norme SGML, *Standard Generalized Markup Language*, est une standardisation ISO de GML défini chez IBM. Comme son nom l'indique, SGML est un langage de balisage qui sert à décrire la structure logique (générique et spécifique) des documents [3, 4]. Il permet, non seulement de décrire du texte, des formules et des tableaux, mais aussi d'intégrer des images et des graphiques dans un document. En revanche, elle ne permet pas de décrire, de manière naturelle, la structure physique des documents. La norme SGML définit avant tout une syntaxe à laquelle il est possible d'associer une sémantique; cette propriété n'est régie par aucune norme. Un document SGML est composé de trois parties pouvant chacune faire l'objet d'un fichier ASCII séparé :

1. Déclaration SGML : définit l'ensemble des caractères et balises autorisés aussi bien dans la définition des DTD que dans la description des documents SGML.

2. DTD (Document Type Definition) : désigne la définition d'un type de documents donnée par une description de la structure logique générique des documents appartenant à une même classe (cf. description 2.2).

3. Document SGML : désigne un document particulier balisé par les entités génériques définies dans la DTD à laquelle se rapport le document et par les symboles de marquage définis dans la déclaration SGML associée (cf. description 2.3).

Il existe une déclaration SGML par défaut. Les entités SGML peuvent être enrichies d'attributs, notamment d'attributs de présentation pour le formatage comme illustrer sur l'entité générique *paragraphe* de la description 2.2.

```
<!ENTITY    %doctype    "article"   --                          >
<!ELEMENT   article        --       (titre, auteur+, section+)  >
<!ELEMENT   titre          oo       (#PCDATA)                   >
<!ELEMENT   auteur         -o       (#PCDATA)                   >
<!ELEMENT   section        -o       (titresection, paragraphe+) >
<!ELEMENT   titresection   oo       (#PCDATA)                   >
<!ELEMENT   paragraphe     -o       (#PCDATA)                   >
<!ATTLIST   paragraphe   INDENT     NUMBER   1cm                >
```

Description 2.2: Un exemple de DTD en SGML

```
<!ENTITY article SYSTEM "/home/iiuf/giraf/azokly/articles/article.dtd" -- >
<article>
<titre> Normes régissant les documents
<auteur> Antoine AZOKLY
<section>
<titresection> Normes régissant le contenu < \titresection>
La norme ISO ...
< \section>
```

Description 2.3: Document conforme à la DTD 2.2

DSSSL

La norme DSSSL, *Document Style Semantics and Specification Language* complètant SGML, permet d'associer aux entités génériques constituant une DTD des traitements spécifiques. Elle permet avant tout de transformer une structure en une autre. Cette caractéristique fait de DSSSL un outil peuvant servir à enrichir une DTD de règles de présentation afin de permettre le formatage des documents. L'avenir de la norme DSSSL nous semble incertain puisqu'elle n'a toujours pas encore connu le succès qui lui a été prédit.

Interpress

Interpress est un langage de description de page développé par *Xerox* qui code ses instructions dans un format binaire ce qui le rend moins convivial à l'utilisation. En conséquence, il n'a pas connu le même succès que PostScript avec lequel il présente de fortes similitudes, en raison de leur origine commune [5, 6, 7]

PostScript

PostScript[3] est un langage de description de pages (PDL) né de *Interpress*. Il permet de décrire la structure physique spécifique des pages de document. Il s'agit d'un format ASCII défini pour piloter des imprimantes laser, domaine dans lequel il s'est imposé comme un standard de fait. Il est constitué de deux parties : (1) un langage de programmation de haut niveau fondé sur le principe d'une machine à pile [8, 9, 10] et (2) un interpréteur de programme PostScript qui prend en charge la composition des documents. PostScript propose de nombreuses fonctions pour créer et manipuler des objets typographiques (caractères, graphiques, etc.) ainsi que pour charger des fontes. Indépendant de tout périphérique, PostScript est supporté par de très nombreux types d'imprimantes et de photocomposeuses [11]. Ses récentes évolutions que sont la visualisation de documents (*DisplayPostScript*), la gestion des couleurs mais aussi la création du format PDF, sont les preuves de son succès dû à sa puissance et à sa flexibilité.

PDF

Le format PDF, *Portable Document Format*, est une spécification de Adobe qui permet de décrire les pages de documents de façon aussi précise que le permet PostScript dont il est d'ailleurs une évolution [12]. PDF tient son originalité du fait qu'il sert non seulement à décrire l'aspect graphique des documents, mais aussi et surtout à décrire les mises à jour, les annotations, les liens hypertextes, l'image réduite des pages et des signets. Un document PDF est stocké dans un fichier ASCII 7 bits ce qui lui assure une grande portabilité sur tout type de plateformes, indépendamment des périphériques (écrans et imprimantes) et de la résolution de ces dernières. Un fichier PDF contient, en plus du document, une description des fontes utilisées; ainsi, à l'issue de son transfert d'une plateforme à l'autre, les fontes manquantes peuvent être substituées de manière à préserver la qualité graphique originelle du document.

ODA

La norme ODA, *Open Document Architecture*, est un langage orienté objets qui permet de mélanger du texte avec des médias comme le son et la vidéo. Elle permet de décrire à la fois le texte et sa présentation puisque ODA distingue la structure logique de la structure physique d'un document [13, 4]. En ODA, le contenu des documents est partagé entre la structure logique et la structure physique comme l'illustre la figure 2.2 sur un exemple.

Le formatage s'appuie sur les styles de mise en page et les styles de présentation associés aux entités logiques. Les styles de mise en page indiquent comment créer des pages et comment les subdiviser en régions, puis les régions en blocs. Les styles de présentation associés aux feuilles

[3] PostScript est créé par la société Adobe en partant des travaux du Parc Xerox

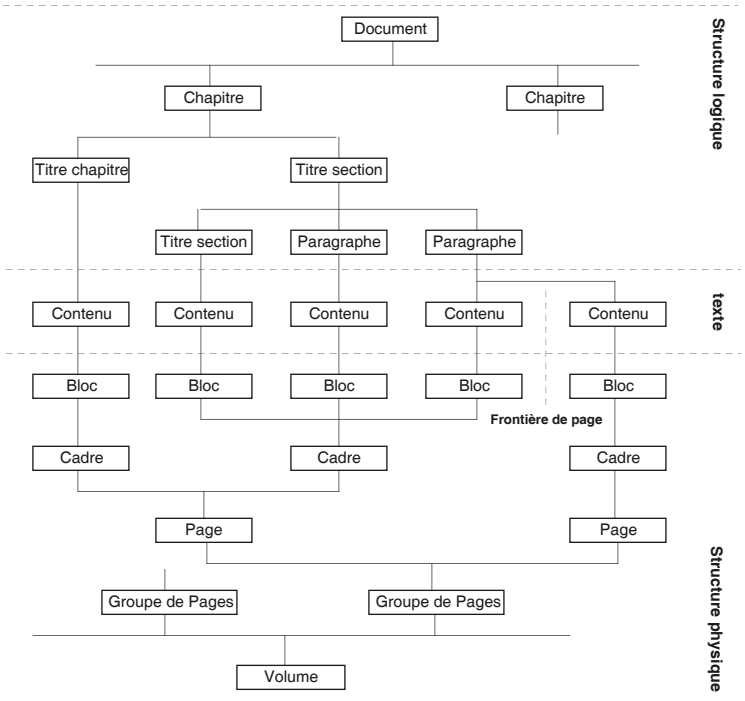


Figure 2.2: Description de documents en ODA.

indiquent comment formater des portions de contenu comme des formules, des tableaux ou des paragraphes. Les styles établissent une correspondance entre les entités logiques et les entités physiques. ODA se prète mal pour la description des tableaux et des figures; ces lacunes pourront être comblées à l'avenir par des architectures spécialisées.

2.2.2 Systèmes de manipulation de documents

En dépit des nombreuses avancées technologiques, il n'est souvent pas évident de transférer un document formaté sur un ordinateur de type A vers un autre ordinateur de type B et dont la plate-forme, la configuration ainsi que les applications natives ne sont pas compatibles avec l'ordinateur de type A. Et pourtant le besoin de pouvoir, sur sa propre machine, afficher, imprimer, annoter ou encore naviguer dans un document formaté sur un autre type de machines que la sienne, est de nos jours plus qu'une nécessité. Dans cette section, nous présentons brièvement deux de ces systèmes servant à manipuler des documents constitués de divers médias; il s'agit de *Acrobat* et de *OpenDoc*.

Acrobat

Le système *Acrobat*, développé par Adobe Systems Inc., est fondé sur le format PDF [14, 12]. D'une part, il sert à créer, à annoter et à transférer des documents et, d'autre part, il sert à gerer les différentes versions d'un même document. Le transfert de documents peut se faire suivant deux modes. Dans le mode qualifié de *distribution*, le document transféré peut être parcourru, affiché et imprimé. Dans celui qualifié d'*échange*, en plus des avantages du mode distribution, le document transféré peut être annoté. A cet effet, le système Acrobat regroupe trois applications complémentaires :

1. *Acrobat exchange* ou *Reader* sert à visualiser, à annoter, à imprimer ou à naviguer dans un document PDF.

2. *Acrobat Writer* sert à convertir en documents PDF des documents *QuickDraw* (format Macintosh) ou des documents *GDI* (format Windows). Cette application est utilisée comme un pilote d'impression qui au lieu d'envoyer le document à l'imprimante le redirige dans un fichier PDF. Ainsi, toute application capables d'imprimer des documents peut aussi facilement produire des documents PDF.

3. *Acrobat Distiller* sert à convertir au format PDF, indépendamment des plateformes, les milliers de documents existant déjà au format PostScript.

OpenDoc

OpenDoc, système concurrent de *Acrobat*, a été initialement conçu par Apple pour la manipulation de documents composés aussi bien de liens hypertextes que d'objets dynamiques, par exemple : des feuilles de calcul, des fenêtres de dialogue et des objets animés [15, 16, 17, 18]. Fondé sur le format *Bento*, il est muni d'un interface homme-machine et d'une librairie orientée objets. Son originalité principale réside dans le fait qu'il permet de combiner, au moyen d'un langage de scriptage, les différentes parties d'un document. Ceci permet de construire des applications interactives comme par exemple les didacticiels[4]. Le concept fondamental dans OpenDoc est celui des *parties* qui correspondent au concept de blocs dans les documents textuels classiques. L'aspect graphique d'une partie peut être d'une forme géométrique quelconque et son contenu de nature dynamique.

Conclusion

En résumé, la structuration des documents a permis d'obtenir une représentation de haut niveau favorisant des traitements variés qui s'étendent bien au-delà de la bureautique. Une étude comparative des normes SGML et ODA est présentée par Heather Brown dans [19]. Les normes publiées ou en cours d'élaboration, notamment PDF qui, à notre avis, possède à l'origine des caractéristiques intéressantes (ASCII 7 bits), devraient ouvrir un champ d'applications plus large sur les documents structurés [2, 20, 1].

Ces normes encourragent les producteurs à adopter une méthodologie aussi bien dans la structuration que dans le formatage des documents. En retour, l'échange de documents s'en trouve facilité et la difficulté de reconnaître ces derniers, en l'occurrence leurs structures physiques, s'entrouve diminuée. En effet, la reconnaissance de l'immense variété de documents n'est possible, à notre avis, que si l'on dispose, a priori, d'un modèle permettant de guider le système de reconnaissance. Dans notre système, ce modèle est défini par la structure physique générique des documents auxquels nous nous sommes intéssés (cf. chapitre 4).

[4] Selon le Petit ROBERT, *logiciel à fonction pédagogique*

Chapitre 3

Un apperçu des techniques de reconnaissance

3.1 Objectif

La reconnaissance de documents est une opération que l'on peut considérée comme inverse de la production, à ce titre, elle peut être qualifiée de *reverse engineering* (cf. figure 3.1).

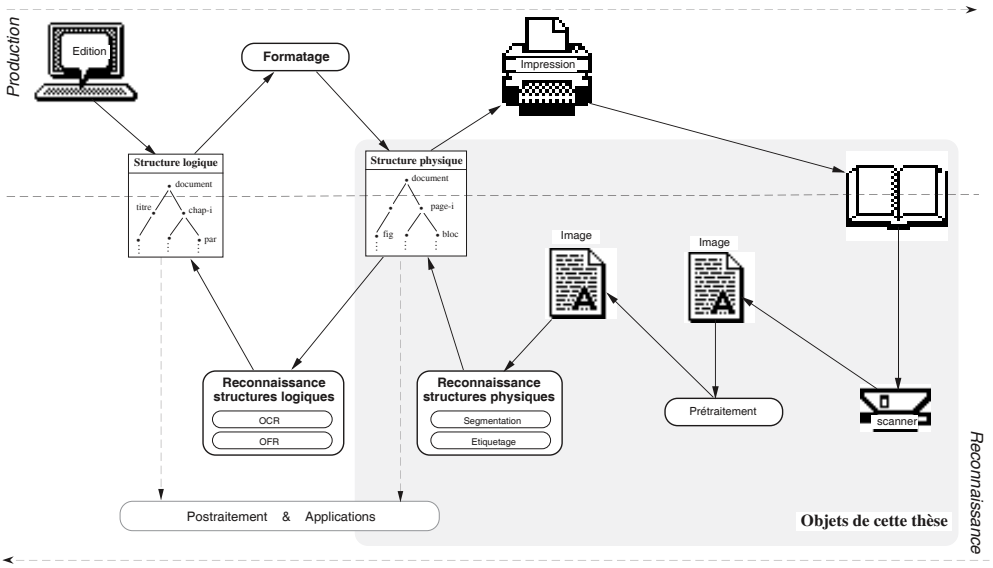


Figure 3.1: De la production à la reconnaissance des documents imprimés.

Production Opération qui consiste à transcrire le message ou l'idée d'un auteur sur un support physique, en l'occurrence papier. Cette opération comprend :

- l'*édition* qui concerne avant tout la structure logique,
- le *formatage* qui concerne essentiellement la structure physique,
- l'*impression* qui consiste à transcrire sur papier le résultat du formatage.

Reconnaissance Opération qui consiste à reconstruire une version électronique à partir d'un document imprimé. Cette reconnaissance implique plusieurs traitements :

- la *digitalisation* des pages au moyen d'un scanner,
- des opérations de *prétraitement* sur les images résultant de la digitalisation,
- la *reconnaissance de structures physiques* qui comprend la segmentation et létiquetage,
- la *reconnaissance de structures logiques* qui comprend l'OCR[1] et l'OFR[2],
- des opérations de *postraitement* comme, par exemple, la conversion du résultat de reconnaissance dans un autre format.

La reconnaissance de l'immense variété de documents n'est possible que si l'on dispose, a priori, d'un modèle permettant de guider les systèmes de reconnaissance. En réalité, il n'existe pas de système universel capable de reconnaître tout type de documents, compte tenue de la difficulté à décrire, dans un seul modèle, l'ensemble de tous les différents types de documents existants. Dès lors, la question qui se pose, est :

> comment modéliser les documents de sorte à prendre en compte, dans un modèle unique, les connaissances, a priori, aussi bien logiques que physiques des documents appartenant à une même classe.

Dans ce chapitre, nous décrivons les principaux traitements qui interviennent dans la reconnaissance des documents. Notre objectif principal est de mettre en évidence, d'une part, l'importance de la reconnaissance des structures physiques et, d'autre part, la limite des techniques usuelles présentées dans la littérature. Dans la section 3.2 nous décrivons brièvement les techniques usuelles de prétraitement. Dans la section 3.3, consacrée à la reconnaissance de structures physiques, nous présentons une synthèse des techniques courantes de segmentation ainsi que celles d'étiquetage. Dans la section 3.4, nous décrivons quelques approches de reconnaissance de structures logiques. Les limites des techniques de segmentation et d'étiquetage, que nous présentons à la section 3.5, ont été à l'origine de nos propres travaux. Le lecteur intéressé par les systèmes d'OCR peut se rapporter aux travaux de S. Kahan [21] et de Anigbogu [22] et, celui intéressé par l'OFR, aux travaux de A. Zramdini [23].

3.2 Prétraitement

La reconnaissance d'un document imprimé commence par la digitalisation de ses pages réalisée au moyen d'un scanner. Généralement, les images résultants de la digitalisation nécessitent, avant la phase proprement dite de reconnaissance, quelques opérations qualifiées de prétraitement et qui comprennent :

- l'élimination des bruits,
- l'estimation de l'inclinaison,
- et le redressement des images.

Dans cette section, nous ne présentons que les opérations de prétraitements les plus usuelles; il s'agit de l'estimation de l'inclinaison et du redressement des images. Nous ne traitons donc pas l'élimination des bruits qui, dans la pratique, peut être réalisée au travers des paramètres du logiciel pilotant la saisie optique. Le lecteur intéressé par ces techniques peut se référer aux travaux présentés dans [24, 25, 26, 27].

[1] Optical Character Recognition
[2] Optical Font Recognition

3.2.1 Estimation de l'inclinaison

L'inclinaison des images est provoquée essentiellement soit par un mauvais positionnement des pages lors de la saisie optique, soit par une mise en page fantaisiste et irrégulière de l'auteur. L'estimation de cet inclinaison est nécessaire pour certaines techniques de segmentation qui n'obtiennent de bons résultats que :

- si les images sont parfaitement redressées,
- ou connaissant l'angle d'inclinaison.

Dans la pratique, une inclinaison de 1° dans une image d'une largeur de $2500\,pixels$ induit un dénivelé de $44 pixels$, suffisant pour être une source de perturbation (cf. tableau 10.2). Dans cette section, nous présentons une méthode, dite des moindres carrés, pour l'estimation de l'inclinaison dans les documents. On relève également dans la littérature d'autres techniques fondées sur la transformée de Hough [28], sur la projection des composantes connexes [29, 30], etc.

Méthode des moindres carrés

Cette technique, que nous devons à Trincklin [31], consiste à estimer l'angle à partir d'un vecteur de points $V(i) = (x_i, y_i)$ formés des premiers pixels de couleur noire rencontrés en balayant, ligne par ligne, l'image de la gauche vers la droite. Ce vecteur V de taille n représente un nuage de points plus ou moins éloignés de la droite d_θ passant par l'origine $(0, 0)$ et inclinée d'angle θ recherché. Soit r le coefficient de corrélation linéaire des points $V(i)$ suivant la méthode des moindres carrés.

1. Si r est proche de 1 alors les points représentés par $V(i)$ sont alignés sur la droite d_θ; ainsi, connaissant l'équation de la droite d_θ, on peut déduire l'angle d'inclinaison recherché.
2. Autrement, s'il n'existe aucune corrélation entre les points $V(i)$, on reprend le calcul par dichotomie sur chaque moitié du vecteur V et, ainsi de suite, jusqu'à ce qu'on obtienne une corrélation sur une portion de $V_k \subset V$, ou que la cardinalité de V_k ne soit plus représentative.

Lorsqu'une corrélation a pu être trouvée sur un segment V_k de cardinalité n_k, alors l'angle θ_k correspondant est pondéré avec n_k. A la sortie de la dichotomie, on dispose d'une suite $S = \{\theta_k, n_k\}$ d'angles pondérés, ordonnée par rapport à θ_k. La suite S est ensuite partitionnée en regroupant les angles consécutifs de poids non-nuls, comme l'indique la figure 3.2.

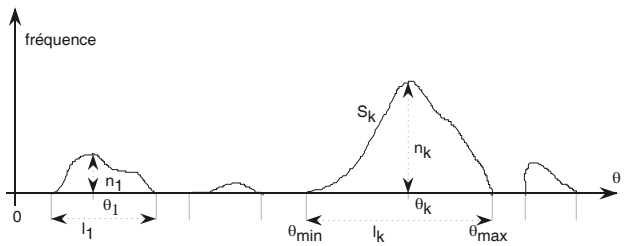


Figure 3.2: Partition des angles dans la méthode des moindres carrés.

Soit $S_i = (l_i, n_i(\theta_i))$ une partition de S dans laquelle l_i désigne la largeur de S_i et $n_i(\theta_i)$ le poids (d'angle θ_i) le plus fort dans S_i. L'angle d'inclinaison $\widehat{\theta}$ recherché est estimé par θ_k de la partition $S_k = (l_k, n_k(\theta_k))$ qui maximise la fonction :

$$E(Si) = l_i.n_i(\theta_i).$$

Pour accroître l'efficacité, la méthode gagnerait à être évaluée sur l'ensemble des composantes connexes au lieu des pixels.

3.2.2 Redressement

L'objectif visé par le redressement est d'obtenir une image avec un minimum de distorsion et de bruits. Les techniques classiques de redressement, basées sur la rotation euclidienne, présentent deux inconvénients majeurs dûs au caractère discrèt des images : (1) la non-bijectivité et, surtout, une trop forte dégradation dans les images redressées. Ces limites ont été à l'origine des nombreux travaux de recherche en cours dans le domaine de la géométrie discrète :

- la *rotation discrète par cercle* [32] qui consiste (1) à calculer, pour chaque pixel, la circonférence du cercle discret d'épaisseur 1 auquel il appartient puis (2) à décaler tous les pixels de ce cercle d'un nombre de pixels induit par l'angle de rotation.

- la *rotation discrète par droite* [32] qui consiste à faire correspondre les pixels d'un réseau horizontal (feuilletage horizontal de droites d'épaisseur 1) aux pixels du réseau oblique, obtenu à l'issue de la rotation des droites horizontales.

Le lecteur désireux d'en savoir plus sur les techniques de rotation discrète peut compléter sa lecture avec [33, 34].

3.3 Reconnaissance de structures physiques

La segmentation a longtemps été considérée, dans les systèmes de reconnaissance de documents, comme une primitives de traitement parmi tant d'autres. Les récentes applications (classification, archivage, compression, etc.) dans le domaine documentaire font de la segmentation un objectif de recherche en soit. Ainsi, de simples primitives de segmentation, on parle aujourd'hui de la reconnaissance de structures physiques [35, 36] au même titre que la reconnaissance de structures logiques. Cette reconnaissance consiste, d'une part, à déterminer une partition hiérarchique de l'image des documents traités et, d'autre part, à attribuer une étiquette logique à chacun des éléments de la partition que l'on désigne par entité physique. La reconnaissance est aussi caractérisée par le fait qu'elle n'implique aucun système reconnaissance optique de caractères. Dans la section 3.3.1, nous présentons une synthèse des techniques courantes de segmentation et dans la section 3.3.2 celle des techniques courantes d'étiquetage.

3.3.1 Segmentation

La segmentation a pour but de localiser, à partir de l'image digitalisée d'une page, les blocs qui composent cette dernière. Plus concrètement, le problème de la segmentation peut se formuler comme suit :

> Etant donnée l'image digitalisée d'une page, déterminer une partition géométrique de cette dernière de sorte à isoler tous les blocs qui la composent.

Dans cette section, nous présentons dans l'ordre les stratégies usuelles d'analyse, un survol des techniques de base puis celui des techniques avancées en matière de la segmentation.

Stratégies usuelles d'analyse

La partition d'une image peut se faire suivant deux stratégies d'analyse différentes à savoir, la stratégie descendante procédant par découpe hiérarchique ou la stratégie ascendante procédant par agglomération d'entités voisines. Ces deux stratégies sont généralement combinées lorsque l'une ou l'autre n'est pas satisfaisante.

Stratégie descendante Elle caractérise les techniques de segmentation procèdant par découpe récursive des images traitées. La récursion se poursuit jusqu'à ce que la composante physique la plus élémentaire (en général, une composante connexe) soit atteinte. Dans ce genre de techniques, la découpe d'une image est fondée, en général, sur une analyse des caractéristiques globales. Quand bien même de telles techniques sont fiables, leur coût, souvent excessif, peut devenir démesuré pour des méthodes admettant le retour en arrière (*backtracking*).

Stratégie ascendante Elle caractérise les techniques de segmentation procèdant par fusion hiérarchique des entités physiques. La fusion se poursuit jusqu'à ce que la structure complète (racine de la hiérarchie) de l'image traitée soit obtenue. Dans ce genre de techniques, la fusion est fondée, en général, sur une analyse des caractéristiques locales. Si de telles techniques sont plus efficaces que celles s'inscrivant dans une stratégie descendante, elles sont toutefois moins fiables que ces dernières à cause de la propagation des erreurs (locales) de fusion tout le long de la segmentation.

Techniques de base

RLSA *Run Lengh Smoothing Algorithm* est une technique due à Wong [37] et fondée sur un double lissage unidirectionnel de l'image à segmenter. Elle consiste à noircir, suivant une direction donnée, les segments de pixels blancs de longueur inférieure à un seuil s donné (cf. figure 3.3). La segmentation est alors obtenue en appliquant l'opérateur logique "and" ($\wedge$) sur les deux images résultant respectivement d'un lissage horizontal et d'un lissage vertical. La nature des blocs isolés est intimement liée au choix des seuils comme le montre la figure 3.3. Les seuils trop faibles provoquent une sur-segmentation alors que les seuils trop élevés provoquent une sous-segmentation. Les principales limites de cette technique sont :

1. le choix arbitraire des seuils de lissage,
2. sa sensibilité aux inclinaisons,
3. son inadaptation à segmenter des blocs graphiques, formules et tableaux.

Plusieurs variantes de la technique RLSA ont été présentées dans la littérature. Il s'agit par exemple, de celle proposée par Takashi dans [38] et qui consiste à réduire l'image d'un facteur donné ceci, dans le but de fusionner des entités proches les unes des autres. Nous avons nous-mêmes proposé une variante bi-directionnelle équivalente à la précédente et qui consiste à balayer l'image par une fenêtre au lieu de la balayer ligne par ligne [39].

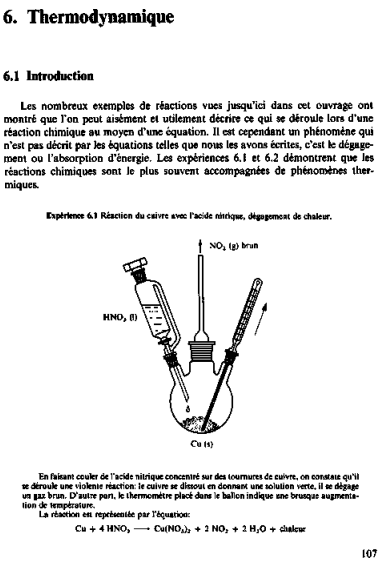
Découpe récursive Un grand nombre de techniques de segmentation procèdent par découpe récursive, alternant l'analyse des profils horizontaux avec celle des profils verticaux [40, 41, 42]. Un profil de projection est une accumulation des pixels noirs d'une image suivant un axe donné. La figure 3.4 illustre le profil de projection vertical qui sert, notamment, à la segmentation en lignes ainsi qu'à l'estimation des lignes de base. Les principales limites pour ce genres de techniques sont :

1. leur sensibilité aux inclinaisons,
2. leur inadaptation à segmenter des blocs mosaïques.

Cette technique de découpe récursive, due à G. Nagy, est aussi utilisée pour modéliser la structure des documents [43, 44] sous la forme d'un arbre XY dont la racine représente la totalité de l'image à segmenter.

Techniques avancées

Nous avons regroupé en trois grandes catégories les techniques de segmentation récemment présentées dans la littérature : (a) les techniques fondées sur une analyse spatiale, (b) les techniques fondées sur une analyse structurelle et (c) les techniques fondées sur une analyse spectrale.



6. Thermodynamique

6.1 Introduction

Les nombreux exemples de réactions vues jusqu'ici dans cet ouvrage ont montré que l'on peut aisément et utilement décrire ce qui se déroule lors d'une réaction chimique au moyen d'une équation. Il est cependant un phénomène qui n'est pas décrit par les équations telles que nous les avons écrites, c'est le dégagement ou l'absorption d'énergie. Les expériences 6.1 et 6.2 démontrent que les réactions chimiques sont le plus souvent accompagnées de phénomènes thermiques.

Expérience 6.1 Réaction du cuivre avec l'acide nitrique, dégagement de chaleur.

NO_2 (g) brun

HNO_3 (l)

Cu (s)

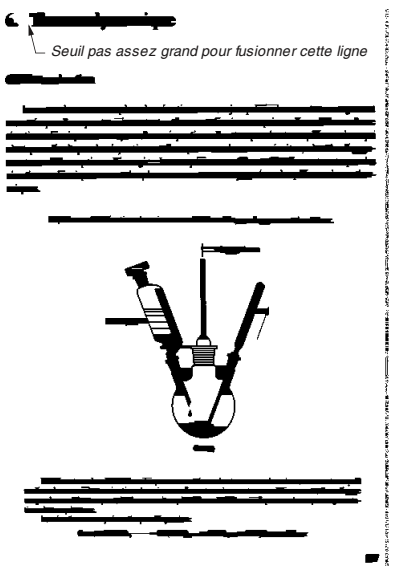
En faisant couler de l'acide nitrique concentré sur des tournures de cuivre, on constate qu'il se déroule une violente réaction: le cuivre se dissout en donnant une solution verte, il se dégage un gaz brun. D'autre part, le thermomètre placé dans le ballon indique une brusque augmentation de température.

La réaction est représentée par l'équation:

$$Cu + 4\,HNO_3 \longrightarrow Cu(NO_3)_2 + 2\,NO_2 + 2\,H_2O + \text{chaleur}$$

107

Image originale



Lissage horizontal avec $s = 12$

Figure 3.3: Segmentation par RLSA.

Analyse des espaces Cette catégorie regroupe l'ensemble des techniques de segmentation fondées sur une analyse des espaces inoccupés. La plus célèbre, due à Pavlidis et connue sous le nom de *white streams*, consiste à fusionner les segments d'espace blanc adjacents dans le but de construire des plages blanches qui serviront à estimer l'inclinaison des images ainsi qu'à segmenter ces dernières. D'autres techniques, inspirées des travaux de Pavlidis, ont été également présentées dans la littérature.

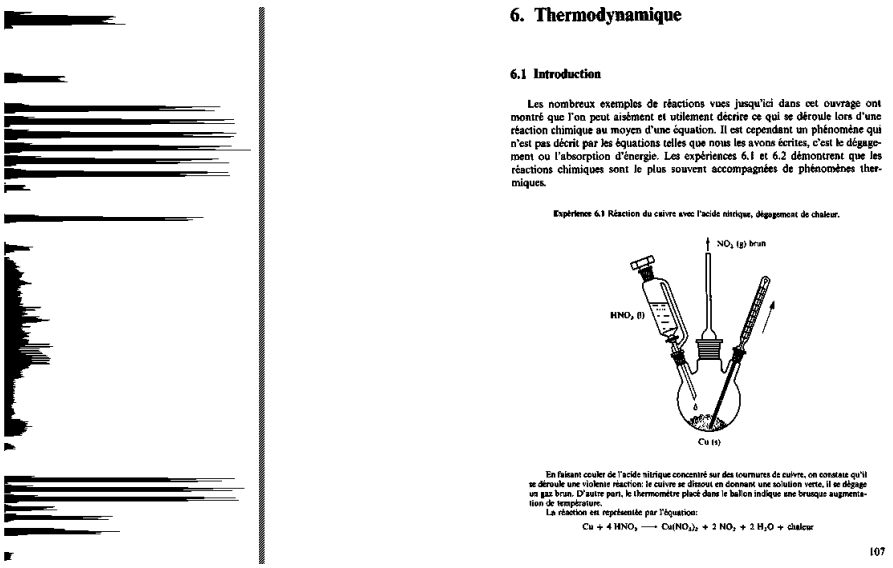
- Akindele [45] propose une technique dans laquelle les plages blanches sont supposées rectangulaires puisque les images traitées ont été préalablement redressées. La particularité de son approche réside dans le fait que les blocs isolés ont été approximés par des polygones de côtés parallèles aux bords des images traitées.

- Baird [46] propose une une technique de segmentation qui consiste à déterminer, dans un premier temps, les blocs en se basant sur une analyse des espaces puis, dans un second temps, la structure des blocs textuels au moyen d'une technique de découpe récursive. La particularité de son approche réside dans le fait qu'elle est non seulement indépendante de l'inclinaison mais aussi capable de segmenter des blocs mosaïques.

D'autres approches basées sur la recherche de séparateurs (espaces ou filets) sont décrites dans la littérature [47, 48, 38].

Analyse structurelle Cette catégorie regroupe l'ensemble des techniques de segmentation guidées par des règles structurelles décrivant le but à atteindre. Nous avons choisi d'en présenter une qui nous semble intéressante et assez représentative.

- Krishnamoorthy [49] présente une technique qui consiste à subdiviser les images suivant une description des profils de projection générique associés à chaque type d'entités pouvant composer les images traitées. Les profils sont décrits au moyen d'une suite alternant les plages noires et les plages blanches, chaque plage étant représentée par sa longueur.



6. Thermodynamique

6.1 Introduction

Les nombreux exemples de réactions vues jusqu'ici dans cet ouvrage ont montré que l'on peut aisément et utilement décrire ce qui se déroule lors d'une réaction chimique au moyen d'une équation. Il est cependant un phénomène qui n'est pas décrit par les équations telles que nous les avons écrites, c'est le dégagement ou l'absorption d'énergie. Les expériences 6.1 et 6.2 démontrent que les réactions chimiques sont le plus souvent accompagnées de phénomènes thermiques.

Expérience 6.1 Réaction du cuivre avec l'acide nitrique, dégagement de chaleur.

En faisant couler de l'acide nitrique concentré sur des tournures de cuivre, on constate qu'il se déroule une violente réaction: le cuivre se dissout en donnant une solution verte, il se dégage un gaz brun. D'autre part, le thermomètre placé dans le ballon indique une brusque augmentation de température.

La réaction est représentée par l'équation:

$$Cu + 4\,HNO_3 \longrightarrow Cu(NO_3)_2 + 2\,NO_2 + 2\,H_2O + \text{chaleur}$$

107

Figure 3.4: Exemple de profil de projection vertical.

Le lecteur désireux d'en savoir plus sur les approches structurelles de segmentation peut se rapporter aux travaux présentés dans [50, 51, 52, 53, 54, 55, 56].

Analyse par changement d'espace Cette catégorie regroupe l'ensemble des techniques de segmentation fondées sur une transformation globale des images en vue de déterminer soit les critères de découpe, soit les critères de fusion.

- *Docstrum* [57], due à O'Gorman, est une technique de regroupement hiérarchique fondée sur une analyse du graphe des k-plus proches entités voisines; l'auteur définit par *spectre* du document le graphe associé à ce dernier. Les nœuds du graphe sont initialement constitués de composantes connexes. *Docstrum*, appropriée pour la segmentation des blocs textuels, présente l'avantage d'être indépendante de toute inclinaison (globale ou locale).

D'autres méthodes de segmentation basées sur l'analyse des transformées de Fourier sont décrites dans la littérature [58, 59].

3.3.2 Etiquetage

La grande majorité des méthodes de segmentation ne prennent pas en compte le problème d'étiquetage des entités physiques. Ces dernières années, l'émergence de nouvelles applications, ne nécessitant pas un système de reconnaissance optique des caractères, a permis d'étendre le rôle de la segmentation qui désormais en plus de la partition des images, identifie la nature des blocs isolés.

Les permières méthodes d'étiquetage servent, en général, à distinguer un bloc textuel des autres blocs que l'on considère le plus souvent comme graphiques. Plus récemment, des méthodes plus fines ont été développées pour l'étiquetage des blocs de type particulier, à savoir : textes, expressions mathématiques, tableaux, graphiques et photographies. Dans cette section, nous présentons pour chacun de ces types, une synthèse des méthodes d'étiquetage présentées dans la littérature.

Texte

Les caractéristiques usuelles pour l'étiquetage des blocs textuels sont la taille des entités physiques [60, 61] et les profils de projection [37, 62]. Souvent, les lignes de texte peuvent être regroupées en blocs textuels en fonction de la régularité des espaces dans un profil de projection vertical.

L'étiquetage des mots et des caractères n'est souvent pas traité faute de pouvoir estimer, pour les premiers, l'intervalle de variation des espaces inter-mots et, pour les seconds, l'intervalle de variation des espaces entre un signe diacritique et son corps. Dans cette thèse, nous apportons une solution à ces limites grâce à une estimation automatique des seuils métriques régissant les blocs textuels (cf. chapitre 7).

Expressions mathématiques

La segmentation des documents scientifiques nécessite, en plus de l'analyse des blocs textuels, celle des blocs représentant des expressions mathématiques, des tableaux ou des graphiques. Contrairement aux documents purement textuels, les règles de formatage d'une expression mathématique sont complexes et très variées. Les systèmes courants de reconnaissance de documents n'aboutissent que si les expressions mathématiques sont, au préalable, filtrées manuellement pour être étiquetées par une méthode spécifique.

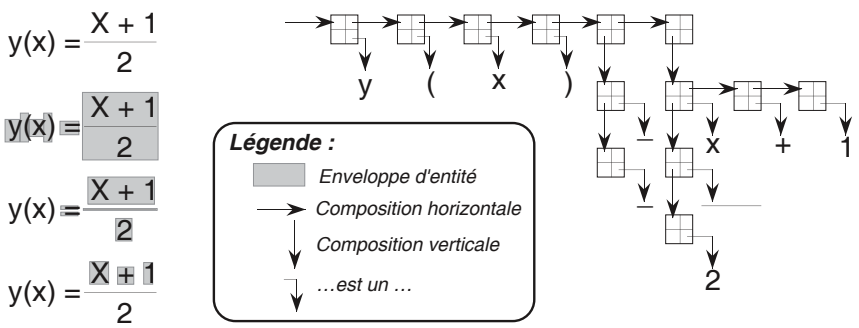


Figure 3.5: Représentation d'une expression mathématique.

Les méthodes classiques d'étiquetage d'expressions mathématiques s'appuient sur une modélisation de leur aspect graphique usuel au moyen d'un arbre XY [43] (cf. figure 3.5) qui s'apprête mieux pour les techniques de découpe récursive utlisées dans ce domaine [63, 64, 65]. Globalement, les méthodes usuelles d'étiquetage ne sont fiables que lorsque les expressions imbriquées, du style $\sqrt{x+y}$, sont soumises à un prétraitement et, que ces méthodes sont couplées à un système de reconnaissance optique de caractères.

Tableaux

Les tableaux sont abondamment utilisés (1) dans les documents commerciaux pour la facturation, (2) dans les documents bancaires pour les bilans d'exercice, (3) dans les documents scientifiques pour la synthèse des résultats, etc. Ils possèdent une structure régulière pouvant varier des plus simples aux plus complexes. Comme pour le formatage, l'étiquetage des tableaux est fortement guidé par leur aspect graphique.

- Kojima [66] présente une méthode d'étiquetage de tableaux guidé par la connaissance, à priori, des emplacements des filets dans le tableau à segmenter.
- Chandran [67] présente une méthode d'étiquetage de tableaux qui suppose la présence d'un nombre minimum de filets, suffisant pour déterminer le contour des tableaux.

- Watanabe [68] présente une méthode de reconnaissance de tableaux guidée par le modèle générique du tableau traité. Le modèle est décrit par un graphe de cellules régissant l'ensemble des tableaux d'une même classe.

Toutes ces méthodes d'étiquetage recherchent principalement la présence d'espaces et de filets qui ont servi à renforcer la structure matricielle des tableaux.

Graphiques

Dans le cas d'un document technique, la structure physique à extraire est souvent composée de plusieurs couches :

- une couche graphique qui comporte l'essentiel de l'information contenu dans le document et peut être subdivisée en plusieurs sous-couches distinctes selon des critères purement structurels ou géométriques (traits forts, traits fins, hachurage, etc.); ces sous-couches sont habituellement converties en primitives graphiques par des techniques de vectorisation [69, 70, 71, 72];
- une couche textuelle, généralement, constituée de mots, de nombres et de cotations dont la position, par rapport au graphique, indique à quelle partie du graphique se rapportent les cotes ou les légendes;
- d'autres couches pouvant contenir, par exemple, des informations additionnelles.

Une synthèse des techniques d'analyse de documents techniques est présentée par Tombre et Belaïd [72].

Dans le domaine de l'analyse des documents composites caractérisés par un contenu varié, nous ne connaissons pas de travaux ayant pour but d'extraire des blocs graphiques la couche textuelle qui les compose, comme c'est le cas dans l'approche que nous préconisons dans cette thèse (cf. section 6.3.5).

3.4 Reconnaissance de structures logiques

La reconnaissance de la structure logique d'un document consiste à reconstruire aussi bien son contenu que l'organisation hiérarchique de ce dernier à partir de l'image digitalisée des pages du document. Dans cette section, nous présentons deux applications qui se distinguent l'un de l'autre par l'approche d'analyse utilisée d'une part, et par la nature des documents traités d'autre part.

3.4.1 Approche syntaxique

Dans [73], R. Ingold présente une approche de reconnaissance qui, en 1988, a contribué à ouvrir une nouvelle voie de reconnaissance intégrant le modèle des documents à reconnaître (cf. la figure 3.6).

La reconnaissance d'un document est alors guidée par une description de la classe à laquelle ce dernier appartient. Cette description est donnée dans un langage formel dont la compilation génère un graphe d'analyse. Ainsi, la reconnaissance de la structure logique spécifique d'un document revient à rechercher dans le graphe un chemin partant d'un nœud initial et aboutissant à un nœud terminal. Lors de la recherche d'une solution dans le graphe d'analyse, un nœud est soit rejeté, soit accepté, sans qu'il ne soit possible d'avoir une réponse nuancée. Cette faiblesse a été à l'origine des travaux de mon collègue de recherche Hu Tao qui, dans sa thèse [74], a présenté une évolution du système en se basant sur la programmation dynamique et sur la théorie de la logique floue. La nouveauté a été d'attribuer des coûts aux nœuds ainsi qu'aux transitions dans le graphe d'analyse afin de permettre une analyse plus fine. La figure 3.7 résume l'arichitecture de ce nouveau système.

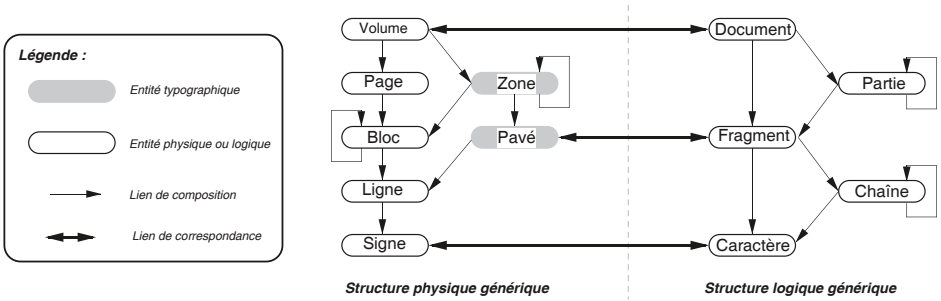


Figure 3.6: Modèle générique de documents (tirée de [73]).

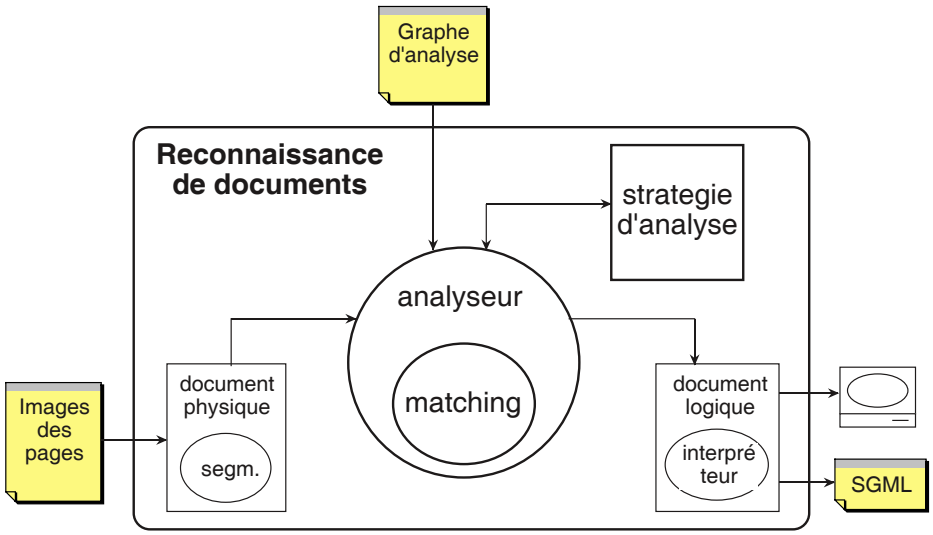


Figure 3.7: Une architecture de reconnaissance guidée par une decription (tirée de [74]).

3.4.2 Approche IA

Contrairement à l'approche préconisée dans l'application précédente, Y. Chenevoy, dans sa thèse [75], présente un modèle de reconnaissance dans lequel la structure physique est privilégiée par rapport à la structure logique. La structure physique est décrite au moyen de constructeurs inspirés de la philosophie ODA et de séparateurs. Un exemple de constructeur est celui permettant de décrire une structure mosaïque, ce qui n'est pas évident à réaliser en ODA. Les séparateurs servent à délimiter les entités physiques; ils sont constitués d'espaces, de segments de droite et de signes de ponctuation.

Une originalité de son approche réside dans le fait que les séparateurs sont traités de la même façon que les entités physiques. La figure 3.8 schématise l'architecture de son modèle de reconnaissance dans lequel les sources de connaissances coopèrent par l'intermédiaire d'une mémoire commune, caractéristique des architectures *blackboard*. Dans ce système, la segmentation est réalisée par un ensemble de *spécialistes*, terminologie utilisée pour désigner, entre autres, les techniques de segmentation et de prétraitement utilisées : RLSA, profil de projection, fusion d'entités, extraction des composantes connexes, séparation entre texte / image / graphique, mesure d'inclinaison, redressement, etc.

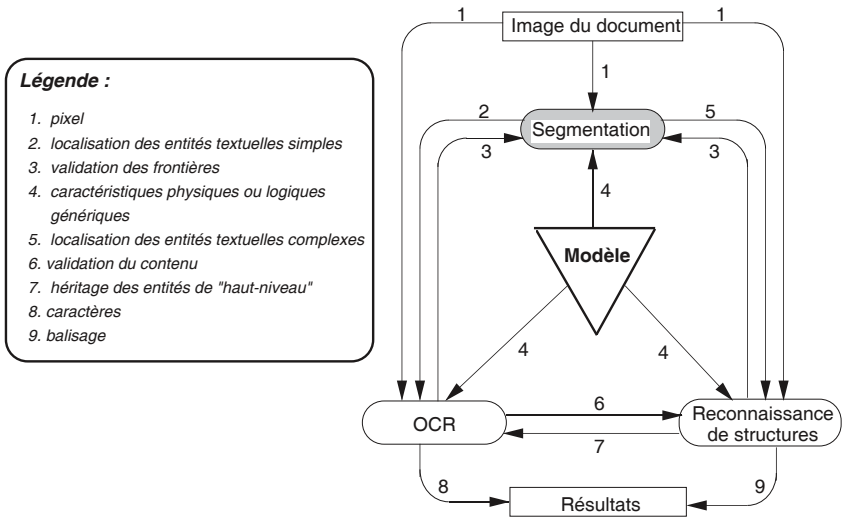


Figure 3.8: Une architecture de reconnaissance orientée *Blackboard* (tirée de [75]).

Le lecteur désireux d'en s'avoir plus sur les systèmes de reconnaissance de documents peut se référer aux *proceedings de ICDAR*[3] [76, 77, 78] qui constitue une réffence importante dans le domaine.

3.5 Les limites

Il ressort de notre synthèse des techniques de segmentation et des méthodes détiquetage deux lacunes très importantes :

1. l'estimation des seuils métriques régissant les blocs textuels, nécessaire pour une segmentation plus fiable, reste un problème ouvert;
2. quand bien même il est possible de trouver des méthodes d'étiquetage spécifiques pour chaque type de blocs (texte, expressions mathématiques, tableaux, graphiques, photographies), leur intégration dans une approche uniforme, pour reconnaître la structure physique des documents à contenu varié, reste également un problème ouvert.

Ces lacunes ont été à l'origine de cette thèse qui contribue à leur résolution par une approche uniforme de reconnaissance de structures physiques fondée sur une analyse des espaces inoccupés dans les documents traités.

Conclusion

Pour étendre l'application des systèmes de reconnaissance aux documents composites de contenu varié (texte, tableaux, expressions mathématiques, graphiques, photographies), il est indispensable, à notre avis, de disposer de techniques de segmentation plus fines; d'où l'intérêt de cette thèse.

Dans les chapitres qui suivent, nous présentons un système de reconnaissance de structures physiques, indépendant de tout système de reconnaissance optique des caractères, capable d'estimer de manière automatique les seuils métriques nécessaires pour son bon fonctionnement et applicable sur des documents composites.

[3] International Conference on Document Analysis and Recognition

Chapitre 4

Modèle des documents composites

4.1 Introduction et définitions

L'objet proprement dit de cette thèse commence avec ce chapitre qui présente le modèle de la structure physique des documents, dits composites, auxquels nous nous sommes intéressés. L'immense variété des différents types de documents existant, nous fait penser que l'on ne peut en déduire, de façon automatique et avec précision, une structure physique que si l'on dispose d'un modèle a priori permettant de guider la stratégie de reconnaissance. Reste à savoir comment modéliser, aussi bien les documents que les approches de reconnaissance, et jusqu'où aller dans la reconnaissance. L'idéal serait d'envisager un modèle particulier pour chaque document spécifique à reconnaître ce qui, de toute évidence, n'est pas réaliste. Non seulement une telle approche rendrait trop fastidieuse la mise en œuvre d'une reconnaissance, mais en plus, elle ne permettrait pas de prendre en compte le caractère générique de certains documents : par exemple, le fait que les numéros d'une telle revue ou d'un tel journal aient une structure graphique semblable. Nous pensons qu'une approche plus réaliste consiste à définir un modèle commun pour l'ensemble des documents dont la structure graphique présente globalement des similitudes. L'ensemble de tels documents constitue une *classe* au sein de la famille des documents, dits *composites*, auxquels nous nous sommes intéressés :

> Un *document composite* est un document imprimé composé essentiellement de blocs textuels et pouvant contenir des expressions mathématiques, des tableaux, des graphiques et des photographies.

Le modèle des documents composites que nous proposons est fondée sur les caractéristiques typographiques qui se dégagent de l'aspect graphique usuel de ces derniers. Ces caractéristiques sont décrites dans la section 4.2 traitant du formatage conventionel des documents. Dans la section 4.3, nous présentons la structure physique générique des documents composites, puis dans la section 4.4, une modélisation des délimiteurs d'entités physiques. Dans la section 4.5, nous résumons l'architecture globale de notre système de reconnaissance. Cette dernière section sert d'introduction aux chapitres suivants.

4.2 Formatage conventionel

Dans cette section, nous présentons les fondements typographiques nécessaires à notre modélisation de l'aspect graphique des documents composites. Dans la section 4.2.1, nous énumérons l'ensemble des attributs usuels de formatage et, dans la section 4.2.2, le formatage conventionnel des différentes entités constituant un document composite. Nous terminons à la section 4.2.3 par une synthèse des pratiques usuelles dans la définition des modèles de pages.

ractérisant les ca
, on désigne par *chasse* la largeur
de la surface imprimable appelée *œil* se trouvent
es qui empêchent deux caractères consécutifs de se toucher (2) les
gnes consécutives de se toucher. On désigne par *x-height* la hauteur de la lettre
ans une fonte donnée, par *hampe* la différence de hauteur entre la lettre "x" et une let
me "h" ou "l" et par *jambage* la différence de hauteur entre la lettre "x" et une lettre comme
" ou "p".

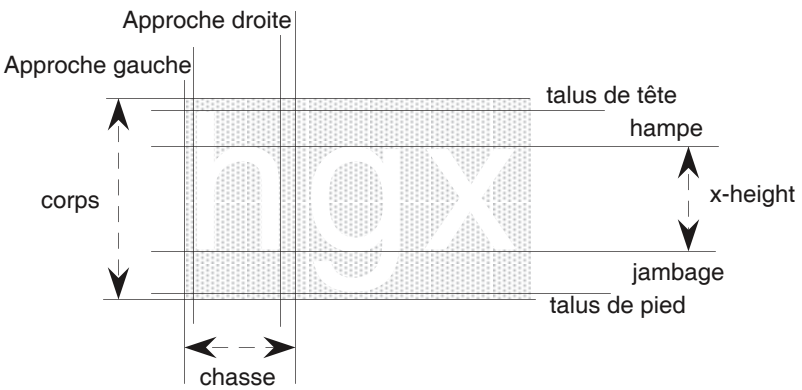


Figure 4.1: Métrique des caractères.

Il n'existe aucune relation universelle (indépendante de la fonte) entre les valeurs du *x-height*, de la *hampe* et du *jambage*. On estime, pour une fonte donnée, la valeur du x-height entre 40% et 60% la valeur du corps. Le lecteur désireux d'en savoir plus sur la métrique des caracères peut se référer au livre de Peter Karow intitulé *Digital Formats for Typefaces* [80] et à l'article de Jacques André intitulé *Font Metrics* [79].

Attributs de blocs

Les attributs usuels de formatage sont : les marges, les longueurs de lignes, l'indentation, l'espace avant et après le bloc des entités d'une même classe, le mode de justification du contenu textuel, les fontes utilisées, l'interligne, etc. Chaque classe d'entités logiques est régie par des prédicats de présentation qui indiquent (1) si le bloc représentant une de ses instances peut être réparti sur plusieurs pages consécutives, (2) si un saut de page après ce bloc est admis, (3) si ce bloc peut débuter une nouvelle page, etc.

Les attributs de formatage ne suffisent pas pour décrire à eux seuls la structure graphique des blocs; les relations structurelles de voisinage entre les blocs et à l'intérieur de ces derniers doivent aussi être prises en compte.

4.2.2 Règles de présentation

But

Le formatage est un traitement qui prend en charge les numérotations, la gestion des références croisées et références bibliographiques, la constitution des tables d'index, des tables de matières et des tables de figures, mais aussi et surtout, la découpe en lignes, en colonnes et en pages au moyen d'une répartition des espaces. Si l'on en croit Roger Chartier, dans le supplément Liber n° 1 du Monde :

> *La signification d'un texte ne se déduit pas de ses seules ressources verbales, mais aussi de ses dispositifs graphiques, de l'écriture à la mise en page, du format au support.*

Les règles de présentation des différentes classes d'entités logiques sont souvents décrites au moyen de boîtes imbriquées appropriées pour décrire aussi bien des blocs textuels que des tableaux, des formules et des graphiques. L'emboîtement s'appuie sur la hiérarchie de la structure logique, ce qui facilite un formatage incrémental comme c'est le cas dans *Grif* [81]. Nous avons observé qu'une bonne mise en page, caractérisée par un meilleur rendu, est une conjonction de quatre paramètres impliquant aussi bien les objets que les espaces :

1. la taille des objets imprimés : ex. lettres, symboles, filets, composantes graphiques;
2. le gris typographique des objets : ex. couleur d'encre, fonte, homogénéité visuelle d'un bloc;
3. l'agencement des objets : ex. composition horizontale d'une ligne par des mots;
4. l'espace entre objets : ex. espace entre mots dans une ligne.

Dans la suite de cette section, nous décrivons les règles conventionelles de présentation apliquées aux principales entités d'un document composite.

Textes

Un texte est organisé en paragraphes, entités logiques dont le formatage donne lieu à un bloc pouvant être soit réparti à cheval entre plusieurs pages consécutives lorsque le bloc est trop grand pour tenir dans l'espace restant sur la page courante, soit coupé en sous-blocs par la présence d'objets flottants comme une figure. Il peut donc être associé à un paragraphe plusieurs blocs dont la mise en évidence se manifeste, généralement, soit par l'indentation de sa première ligne de texte, soit par un léger retrait de cette dernière.

Un *bloc textuel* est composée de lignes dont les *lignes de base* sont séparées les une des autres d'une distance régulière et constante appelée *interligne* [82]. La *ligne de base*, comme l'indique la figure 4.2, désigne la ligne virtuelle sur laquelle reposent les lettres sans jambage d'une ligne de texte. La *césure* est un procédé qui consiste à couper des mots trop longs en fin de ligne suivant

des règles données par l'éditeur ou en consultant un dictionnaire. E
équilibre visuel des blocs textuels. Un bloc textuel est dit (1) *justifié à*
sont alignées par rapport à la marge gauche, (2) *justifié à droite* quand se
rapport à la marge droite, (3) *centré* quand chaque ligne est centrée par ra
englobant et finalement (4) *justifié* quand chaque ligne est justifiée à la fois

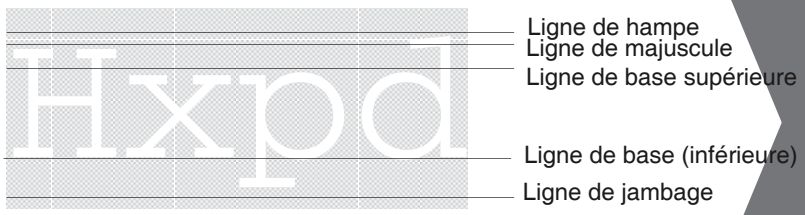


Figure 4.2: Attributs des lignes.

Une *ligne de texte* est une chaînes de mots alternés avec des espaces que nous désignons par *inter-mots*. Ces espaces sont constants dans les blocs textuels justifiés soit à gauche, soit à droite. Pour des textes justifiés, le solde de l'espace restant après remplissage d'une ligne est distribué entre les mots de sorte à préserver une égalité apparente des espaces inter-mots, aussi bien à l'intérieur de la ligne que dans le bloc auquel la ligne appartient.

Un *mot* est une chaîne de caractères alternés avec des espaces que nous désignons par *inter-caractères*. Indépendant du mode de justification, l'espace inter-caractère dérive à la fois des *approches* et du *crénage* des caractères impliqués. Le crénage consiste à prédéfinir, pour chaque paire de caractères d'une fonte, un espace qui préserve son harmonie visuelle lorsque cette paire se retrouve dans un mot.

En fonction de la culture typographique en vigueur, un système de formatage peut mettre plus ou moins d'espace après et avant un signe de ponctuation. Pour faciliter la perception de l'enchaînement des composantes textuelles, la convention exige que les espaces inter-lignes soient plus grands que les espaces inter-mots qui, à leur tour, doivent être plus grands que les espaces inter-caractères [82].

Formules

Dans les premiers éditeurs d'expressions mathématiques, une formule était considérée essentiellement comme une mosaïque de caractères et de symboles juxtaposés dans le plan. Les formules ne possédaient pas une réelle structure hiérarchique, mais plutôt une structure graphique dérivée directement de leur structure sémantique. Les systèmes plus récents fondent le formatage des expressions mathématiques sur la structure hiérarchique qui se dégage de leur structure logique. L'aspect graphique d'une formule est défini de sorte que la structure mathématique apparaisse clairement au lecteur. Quand bien même les expressions mathématiques ont une structure sémantique assez forte, certains auteurs manquent parfois de systématique dans leur formatage. Ainsi, il n'est pas rare de trouver des formules qui s'étalent sur plusieurs lignes.

Tableaux

Tout le monde s'accorde à dire que les tableaux sont ce qu'il y a de plus difficile à modéliser [83, 84]. La plupart des systèmes guident la saisie des tableaux par leur structure graphique. On utilise pour cela une structure matricielle qui sert à quadriller les cellules du tableau. La lisibilité des tableaux dépend fortement de la régularité des espaces servant à délimiter leur structure interne en lignes et en colonnes. Les *filets* dans les tableaux servent avant tout à renforcer la séparation entre les cellules.

En pratique, les tableaux d'un même document auront un même formatage ce qui facilite leur lisibilité. Dans certains systèmes de formatage, les tableaux sont traités comme des objets flottants caractérisés par le fait que leur emplacement définitif est déterminé selon des critères d'esthétique qui échappent au contrôle de l'auteur.

Graphiques

Les graphiques, comme les tableaux et autres illustrations, sont en général traités comme des objets flottants souvent placés soit en haut, soit au bas des pages. Les composantes graphiques sont des segments de droite, des cercles, des carrés, du texte, etc. Il existe des graphiques fortement structurés et conformes à des modèles bien définis. C'est le cas des documents techniques pour lesquels la structure physique est composée de plusieurs couches [72] :

- une couche graphique comportant l'essentiel de l'information contenue dans le document et pouvant être séparée en plusieurs sous-couches distinctes par des critères purement structurels et géométriques (traits forts, traits fins, hachurage, etc.);
- une couche de texte qui généralement se présente sous la forme de mots, de nombres ou de cotations, dont les positions par rapport au graphique indiquent à quelle partie du dessin se rapportent ces objets;
- d'autres couches éventuelles dédiées à des informations additionnelles.

Photographies

Il est plus difficile de définir la structure physique d'une photographie ainsi que de certains dessins. Les photographies sont généralement représentées par un *bitmap* directement composés de pixels[1].

4.2.3 Modèle de pages

Généralement, les compositeurs créent, pour chaque famille de documents, des modèles de pages qui servent à guider leur mise en page. Ces modèles décrivent une partition élémentaire des pages qui repose sur l'utilisation des *grilles* permettant, d'une part, une utilisation cohérente et consistante des espaces et, d'autre part, l'obtention d'une efficacité dans le processus de formatage [85, 86]. Les grilles servent à prédéfinir l'emplacement des marges, des colonnes, des titres courants, des numéros de page, des illustrations, du contenu et d'autres entités répétées sur toutes les pages. Par exemple, aux numéros d'une même revue, les modèles confèrent un aspect graphique global qui aide le lecteur à les identifier. Plus souvent, au lieu d'une lecture linéaire, le lecteur a plutôt besoin de repères pour accéder plus rapidement à certaines parties du document. Il s'agit par exemple des tables de matières, des index, des résumés, des titres courants, des numéros de sections, des paragraphes, des numéros de pages, des noms des auteurs, etc. La figure 4.3 est un exemple de modèle usuels pour les pages de revues scientifiques, la figure 4.4 pour les pages de magazines, la figure 4.5 pour les pages de livres et figure 4.6 pour les premières pages de journaux.

Un modèle de pages est souvent enrichi par la *géométrie des pages* définie par un certain nombre d'attributs au rang desquels figurent la longueur des lignes de texte, le mode de justification et le nombre de colonnes. On apprend, par exemple, dans [82] que la longueur des lignes de texte est d'environ 42 caractères pour la Bible de Gutenberg alors que, pour un livre normal, la longueur idéale se situe entre 55 et 60 caractères, ce qui correspond en moyenne à 10 mots. Par contre, pour des journaux, la longueur des lignes est d'environ 40 caractères avec une moyenne de 5 mots. Dans les systèmes de formatage, les modèles sont donnés soit au moyen de règles descriptives regroupées dans des feuilles de style, soit au moyen de gabarits servant de maquette.

[1] Un pixel désigne la composante atomique dans une image

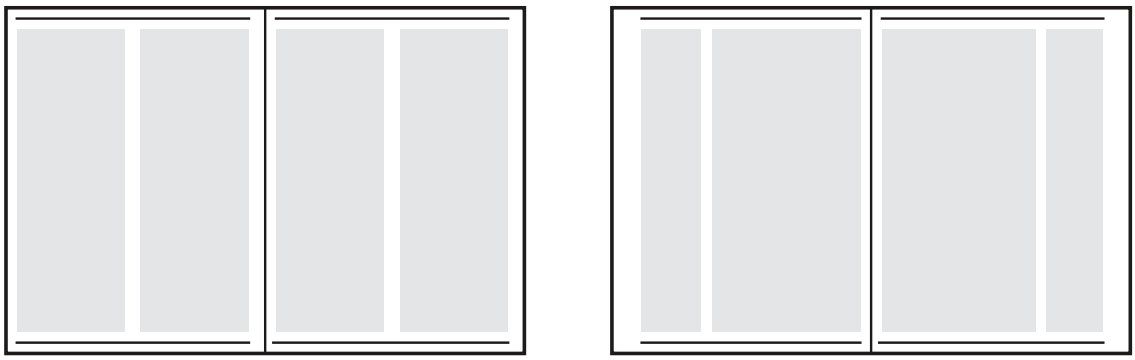


Figure 4.3: Modèles de page usuels pour les revues.

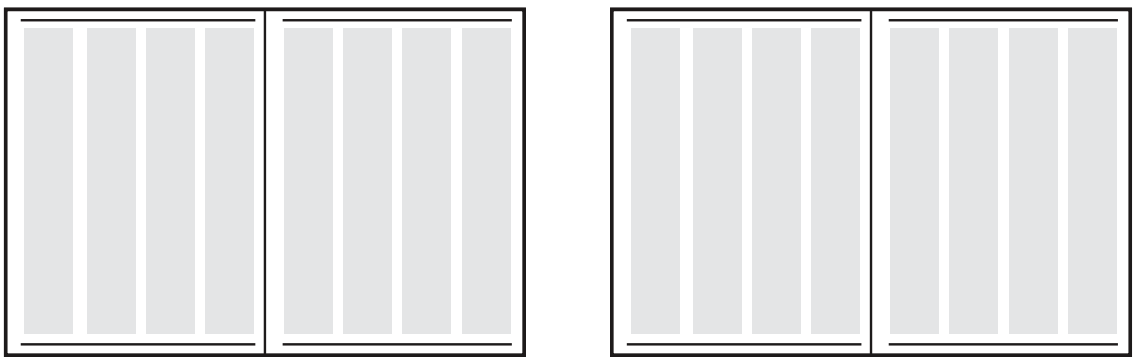


Figure 4.4: Modèles de page usuels pour les magazines.

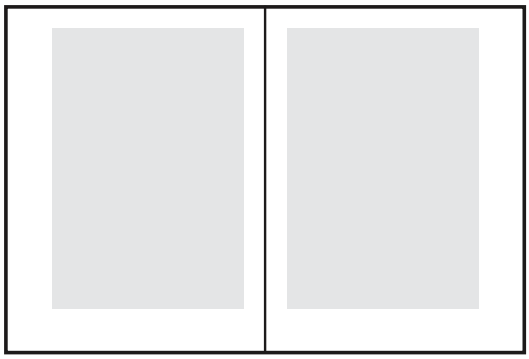


Figure 4.5: Modèles de page usuels pour les livres.

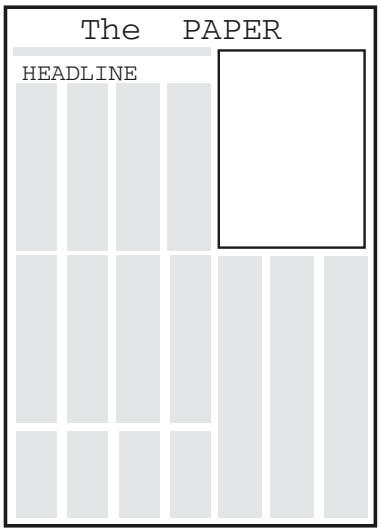


Figure 4.6: Modèle de page usuel pour la première page d'un journal.

4.3 Structure physique générique

Selon le dictionnaire Petit Robert, une structure décrit la manière dont un édifice est construit; en architecture par exemple, elle désigne l'agencement des parties d'un bâtiment. Par analogie, la structure physique spécifique à un document se doit de décrire ses constituants ainsi que leurs relations de voisinage. Une structure physique générique décrit, quant à elle, un moule qui permet non seulement de dériver différentes structures spécifiques, mais aussi de valider les structures spécifiques qui lui sont conformes.

A l'instar des modèles proposés dans la littérature, en l'occurence celui de R. Ingold dans [41] ou celui de V. Quint dans [87], nous représentons la structure physique d'un document au moyen d'une structure arborescente, naturelle pour décrire les relations de hiérarchie entre les entités constituant un document. La structure physique générique des documents composites, sous-jacente à notre analyse, est donn'ee par la figure 4.7. On y distingue trois niveaux de raffinement partant de l'entité physique *volume* qui désigne la structure physique d'un document, à l'entité physique *CConnexe* composée de *Pixels* et qui désigne les composantes connexes.

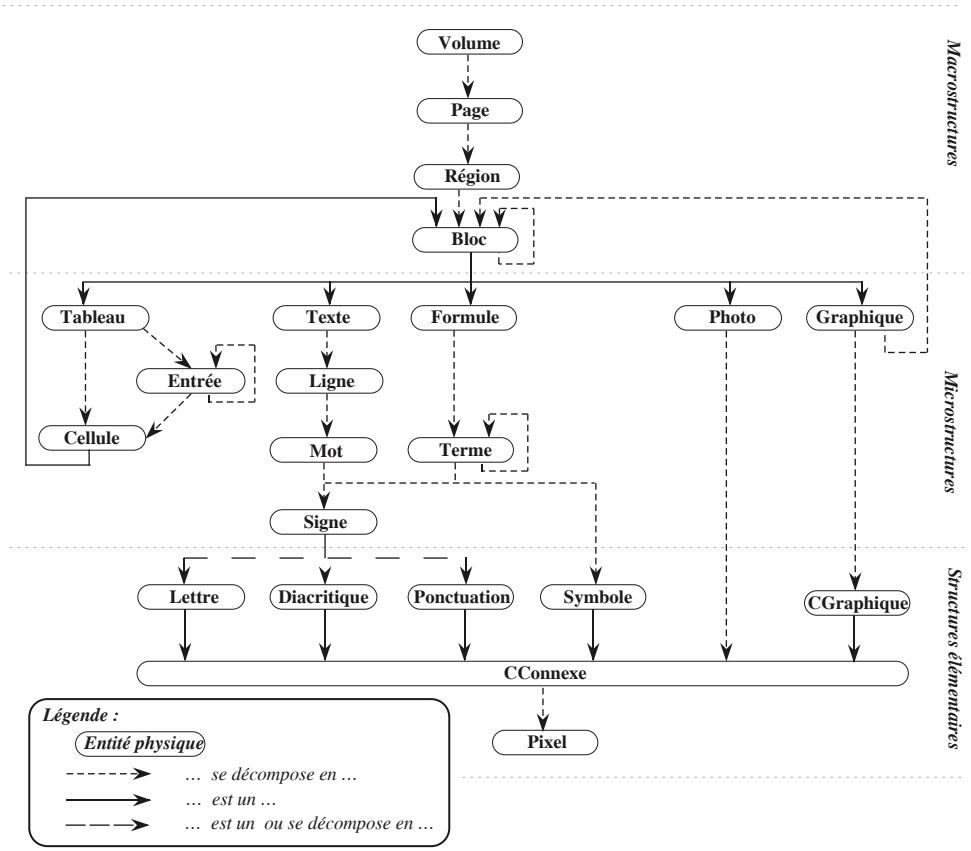


Figure 4.7: Structure physique générique des documents composites.

4.3.1 Macrostructures

Une macrostructure, comme nous le présentons dans le chapitre 2.1.5, décrit l'organisation globale d'un document en partant de l'ensemble de ses pages et en s'arrêtant au niveau de ses blocs.

Volume et Page

Un *Volume* est une entité physique qui désigne l'ensemble des pages d'un document composite; c'est, par exemple, l'ensemble des pages constituant un article scientifique, un livre ou un de ses chapitres, une référence bibliographique. Une *Page* est une entité physique correspondant à la notion usuelle de page dans un document imprimé. L'exemple de la figure 4.8 illustre la

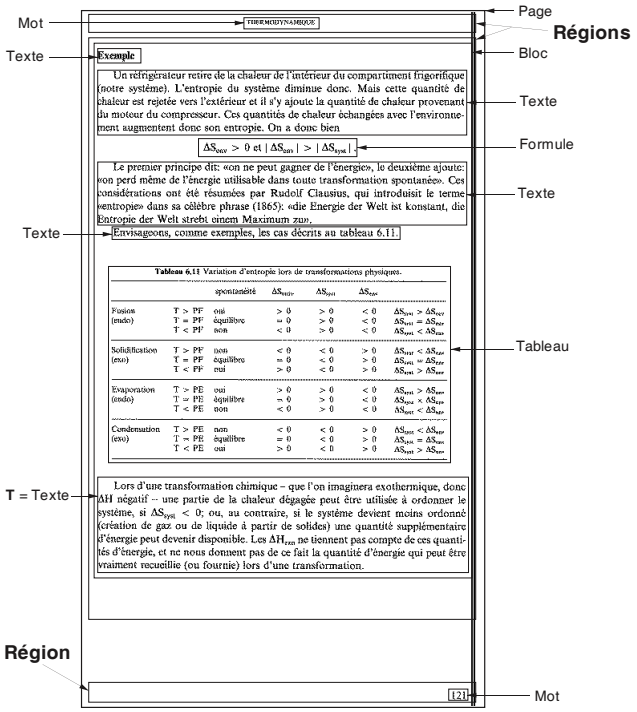


Figure 4.8: Un exemple de macrostructure de page.

macrostructure d'une page de document. Le bloc textuel désigné par T servira à illustrer la microstructure des blocs textuels (cf. figure 4.9).

Région

En observant, par exemple, les pages du présent chapitre, on note, qu'excepté la première page, toutes les autres sont subdivisées en quatre portions dédiées à des contenus différents : (1) la portion dédiée au titre courant située dans la partie supérieure gauche pour les pages impaires et dans la partie supérieure droite pour les pages paires, (2) la portion dédiée au numéro de page située, à l'inverse du titre courant, dans la partie supérieure gauche pour les pages paires et dans la partie supérieure droite pour les pages impaires, (3) la portion dédiée au contenu proprement dit et, pour finir, (4) la portion optionnelle dédiée aux notes de bas de page. Cette dernière portion est de taille variable; lorsqu'elle est présente, elle est séparée du contenu principal par un filet (cf. [2]). Cette partition des pages en régions présente l'avantage d'être invariante à travers les pages d'un même document. Dans notre modèle, une *région* est formellement définie comme suit :

> Une *Région* est une entité physique qui désigne une portion rectangulaire dédiée à une information de nature différente par rapport aux autres régions présentes dans la page; sa position relative est invariante à travers les pages d'un même document.

[2] Ceci n'a pour but, que d'illustrer la portion de page dédiée aux notes de bas de page.

Bloc

Un *Bloc* désigne une entité physique qui peut soit contenir d'autres blocs, soit désigner une microstructure (cf. section 4.3.2) pouvant être un bloc textuel, une formule, un tableau, un graphique ou encore une photographie.

4.3.2 Microstructures

La microstructure, contrairement à la macrostructure, décrit plus finement l'organisation d'un document en partant de ses blocs spécifiques pour s'arrêter au niveau des entités élémentaires (cf. section 4.3.3). Si la macrostructure décrit globalement la structure graphique des pages d'un document, la microstructure décrit plutôt la structure graphique du contenu de ce dernier.

Texte

L'entité physique *Texte* désigne un bloc textuel qui correspond soit à une entité logique de type paragraphe, soit à une partie de celle-ci. L'entité Texte est composé de *Lignes* (de texte) disposées harmonieusement les unes sous les autres et caractérisées par un interligne régulier. L'entité physique Ligne est composée de *Mots* (éventuellement suivis d'indices mathématiques) alignés dans le sens de la lecture et séparés par des espaces inter-mots harmonieusement distribués dans la ligne en fonction du mode de justification. L'entité physique Mot ne correspond pas nécessairement à la notion usuelle de mot (logique) qui après formatage peut se répartir sur deux lignes consécutives suite à une césure. Un mot est constitué de *Signes* alignés dans le sens de la lecture et séparés par des espaces inter-signes inférieurs aux espaces inter-mots. L'exemple de la figure 4.9 schématise la

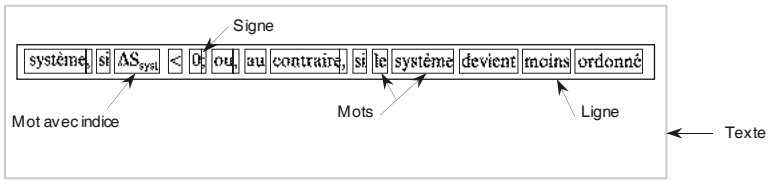


Figure 4.9: Un exemple de microstructure de texte.

microstructure de la 3ème ligne de texte du bloc textuel désigné par T dans la figure 4.8.

Formule

Nous désignons toute expression mathématique par l'entité physique *Formule* qui se compose de *Termes* mathématiques. Une entité physique Terme est composée soit d'autres Termes, soit d'entités physiques élémentaires, notamment des *Symboles* et des *Signes*. Nous supposons qu'une formule doit tenir sur une ligne.

Tableau

Nous désignons par l'entité physique *Tableau*, tout ensemble d'entités physiques dont l'organisation globale reflète une structure matricielle. Cette structure bidimensionnelle peut se décomposer en privilégiant une direction par rapport à l'autre (c.-à-d. la décomposition en colonnes ou en rangées). L'entité physique *Entrée*, désignant une dimension du tableau, correspond soit à une colonne, soit à une ligne de celui-ci. Elle se compose *Entrées* et de *Cellules* qui constituent les éléments du tableau; une *Cellule* est une microstructure.
La figure 4.10 illustre la microstructure d'un tableau dans laquelle nous avons privilégié la structure en colonnes. Ce modèle facilite la description de la structure physique d'un tableau au moyen d'une

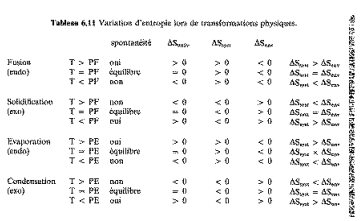


Tableau 6.11 Variation d'entropie lors de transformations physiques.

		spontanéité	ΔS_{sup}	ΔS_{syst}	ΔS_{ext}	
Fusion (endo)	T > PF	oui	> 0	> 0	< 0	$\Delta S_{syst} > \Delta S_{ext}$
	T = PF	équilibre	= 0	> 0	< 0	$\Delta S_{syst} = \Delta S_{ext}$
	T < PF	non	< 0	> 0	< 0	$\Delta S_{syst} < \Delta S_{ext}$
Solidification (exo)	T > PF	non	< 0	< 0	> 0	$\Delta S_{syst} < \Delta S_{ext}$
	T = PF	équilibre	= 0	< 0	> 0	$\Delta S_{syst} = \Delta S_{ext}$
	T < PF	oui	> 0	< 0	> 0	$\Delta S_{syst} > \Delta S_{ext}$
Evaporation (endo)	T > PE	oui	> 0	> 0	< 0	$\Delta S_{syst} > \Delta S_{ext}$
	T = PE	équilibre	= 0	> 0	< 0	$\Delta S_{syst} = \Delta S_{ext}$
	T < PE	non	< 0	> 0	< 0	$\Delta S_{syst} < \Delta S_{ext}$
Condensation (exo)	T > PE	non	< 0	< 0	> 0	$\Delta S_{syst} < \Delta S_{ext}$
	T = PE	équilibre	= 0	< 0	> 0	$\Delta S_{syst} = \Delta S_{ext}$
	T < PE	oui	> 0	< 0	> 0	$\Delta S_{syst} > \Delta S_{ext}$

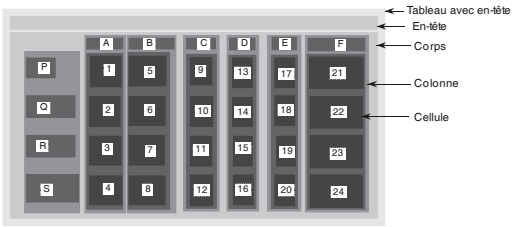


Figure 4.10: Un exemple de microstructure d'un tableau.

arborescence pure. Le lecteur intéressé par la modélisation de la structure logique des tableaux pour lesquels une structure d'arbre n'est plus suffisante, peut se rapporter aux travaux de C. Vanoirbeek dans [84].

Graphique et photographie

L'entité physique *Graphique* désigne par exemple un dessin, une illustration ou un diagramme. Elle est constituée de composantes graphiques *CGraphiques* et d'autres blocs que nous supposons avant tout textuels. Une composante graphique est une flèche, un filet, un arc, un cercle, un triangle, ou toute autre forme géométrique élémentaire.

Nous désignons par entité physique *Photo* toute photographie qui généralement se présente sous la forme d'une image trammée. Nous supposons qu'une entité de type *Photo* est directement composée de composantes connexes qui, elles, se décomposent en *Pixels*.

4.3.3 Structures élémentaires

Les structures élémentaires comprennent en gros les caractères et toutes les entités physiques que l'on ne peut décomposer en se basant sur les délimiteurs de type espace (cf. section 4.4). Les signes désignent des entités presque élémentaires dans une structure physique. Il s'agit de caractères tels que les *Lettres* simples comme "a" ou les lettres composées comme "ê" constituées d'une lettre simple ("e") et d'un signe *Diacritique* ("^"), les marques de *Ponctuation* comme ";", les *Symboles* mathématiques comme $\sum$ et les composantes graphiques *CGraphiques* comme $\triangle$. Les signes que nous supposons indépendants et séparables les uns des autres sont formés de composantes connexes *CConnexes* qui, à leur tour, se décomposent en *Pixels*.

4.3.4 Modélisation des entités physiques

En général, la structure d'arbre pure est suffisante pour modéliser la plupart des documents, en l'occurence ceux dont la segmentation peut être effectuée par une technique de découpe récursive XY : c'est le choix dans nombreux de systèmes de reconnaissance [87, 41]. Dans notre analyse, nous avons également opté pour une structure d'arbre XY que nous avons enrichie de nœuds mosaïques. En effet :

> La *structure physique* d'un document composite peut se présenter soit sous la forme d'une structure de *Manhattan* que l'on peut représenter au moyen d'un arbre XY [43, 47], soit sous la forme d'une structure *mosaïque*, structure que l'on ne peut représenter au moyen d'un arbre XY.

Dans une structure d'arbre XY pure, chaque nœud représente une entité physique pouvant se décomposer soit selon l'axe X, soit selon l'axe Y. La figure 4.11 décrit notre représentation des entités physiques. Cette représentation a l'avantage d'être uniforme quelle que soit la nature des entités décrites, par exemple, un document, une page, une région, un bloc, un tableau, un texte,

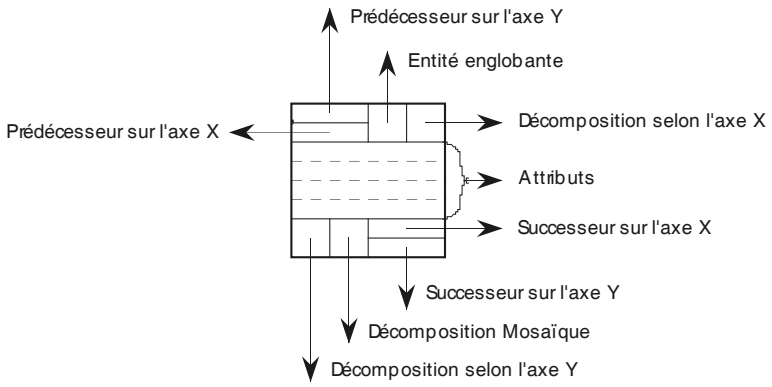


Figure 4.11: Schéma de représentation d'une entité physique.

une expression mathématique, etc. L'entité physique décrite à la figure 4.12 représente le tableau de la figure 4.10.

4.4 Délimiteurs d'entités physiques

4.4.1 Importance des espaces

Comme dans une exposition artistique, l'espace sert dans un document à mettre en évidence la structure logique et à faciliter l'accès au contenu de ce dernier. L'espace entre les objets est tout aussi important que les objets eux-mêmes; cette interdépendance est prépondérante pour une bonne perception lors de la lecture.

> *Readibilty is not wholly built in the letterforms as such. One half of it is in the spacing between the words, the lines, the columns; in the geometry of the text and the margins, i.e. in the visual editing.*
>
> *Fernand Baudin* cité par *Richard Rubinstein* dans [88]

Le *rythme typographique* [89, 88] dans un document est déterminé par cette relation d'interdépendance et par les changements de fontes. James Hartley dans [90] présente la manière dont l'espace peut être utilisé de façon systématique pour représenter la structure d'un document. A en croire Richard Rubinstein :

> *Blank space is a thing to be described, be it a margin, the leading, the space between paragraphs, or an indentation... An aesthetically pleasing page is one in which all elements, both objects and spaces, are in harmony.*
>
> Richard Rubinstein [88]

Forts de ces observations, nous avons choisi de décrire la structure physique d'un document par le biais de la description des espaces inoccupés dans le document et constituant le *fond* de ce dernier.

4.4.2 Rectangles structurants

Pour le typographe ou l'imprimeur, l'espace sert de délimiteur dans la conception des modèles de pages. Certains de ces modèles sont définis par une grille partitionnant l'espace en cadrans

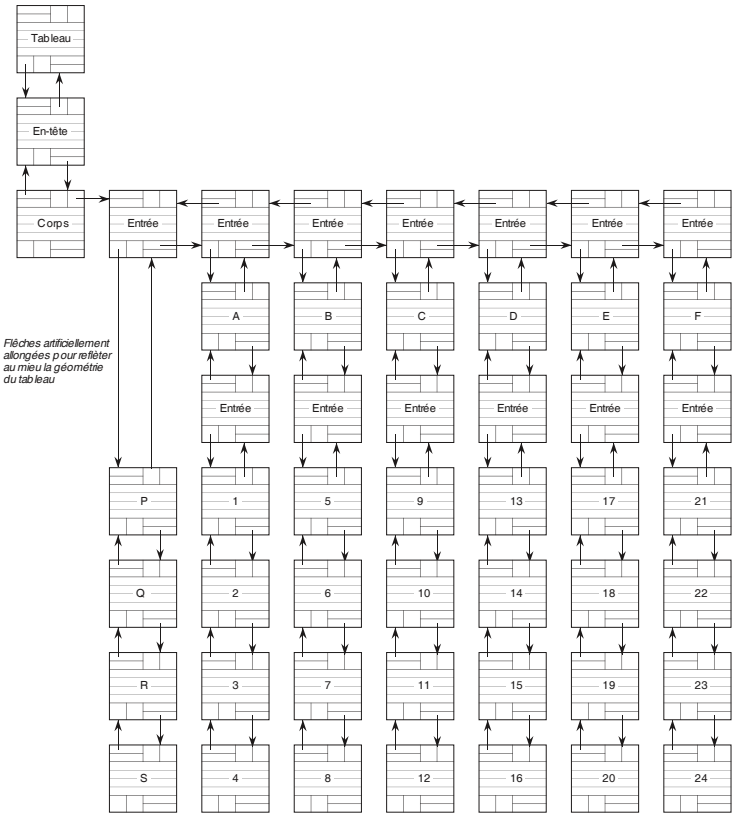


Figure 4.12: Représentation au moyen d'un arbre de la structure physique d'un tableau.

dédiés à des types de contenu bien définis. L'espace est utilisé comme délimiteur pour marquer la frontière entre les cadrans. Nous appelons de tels délimiteurs des *rectangles structurants* que nous définissons comme suit :

> Un *rectangle structurant* est une zone rectangulaire blanche maximale et délimitée de chaque côté soit par la présence d'une entité physique, soit par le bord de la page.

Nous dirons d'un rectangle structurant qu'il est horizontal (resp. vertical) s'il permet de séparer horizontalement (resp. verticalement) une entité physique en deux autres. La figure 4.13 illustre quelques exemples de rectangles structurants extraits du tableau de la figure 4.10. La structure rectangulaire de certains de ces rectangles structurants est perturbée en cas de recoupement avec d'autres rectangles structurants.

Dans notre approche, les rectangles structurants servent avant tout de primitives de structuration des espaces inoccupés dans un document, aussi bien au niveau des macrostructures que des microstructures.

4.4.3 Séparateurs

Dans un modèle de pages, les filets servent généralement, au même titre que les rectangles structurants, à délimiter les blocs. Ils contribuent ainsi à renforcer la séparation entre les différentes parties composant une page. C'est le cas dans la structure graphique des tableaux où les filets sont souvent utilisés pour renforcer la séparation entre les cellules. Dans le but de décrire les modèles de pages entièrement au moyen de ce genre de délimiteurs, nous avons introduit la notion de séparateur que nous définissons comme suit :

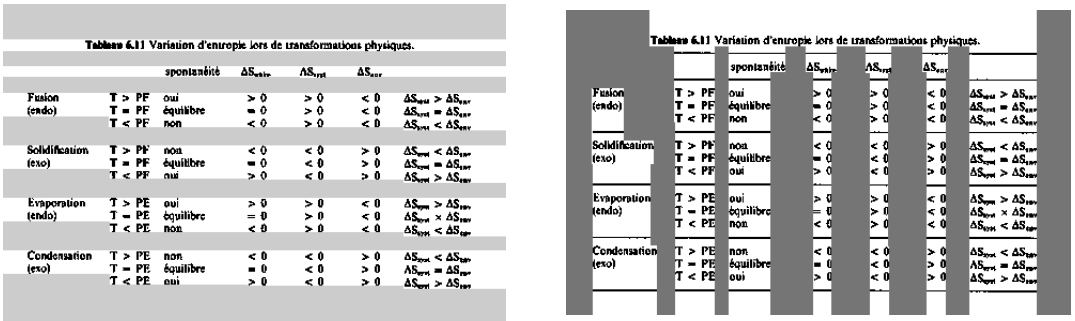


Figure 4.13: Exemples de rectangles structurants extraits d'un tableau.

> Un *séparateur horizontal* (resp. vertical) désigne un rectangle structurant horizontal (resp. vertical) ou un filet horizontal (resp. vertical) appartenant à une zone blanche maximale; il est délimité à chacune de ses extrémités gauche et droite (resp. en haut et en bas) par un séparateur vertical (resp. horizontal).

Grâce à cette définition récursive dans laquelle un séparateur est défini par rapport à deux autres, un modèle de pages est transformé en un réseau de séparateurs génériques. Comme l'indique la figure 4.14, un séparateur est caractérisé par sa position relative *pos* dans la page, par sa taille minimale *size* (ex. la hauteur minimale pour un séparateur horizontal) et par les deux séparateurs sep_1 et sep_2 qui le délimitent.

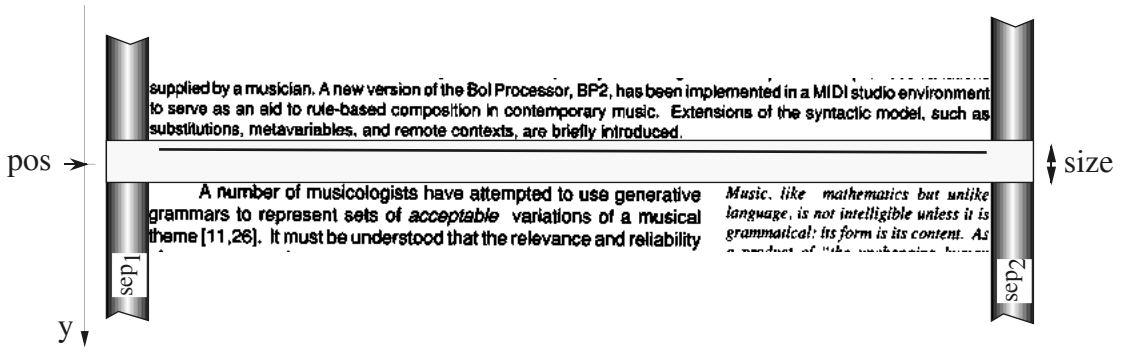


Figure 4.14: Un exemple de séparateur horizontal.

Nous distinguons, en fonction de ces attributs, trois types de séparateurs :

1. les séparateurs *statiques* désignent des séparateurs dont l'attribut de position *pos* est statique et dont les deux séparateurs sep_1 et sep_2 le délimitant sont également statiques; ils servent à délimiter des blocs dont la position est connue a priori.

2. les séparateurs *flottants* désignent des séparateurs dont l'attribut de position *pos* est variable et dont les deux séparateurs sep_1 et sep_2 le délimitant sont statiques; ils servent à délimiter des blocs dont la position est variable comme c'est le cas pour les figures.

3. les séparateurs *élastiques* désignent des séparateurs dont l'attribut de position *pos* est statique et dont l'un au moins des deux séparateurs sep_1 ou sep_2 le délimitant est flottant; ils servent, par exemple, à marquer la séparation entre deux colonnes précédées d'un objet flottant qui interrompt la continuité du séparateur.

L'intérêt de la modélisation des espaces réside dans le fait qu'elle permet de décrire de manière générique les séparateurs qui serviront de délimiteurs d'entités génériques. Les séparateurs ainsi définis, peuvent aussi servir de caractéristiques dans des appications de classification de documents.

4.5 Architecture du système de reconnaissance

Dans cette section, nous décrivons l'architecture de notre système de reconnaissance de la strcuture physique des documents composites. Le système, illustré au moyen de la figure 4.15, se compose de trois phases d'analyse correspondant chacune à un des trois niveaux de notre modèle des documents composites (cf. figure 4.7) :

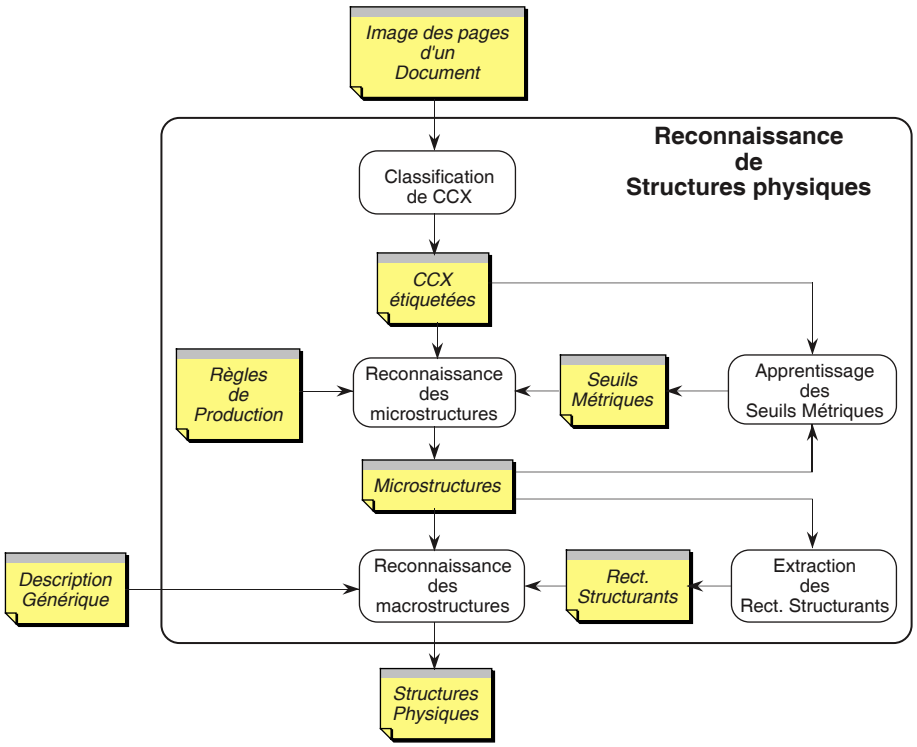


Figure 4.15: Architecture du système de reconnaissance.

Classification des composantes connexes Elle est réalisée au moyen d'une technique de classification statistique de la famille des *k-plus proches voisins*. Cette phase correspond à la reconnaissance des entités élémentaires (cf. section 6.1).

Reconnaissance des microstructures Elle est guidée par les règles régissant l'aspect graphique usuel des des microstructures. Dans le chapitre 5, nous établissons, après une étude exhaustive sur la topologie locale entre entités physiques, les règles de production qui gouvernent l'aspect graphique usuel des microstructures; il s'agit là, de la première originalité de notre travail. Dans le chapitre 6, nous décrivons une approche mixte de reconnaissance fondée sur les règles de production établies au chapitre 5. Dans le chapitre 7, nous abordons de front un problème souvent passé sous silence dans ce genre de système : celui de la sélection automatisée des seuils métriques

utilisés dans les différentes étapes de segmentation; il s'agit ici, de la deuxième originalité de notre travail.

Reconnaissance des macrostructures Elle est guidée par la description générique de la classe du document traité. Dans le chapitre 8, nous proposons un langage qui permet de décrire partiellement la macrostructure générique des documents puisqu'il n'existe, pour la reconnaissance de cette dernière, aucune méthode universelle. Le langage que nous avons défini tient son originalité du fait que les documents sont décrits par rapport aux espaces inoccupés (c.-à-d. le *fond* du document) et non par rapport aux objets eux-mêmes. Dans le chapitre 9, nous décrivons notre approche de reconnaissance des macrostructures.

Conclusion

Dans ce chapitre, nous avons présenté une modélisation de la structure graphique des documents composites qui sont caractérisés par une forte proportion de texte, mais pouvant aussi contenir des formules, des tableaux, des graphiques et des photographies. Nous avons montré, dans la section 4.4, l'importance des espaces qui servent principalement de délimiteurs dans les documents. Dans le chapitre 5, nous présentons une étude systématique de l'aspect graphique usuel des microstructures qui a donné lieu à un ensemble de règles de production régissant l'aspect graphique usuel de ces dernières.

Chapitre 5

Règles de production des microstructures physiques

Dans chaque culture, on peut rélever des conventions régissant la mise en page usuelle des microstructures. En occident, par exemple, on écrit de gauche à droite et de haut en bas. Dans ce chapitre, nous présentons une étude systématique de l'aspect graphique des microstructures en l'exploitant aussi bien les espaces que des objets imprimés. Dans la section 5.1, nous posons les fondements nécessaires à cette étude. De la section 5.2 à la section 5.6, nous décrivons les règles régissant l'aspect graphique usuel des microstructures. Ces règles ont servi à la reconnaissance des microstructures présentée dans le chapitre 6.

5.1 Fondement

Pour chaque type de microstructures étiqueté X, nous avons établi des règles régissant son aspect graphique usuel au moyen d'une grammaire étiquetée de la forme suivante :

$$X \longrightarrow s_X(X_1, \ldots, X_n); \quad m_X(X_1, \ldots, X_n)$$

où

- X : une microstructure,
- $X_1, \ldots, X_n$: entités physiques *constituants* directs de X,
- s_X : définit, sous une forme graphique, la *règle structurelle* régissant l'aspect graphique usuel de X,
- m_X : définit un *prédicat métrique* régissant la topologie des constituants X_i de X.

Exemple Pour une ligne de texte, la règle s_{Ligne} se présente en partie sous la forme suivante (cf. règles 5.2 pour sa description complète) :

$$\boxed{\text{Ligne}} \longrightarrow \boxed{\boxed{\text{Ligne } e_1}\ \boxed{\text{Composante\_Ligne } e_2}} \quad ; \quad m_{ligne}(e_1, e_2)$$

où le prédicat métrique m_{ligne} régit la topologie de voisinage des mots qui constituent la ligne.

L'objectif de la suite de cette section est de présenter les bases nécessaires à l'établissement des règles régissant l'aspect graphique des microstructures : les attributs métriques dans la section 5.1.1, les topologies locales entre entités dans la section 5.1.2, les seuils métriques régissant les espaces délimitant les entités dans la section 5.1.3 et finalement les structures terminales dans la section 5.1.4.

5.1.1 Attributs métriques

Soit e une entité physique représentée par son enveloppe rectangulaire définie comme suit :

> Une enveloppe désigne le rectangle minimum (x_1, y_1, x_2, y_2) englobant une entité physique et dont les bords sont parallèles aux axes X et Y.

Dans notre analyse, les attributs métriques qui ont servi à la classification des entités physiques sont présentés ci-après :

- $x_1(e)$: abscisse du coin haut gauche de l'enveloppe de e,
- $x_2(e)$: abscisse du coin bas droit de l'enveloppe de e,
- $y_1(e)$: ordonnée du coin haut gauche de l'enveloppe de e,
- $y_2(e)$: ordonnée du coin bas droit de l'enveloppe de e,
- $p(e)$: nombre de pixels de couleur noire dans e,
- $h(e) = x_2(e) - x_1(e)$: hauteur de l'enveloppe de e,
- $l(e) = y_2(e) - y_1(e)$: largeur de l'enveloppe de e,
- $s(e) = l(e) \times h(e)$: surface de l'enveloppe de e,
- $d(e) = p(e)/s(e)$: densité de e par rapport à son enveloppe.

Nous avons ajouté à cette liste, deux attributs auxiliaires relevant, l'un du domaine logique, et l'autre de la digitalisation :

- $etiq(e)$: étiquette logique associée à e,
- res : résolution à laquelle le document en cours de traitement a été digitalisé,

Sur la base de ces attributs, nous présentons dans la section 5.1.2 les primitives servant à l'établissement des prédicats métriques.

5.1.2 Topologie locale entre entités physiques

Inventaire de topologies locales entre entités

Dans le plan XY, nous avons mené une étude systématique des différents types de relations spatiales possibles entre les enveloppes de deux entités physiques. Nous avons dénombré au total 169 topologies possibles illustrées à la figure 5.1. En effet, lorsque l'on considère le profil des enveloppes sur chacun des deux axes, on répertorie 13 configurations différentes qui correspondent aux 13 relations spatiales unidimensionnelles de Allen définies dans son article [91] consacré à la logique temporelle. Cette classification des topologies locales possibles entre entités physiques est utilisée dans l'établissement des prédicats métriques de voisinage. Elle sert également à établir des critères d'ordre de parcours sur les entités formant les nœuds d'une structure physique (cf. section 6.5). Parmi les primitives classiques de gestion d'arbre, celles utilisées dans notre analyse sont les suivantes sont :

- $prem(e)$: première entité parmi les descendants de e,
- $dern(e)$: dernière entité parmi les descendants de e,
- $suiv(e)$: entité suivant e dans sa structure englobante,
- $prec(e)$: entité précédant e dans sa structure englobante,
- $pere(e)$: entité englobant e,
- $N(e)$: nombre de descendants directs de e.

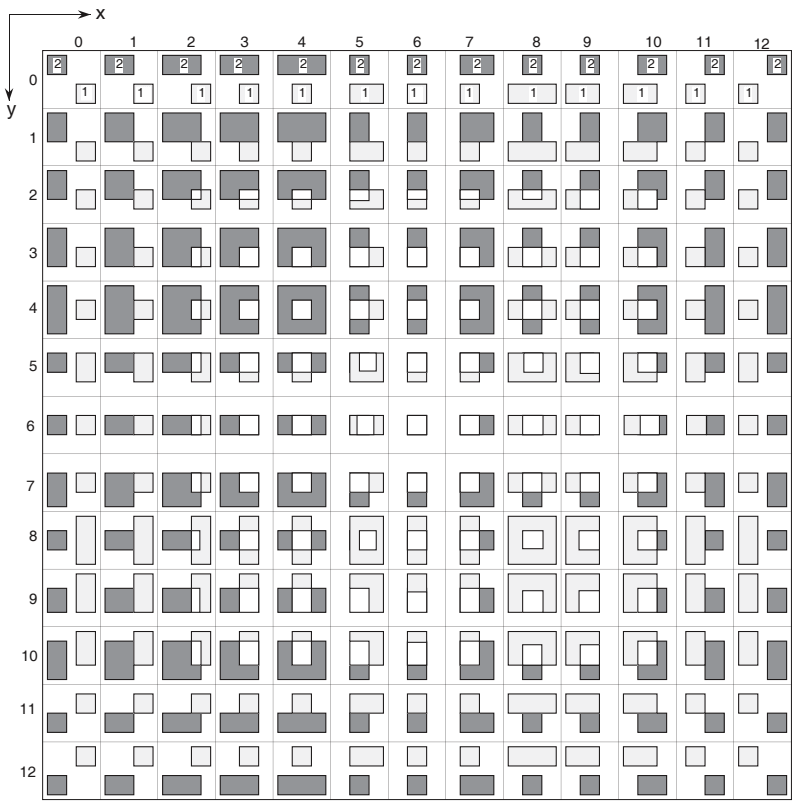


Figure 5.1: Topologies locales possibles entre entités physiques.

Mesures de topologie locale

L'homogénéité constitue un critère fondamental dans la décision soit de regrouper des entités voisines en une super entité englobante et homogène, soit de subdiviser une entité hétérogène en sous entités plus homogènes. Le calcul de l'homogénéité d'une entité est fonction d'une part de l'espace séparant ses constituants et, d'autre part, de l'alignement de ces derniers. Par exemple, un mot sera associé à une ligne de texte (1) si il repose sur la même ligne de base que celle-ci et (2) si il est distant de celle-ci d'une distance horizontale comprise entre l'espace inter-mots maximal et l'espace inter-caractères maximal.

Désignons par profil horizontal (resp. vertical) d'une entité, la projection de son enveloppe sur l'axe Y (resp. X); soient e_1 et e_2 deux entités physiques : nous énumérons ci-dessous les primitives servant à la mesure de la topologie locale entre e_1 et e_2 selon l'axe Y. Ces définitions se rapportent pour les deux premières à la figure 5.1 et pour les suivantes à la figure 5.2.

- $C_y(e_1, e_2)$: retourne par rapport à l'axe Y le type de topologie locale entre e_1 et e_2,
- type de topologie locale est une valeur entière comprise entre 0 et 12,
- $P_y(e_1, e_2)$: compare le profil vertical de e_1 avec celui de e_2,

$$P_y(e_1,e_2) = \begin{cases} 0 & si \quad C_y(e_1e_2) \leq 1 & \text{on dira } e_1 < e_2 \text{ sur l'axe Y} \\ 1 & si \quad 1 < C_y(e_1e_2) < 12 & \text{on dira } e_1 = e_2 \text{ sur l'axe Y} \\ 2 & si \quad C_y(e_1e_2) \geq 12 & \text{on dira } e_1 > e_2 \text{ sur l'axe Y} \end{cases}$$

- $I_y(e_1, e_2) = min(y_2(e_1), y_2(e_2)) - max(y_1(e_1), y_1(e_2))$: intersection des profils verticaux,
- $U_y(e_1, e_2) = max(y_2(e_1), y_2(e_2)) - min(y_1(e_1), y_1(e_2))$: union des profils verticaux,
- $D_y(e_1, e_2) = min(\mid y_1(e_1) - y_1(e_2) \mid, \mid y_2(e_1) - y_2(e_2) \mid)$: décalage le plus faible entre les deux profils,
- $D_Y(e_1, e_2) = max(\mid y_1(e_1) - y_1(e_2) \mid, \mid y_2(e_1) - y_2(e_2) \mid)$: décalage le plus fort entre les deux profils,
- $d_y(e_1, e_2) = max(y_1(e_2) - y_2(e_1), y_1(e_1) - y_2(e_2))$: distance verticale séparant e_1 de e_2.

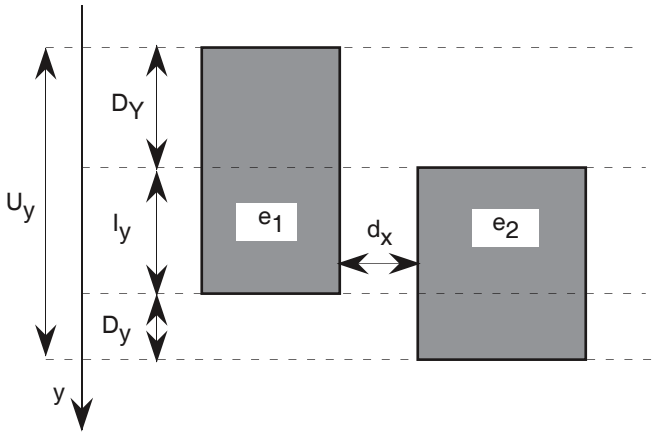


Figure 5.2: Schéma d'alignement horizontal.

Remarques :

- $I_y(e_1, e_2)$ peut être négatif, cela signifie que les enveloppes de e_1 et de e_2 sont disjointes,
- $d_y(e_1, e_2) = -I_y(e_1, e_2)$,
- $U_y(e_1, e_2) = I_y(e_1, e_2) + D_y(e_1, e_2) + D_Y(e_1, e_2)$,
- On définit de façon analogue C_x, P_x, I_x, U_x, D_x, D_X et d_x.

5.1.3 Seuils métriques

Le but de cette section est d'établir avant tout une classification hiérarchique des espaces délimitant les différentes catégories d'entités composant un bloc textuel. Chaque type d'espaces est donné au moyen d'un intervalle d'entiers défini comme suit :

- $[0, d_y^d]$: d_y^d est le seuil maximal des espaces verticaux entre signes *diacritiques* et leur corps,
- $[0, d_x^c]$: d_x^c est le seuil maximal des espaces horizontaux entre *caractères* d'un même mot,
- $]d_x^c, d_x^m]$: d_x^m est le seuil maximal des espaces horizontaux entre *mots* d'une même ligne,
- $[0, d_y^l]$: d_y^l est le seuil maximal des espace verticaux entre *lignes* d'un même bloc textuel,
- d_y^b : l'*interligne*, espace vertical séparant deux lignes de *base* consécutives,
- d_x^i : l'*identation*, retrait de la première ligne d'un bloc textuel par rapport à la seconde.

A ces seuils contribuant à l'établissement des prédicats métriques de voisinage qui régisssent l'aspect graphique des microstructures, nous avons ajouté :

- h_f : la hauteur la plus fréquente des composantes connexes,
- l_f : la largeur la plus fréquente des composantes connexes,

Intuitivement, h_f est une estimation du paramètre typographique *x-height* qui désigne la hauteur de la lettre miniscule "x" dans une fonte donnée. Dans les méthodes que nous préconisons, tous ces paramètres, talon d'Achille des techniques classiques de segmentation, ont fait l'objet d'une estimation par apprentissage automatique au cours de l'analyse des documents. Par ailleurs, ces estimations peuvent également provenir de connaissances acquises lors des traitements antérieurs. Dans cette partie, nous supposons connus tous ces paramètres; le chapitre 7 est consacré à leur estimation.

5.1.4 Structures terminales

Dans la suite de ce chapitre, nous présentons la grammaire des règles traduisant l'aspect graphique usuel des microstructures. Dans cette grammaire dont la partie structurelle est présentée sous une forme graphique, les structures terminales (symboles terminaux), énumérées ci-après, commencent par une lettre miniscule :

lettre_a	ponctuation	symbole	graphique	barre_h	filet_v
lettre_b	tiret_h	symbole_v	graphique_h	barre_v	
opérateur	tiret_v	symbole_h	graphique_v	filet_h	

Dans la nomenclature des symboles terminaux : _h caractérise les entités allongées horizontalement, _v celles allongées verticalement, _a les lettres de même hauteur que la lettre "x" et pour finir _b caractérise les lettres de hauteur différente que la lettre "x". Ces structures terminales résultent d'une classification des composantes connexes réalisée au moyen d'une technique d'agglomération itérative dans l'espace des attributs métriques (cf. section 6.1).

5.2 Structure graphique usuelle des blocs textuels

5.2.1 Texte

Un bloc textuel représente une partie ou l'entièreté d'un paragraphe. Il est caractérisé, en partie, par des lignes de base parallèles entre elles et perpendiculaires aux bords gauche et droit. L'aspect graphique d'un bloc textuel est traditionnellement constitué d'une séquence de lignes de texte alignées les unes sous les autres et séparées d'une distance verticale inférieure au seuil d_y^l. Cette mise en page traditionnelle est, de plus, influencée par le mode de justification des paragraphes (GAUCHE, DROIT, JUSTIFIE ou CENTRE). Les règles de la figure 5.1, paramétrées par le mode de justification, traduisent l'aspect graphique usuelle des blocs textuels.

Texte(mode) $\longrightarrow$ [Comp_Texte e_1 / Comp_Texte e_2] ; $m_{texte}(e_1, e_2) \wedge mode_0(e_1, e_2)$

Texte(mode) $\longrightarrow$ [Texte(mode) e_1 / Comp_Texte e_2] ; $m_{texte}(e_1, e_2) \wedge mode_1(e_1, e_2)$

Comp_Texte $\longrightarrow$ Ligne | Mot

Règles 5.1: Règles descrivant l'aspect graphique usuel des blocs textuels.

Dans les règles 5.1, le prédicat métrique régissant la topologie locale entre e_1 et e_2 est donné par la conjonction d'une part du prédicat m_{texte}, et d'autre part du prédicat $mode_0$ ou $mode_1$ qui permettent de tenir compte du mode de justification en vigueur dans le bloc textuel. La définition de chacune de ces trois prédicats est données ci-après :

$$m_{texte}(e_1,e_2) = (d_y(e_1,e_2) \leq d_y^l) \quad \wedge \quad (I_x(e_1,e_2) > 0) \wedge (y_2(e_1) \leq y_1(e_2))$$

$$mode_0(e_1,e_2) = \begin{cases} (x_1(e_1) - x_1(e_2) \geq d_x^i) & \text{si mode = GAUCHE} \\ (x_2(e_1) - x_2(e_2) == 0) & \text{si mode = DROIT} \\ (x_1(e_1) - x_1(e_2) \geq d_x^i) & \\ \quad \wedge \quad (x_2(e_1) - x_2(e_2) == 0) & \\ \quad \wedge \quad m(e_2, suiv(e_2)) & \text{si mode = JUSTIFIE} \\ \frac{x_1(e_1)+x_2(e_1)}{2} == \frac{x_1(e_2)+x_2(e_2)}{2} & \text{si mode = CENTRE} \end{cases}$$

$$mode_1(e_1,e_2) = \begin{cases} (x_1(e_1) - x_1(e_2) == 0) & \text{si mode = GAUCHE} \\ (x_2(e_1) - x_2(e_2) == 0) & \text{si mode = DROIT} \\ (x_1(e_1) - x_1(e_2) == 0) & \\ \quad \wedge \quad (x_2(e_1) - x_2(e_2) == 0) & \\ \quad \wedge \quad m(e_2, suiv(e_2)) & \text{si mode = JUSTIFIE} \\ \frac{x_1(e_1)+x_2(e_1)}{2} == \frac{x_1(e_2)+x_2(e_2)}{2} & \text{si mode = CENTRE} \end{cases}$$

5.2.2 Ligne

L'aspect graphique usuel d'une ligne de texte est constitué d'une séquence de mots, de marques de ponctuation ou éventuellement de symboles particuliers (ex. "$\longrightarrow$") séparés les uns des autres par une distance horizontale comprise entre les seuils d_x^c et d_x^m . Les constituants d'une ligne sont caractérisés par le fait (1) qu'ils reposent sur une même ligne de base et (2) qu'ils ont une hauteur globalement harmonieuse (pas forcément identique) dans toute la ligne. L'aspect graphique usuel d'une ligne de texte est décrit par les règles 5.2.

Ligne $\longrightarrow$ Comp_Ligne e_1 Comp_Ligne e_2 ; $m_{ligne}(e_1,e_2)$

Ligne $\longrightarrow$ Ligne e_1 Comp_Ligne e_2 ; $m_{ligne}(e_1,e_2)$

Comp_Ligne $\longrightarrow$ Mot | Signe | Symbole

Règles 5.2: Règles descrivant l'aspect graphique usuel des lignes de texte.

Dans les règles 5.2, le prédicat métrique m_{ligne} régissant la topologie locale entre e_1 et e_2 est donné par la conjonction logique définie ci-après, dans laquelle $h_{min} = \min(h(e_1), h(e_2))$:

$$\begin{aligned} m_{ligne}(e_1,e_2) = \; & (d_x^c < d_x(e_1,e_2) \leq c \times d_x^m) \\ & \wedge \quad (I_y(e_1,e_2) > 0) \\ & \wedge \quad ((D_Y(e_1,e_2) \leq I_y(e_1,e_2)) \\ & \quad \vee \quad ((h_{min} < h_f) \quad \wedge \quad (h_{min} < 2 \times I_y(e_1,e_2)))) \end{aligned}$$

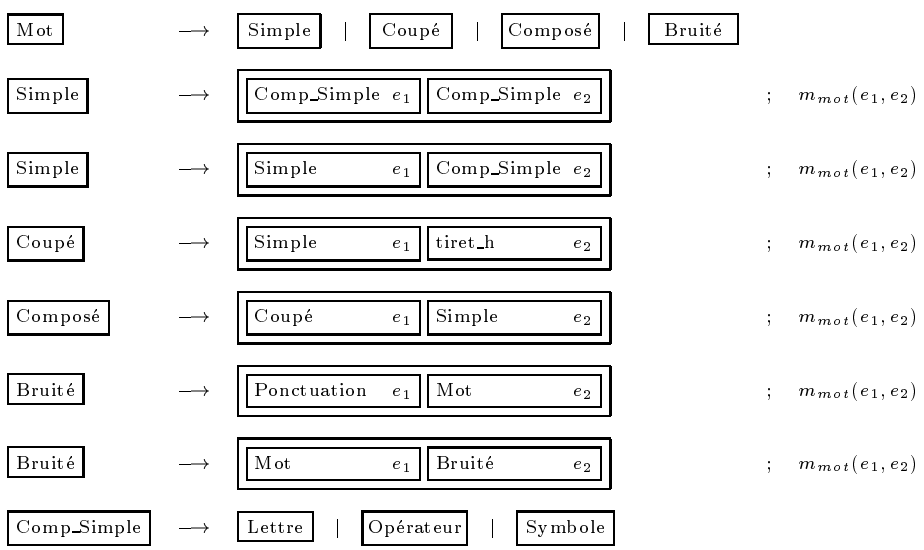
Intuitivement, on dira de deux mots consécutifs qu'ils sont alignés lorsque l'intersection de leur profil horizontal respectif est plus importante que le plus grand décalage présenté par ces deux profils. En général, l'espace après une marque de ponctuation est légèrement plus grand que l'espace

entre deux mots consécutifs; cet espace varie en fonction des cultures typographiques. Cette particularité est prise en compte, au moyen du facteur multiplicatif c, lorsque l'entité physique e_1 est une marque de ponctuation. L'estimation empirique du facteur c est établie, quelque soit le document, à :

$$c = \begin{cases} 6/5 & \text{si } e_1 \text{ est une marque de ponctuation} \\ 1 & \textit{sinon.} \end{cases}$$

5.2.3 Mot

L'aspect graphique usuel d'un mot est constitué par une séquence de lettres et de symboles séparés les uns des autres par une distance horizontale inférieure au seuil d_x^c. Les composantes d'un mot, comme pour une ligne, sont caractérisées par le fait (1) qu'elles reposent sur une même ligne de base (celle de la ligne) et (2) qu'elles ont une hauteur globalement harmonieuse. Dans la pratique, on distingue, en plus des mots classiques, des mots composés et des mots coupés en fin de ligne. L'effet de digitalisation peut entacher certains mots de bruits qui se manisfestent soit par des coupures dans un même caractère, soit par des fusions de caractères, soit par la présence de pixels isolés. L'aspect graphique usuel d'un mot est décrit par les règles 5.3 qui prennent en compte ces différents types de mots.



Règles 5.3: Règles descrivant l'aspect graphique usuel des mots.

Dans les règles 5.3, le prédicat métrique m_{mot} entre e_1 et e_2 est donné par la conjonction définie ci-après, dans laquelle $h_{min} = \min(h(e_1), h(e_2))$:

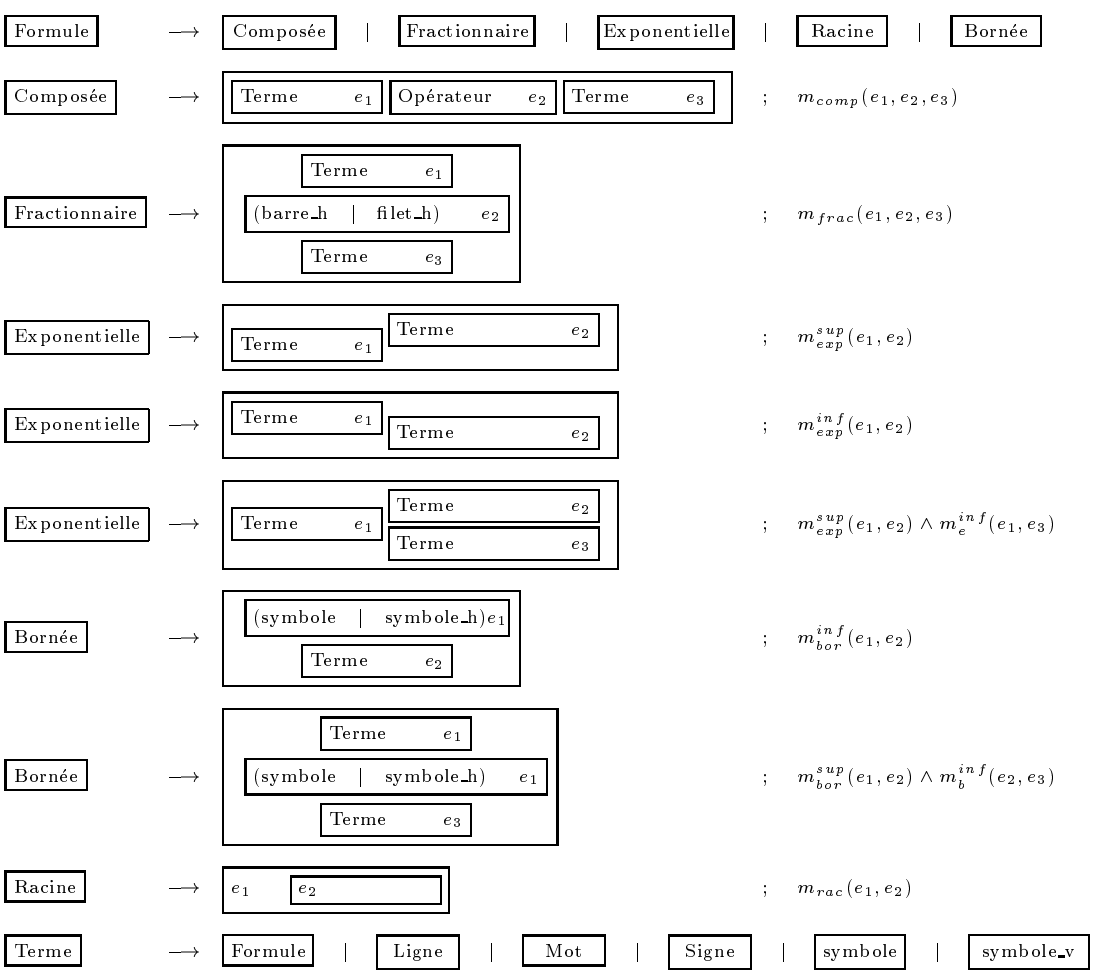
$$\begin{aligned} m_{mot}(e_1, e_2) \quad = \quad & (d_x(e_1, e_2) \leq d_x^c) \\ & \wedge \quad (I_y(e_1, e_2) > 0) \\ & \wedge \quad (x_2(e_1) < x_1(e_2)) \\ & \wedge \quad ((D_Y(e_1, e_2) \leq I_y(e_1, e_2)) \\ & \qquad \vee \quad ((h_{min} < h_f) \quad \wedge \quad (h_{min} < 2 \times I_y(e_1, e_2)))) \end{aligned}$$

Intuitivement, on dira de deux signes consécutifs qu'ils sont alignés lorsque l'intersection de leur profil horizontal respectif est plus importante que le plus grand décalage présenté par ces deux profils.

5.3 Structure graphique usuelle des formules

L'aspect graphique des expressions mathématiques représentés dans notre modèle (voir figure 4.7) par l'entité générique *Formule*, est traditionnellement constituée de termes, d'opérateurs et de symboles dont la disposition graphique est essentiellement dirigée par la sémantique de l'expression représentée. Comparées aux lignes de texte, les expressions mathématiques sont caractérisées par un aspect graphique plus aéré. En effet dans une expression mathématique, on observe que les espaces séparant les opérandes des opérateurs sont en général légèrement plus grands que le seuil d_x^m des espaces séparant les mots à l'intérieur d'un bloc textuel. L'aspect graphique des expressions mathématiques varient des plus simples aux plus complexes. Nous avons limité notre analyse aux structures graphiques les plus courantes (cf. règles 5.4).

- les expressions composées (ex. $e_1 + e_2$),
- les expressions fractionnaires (ex. $\frac{e_1}{e_2}$),
- les expressions exponentielles (ex. ${e_1}^{e_2}$ ou $e_{1_{e_2}}$),
- les expressions bornées (ex. $\sum_{e_1}^{e_2}$),
- les expressions racine (ex. $\sqrt{e_1}$).



Opérateur $\longrightarrow$ symbole_h | SigneOpérateur

Règles 5.4: Règles descrivant la structure graphique usuelle des formules.

Dans les règles 5.4, le prédicat métrique associé à chaque type d'expressions mathématiques est donné par une conjonction logique définie comme ci-après :

Pour les expressions composées :

$$m_{comp}(e_1,e_2,e_3) \quad = \quad m_a(e_1,e_2,e_3) \quad \wedge \quad m_b(e_1,e_2) \quad \wedge \quad m_b(e_2,e_3) \quad \wedge \quad m_b(e_1,e_3)$$

où

$$\begin{aligned} m_a(e_1,e_2,e_3) &= (d_x(e_1,e_2) \leq c \times d_x^m) \quad \wedge \quad (d_x(e_2,e_3) \leq c \times d_x^m) \\ m_b(e_i,e_j) &= (I_y(e_i,e_j) > 0) \\ &\quad \wedge \quad (x_2(e_i) < x_1(e_j)) \\ &\quad \wedge \quad ((D_Y(e_i,e_j) \leq I_y(e_i,e_j)) \\ &\quad\quad \vee \quad (\min(h(e_i),h(e_j) \approx I_y(e_i,e_j)))) \end{aligned}$$

Intuitivement, on dira d'une séquence de trois entités u, v et w alignées horizontalement qu'elles forment une expression composée lorsque (1) v est un opérateur et (2) u et w sont deux termes situés à équidistance de v. En se basant sur l'observation que les expressions mathématiques sont plus aérées que les blocs textuels, nous avons limité la distance séparant un terme de son opérateur à $c \times d_x^m$ où c est empiriquement estimé, quelque soit le document, à :

$$c = 3/2.$$

Pour les expressions fractionnaires :

$$m_{frac}(e_1,e_2,e_3) \quad = \quad m_a(e_1,e_2,e_3) \quad \wedge \quad m_{num}(e_1,e_2) \quad \wedge \quad m_{den}(e_2,e_3)$$

où

$$\begin{aligned} m_a(e_1,e_2,e_3) &= (d_y(e_1,e_2) \leq c \times d_y^d) \quad \wedge \quad (d_y(e_2,e_3) \leq c \times d_y^d) \\ m_{num}(e_1,e_2) &= (C_x(e_1,e_2) == 4) \quad \wedge \quad (D_X(e_1,e_2) == D_x(e_1,e_2)) \quad \wedge \quad (y_2(e_1) < y_1(e_2)) \\ m_{den}(e_2,e_3) &= (C_x(e_2,e_3) == 8) \quad \wedge \quad (D_X(e_2,e_3) == D_x(e_2,e_3)) \quad \wedge \quad (y_2(e_2) < y_1(e_3)) \end{aligned}$$

Intuitivement, on dira d'une séquence de trois entités u, v et w alignées verticalement qu'elles forment une expression fractionnaire lorsque (1) v est une barre de fraction et (2) u et w sont deux termes centrés par rapport à v et situés à équidistance de v, mais limités à $c \times d_y^d$ où c est empiriquement estimé, quelque soit le document, à :

$$c = 3.$$

Pour les expressions exponentielles :

$$\begin{aligned} m_{exp}^{sup}(e_1,e_2) &= (d_x(e_1,e_2) \leq d_x^c) \ \wedge \ (x_2(e_1) < x_1(e_2)) \ \wedge \ (1 \leq C_y(e_1,e_2) \leq 2) \ \wedge \ (h(e_1) \geq 3 \times I_y(e_1,e_2)) \\ m_{exp}^{inf}(e_1,e_2) &= (d_x(e_1,e_2) \leq d_x^c) \ \wedge \ (x_2(e_1) < x_1(e_2)) \ \wedge \ (10 \leq C_y(e_1,e_2) \leq 11) \ \wedge \ (h(e_1) \geq 3 \times I_y(e_1,e_2)) \end{aligned}$$

Intuitivement, on dira de deux entités u et v alignées horizontalement qu'elles forment une expression exponentielle lorsque l'intersection de leur profil horizontal est située dans le tiers supérieur ou dans le tiers inférieur du profil de u.

Pour les expressions bornées :

$$\begin{aligned} m_{bor}^{sup}(e_1,e_2) &= (d_y(e_1,e_2) \leq c \times d_y^d) \\ &\quad \wedge \quad (y_2(e_1) < y_1(e_2)) \\ &\quad \wedge \quad ((C_x(e_1,e_2) == 4) \quad \vee \quad (C_x(e_1,e_2) == 8)) \\ m_{bor}^{inf}(e_1,e_2) &= m_{bor}^{sup}(e_1,e_2) \end{aligned}$$

Intuitivement, on dira d'une séquence de trois entités u, v et w alignées verticalement qu'elles forment une expression bornée lorsque (1) v est un symbole et (2) u et w sont deux termes centés par rapport à v et situés à équidistance de v, mais limités à $c \times d_y^d$ où c est empiriquement estimé, quelque soit le document, à :

$$c = 2.$$

Pour les expressions racines :

$$m_{rac}(e_1,e_2) = (C_x(e_1,e_2) == 8) \wedge (C_y(e_1,e_2) == 8)$$

Intuitivement, on dira d'un symbole u contenant un terme v, qu'il est une expression racine.

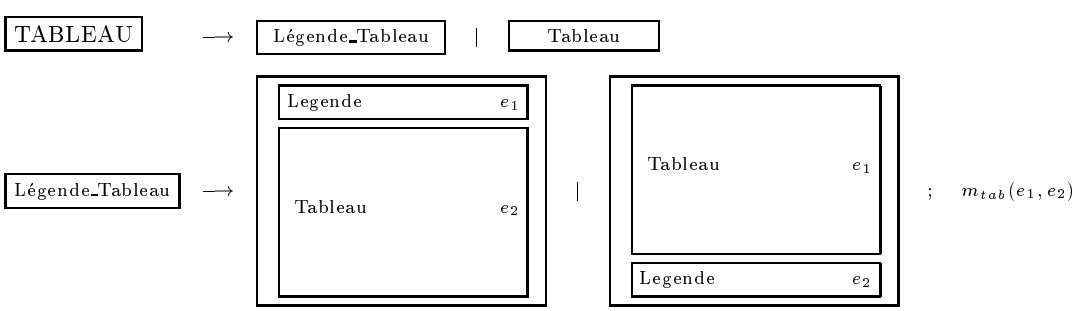
L'étude de l'aspect graphique des vecteurs ainsi que celui des matrices n'a pas été abordée dans cette thèse. Toutefois, l'aspect graphique de certaines matrices peut être interprétée comme un tableau dont l'aspect graphique reflète celui d'une matrice.

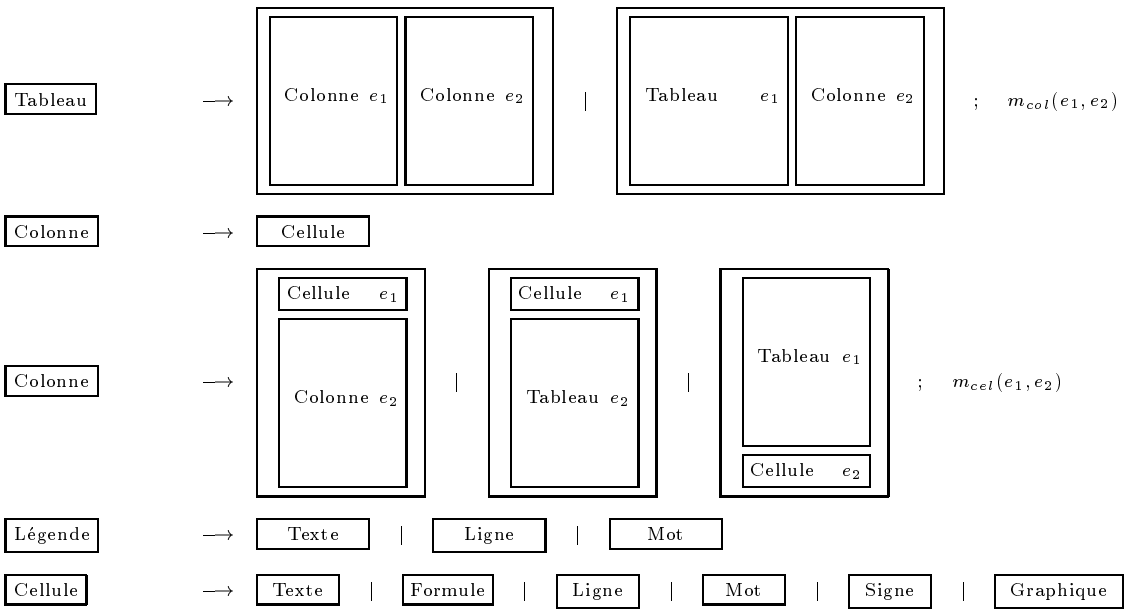
5.4 Structure graphique usuelle des tableaux

Les tableaux sont les seules entités pour lesquelles l'ordre de lecture est influencé par le point de vue du lecteur. Nous désignons par *dimension* d'un tableau, le nombre de points de vue (ordres de lecture) différents que l'on peut avoir en observant l'aspect graphique du tableau qui traditionnellement a un aspect matriciel et peut être lu :

- soit de haut en bas en lisant une rangée après l'autre,
- soit de gauche à droite en lisant une colonne après l'autre,
- soit par accès direct aux cellules du tableau.

Les règles 5.5 traduisent l'aspect graphique traditionnelle des tableaux pour lesquels nous avons privilégié la découpe en colonnes. Le choix contraire aurait pour conséquence de confondre avec un tableau toute séquence de blocs textuels alignés horizontalement.





Règles 5.5: Règles descrivant la structure graphique usuelle des tableaux.

Dans les règles 5.5, les prédicats métriques régissant les composantes d'un tableau sont donnés par une conjonction logique définie comme ci-après :

$$m_{tab}(e_1, e_2) = (d_y(e_1, e_2) \leq d_y^l) \wedge (y_2(e_1) < y_1(e_2)) \wedge (D_X(e_1, e_2) == D_x(e_1, e_2))$$

$$m_{col}(e_1, e_2) = (d_x(e_1, e_2) > d_x^m) \wedge (x_2(e_1) < x_1(e_2))$$

$$m_{cel}(e_1, e_2) = (d_y(e_1, e_2) > d_y^l) \wedge (y_2(e_1) < y_1(e_2)) \wedge (I_x(e_1, e_2) > 0))$$

Généralement dans l'aspect graphique des tableaux, la légende est centrée par rapport aux tableaux, les colonnes de ces derniers sont séparées par un espace vertical, supérieur au seuil des espaces inter-mots d_x^m, et les cellules d'une colonne sont séparées par un espace horizontal, supérieur au seuil des espaces inter-lignes d_y^l.

Comme le lecteur l'aurait remarqué, nous avons totalement fait abstraction des filets pourtant très fréquents dans l'aspect graphique des tableaux. Notre choix est justifié par l'observation suivante :

> Plus que les filets, c'est la régularité de la distribution des espaces entre cellules qui confère aux tableaux leur structure matricielle; les filets servent avant tout à renforcer cette structure matricielle.

Ainsi, comme pour n'importe quelle autre microstructure, l'analyse des tableaux est entièrement fondée sur la recherche d'une régularité dans la distribution des espaces. Cette manière de décrire les tableaux constitue un avantage certain pour notre approche, en comparaison avec les techniques présentées dans la littérature [66, 67, 68] qui toutes présupposent la présence d'un nombre minimum de filets dans l'aspect graphique des tableaux.

5.5 Structure graphique usuelle des illustrations

Dans un document composite, un graphique désigne avant tout une illustration. D'une manière générale, il n'y a aucune (ou que très peu de) systématique dans l'aspect graphique des illus-

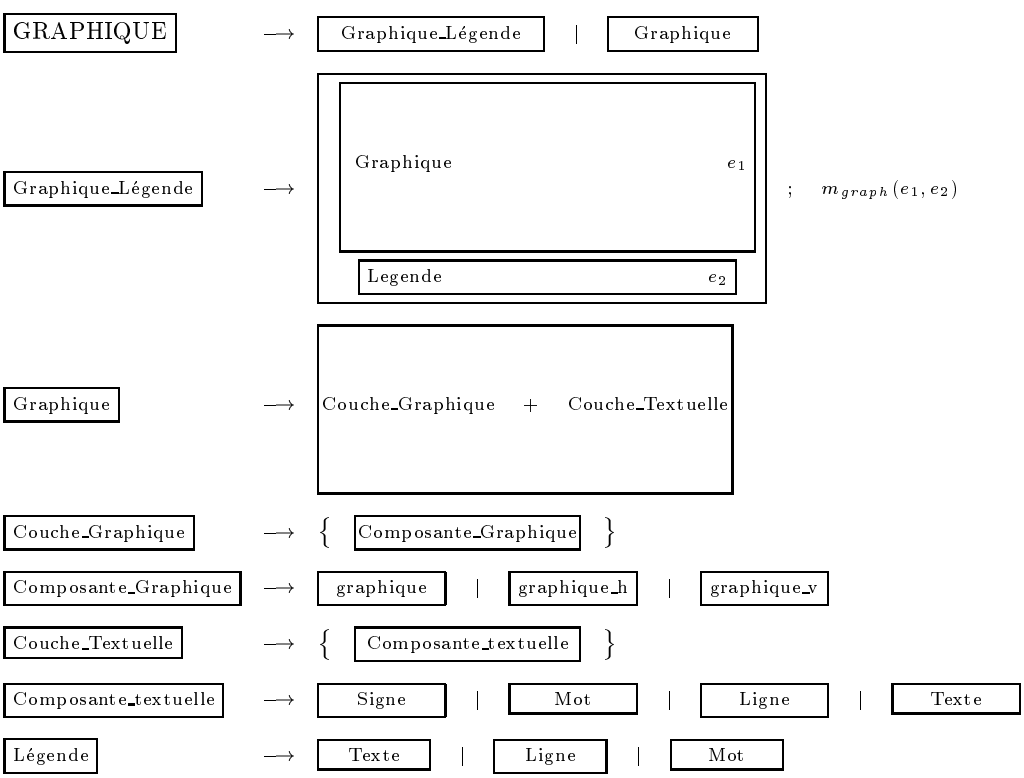
tractions observées dans les documents composites. Elles sont grossièrement composées de deux couches :

1. une couche purement graphique constituée de primitives graphiques (ex. droite, cercle, etc.),

2. une autre couche textuelle constituée de mots, de nombres et de cotations dont les positions par rapport à la couche graphique indiquent à quelle partie du dessin se rapportent les cotes et légendes.

Partant de cette observation, nous nous sommes fixés pour objectif d'extraire des entités graphiques les informations textuelles qu'elles comportent.

La structure graphique d'une photographie est quelque chose de très complexe à modéliser et dépasse le cadre de cette thèse. Par conséquent, nous nous sommes limités à son étiquetage sans nous préoccuper de sa structure interne que nous supposons directement constituée de pixels. Les règles 5.6 sont celles utilisées pour guider l'étiquetage des structures graphiques.



Règles 5.6: Etiquetage des structures graphiques et photographiques.

Dans les règles 5.6, le prédicat métrique régissant les graphiques y compris leurs légendes est donné par une conjonction logique définie comme ci-après :

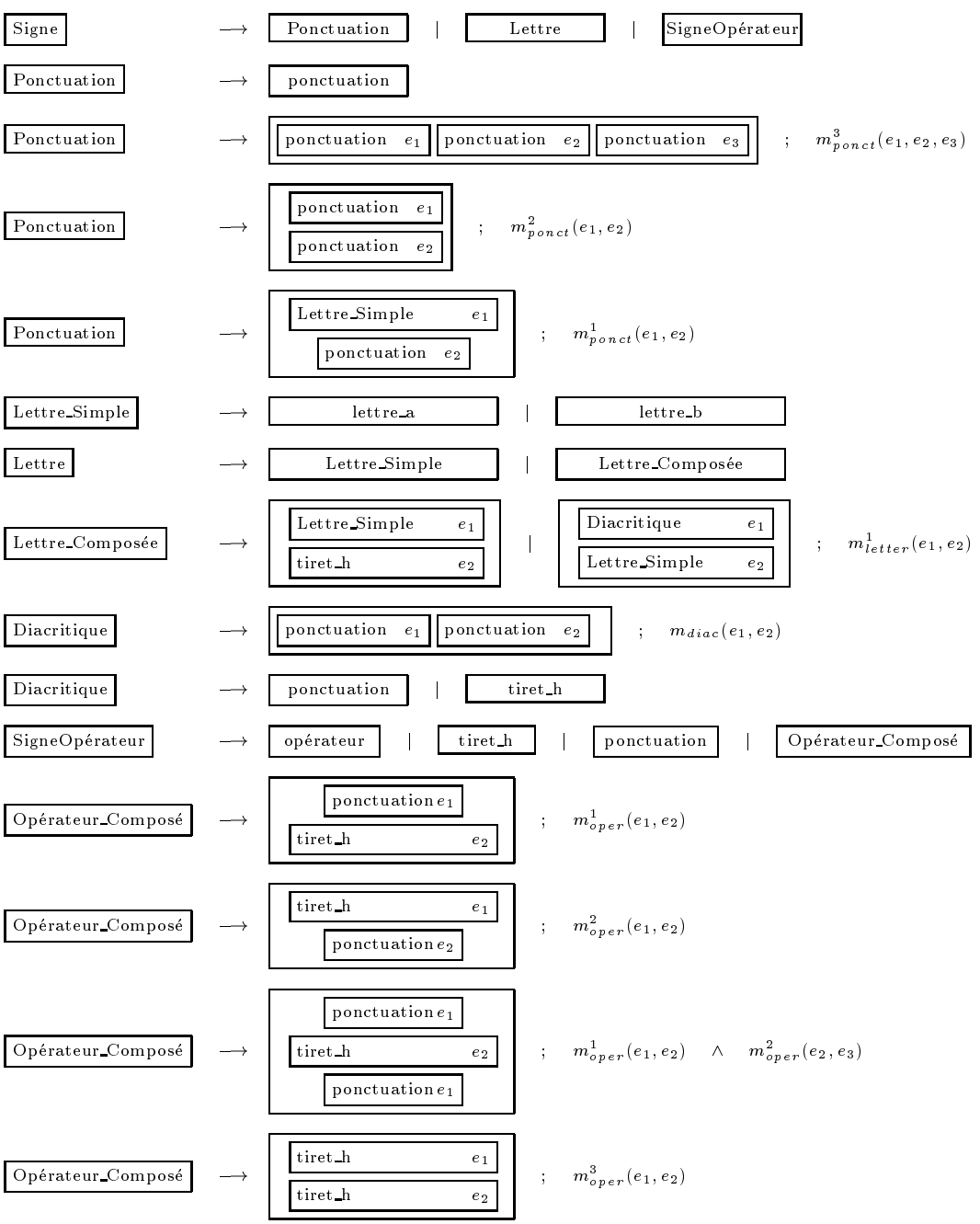
$$m_{graph}(e_1, e_2) \quad = \quad (d_y(e_1, e_2) \leq d_y^l) \quad \wedge \quad (y_2(e_1) < y_1(e_2)) \quad \wedge \quad (D_X(e_1, e_2) == D_x(e_1, e_2))$$

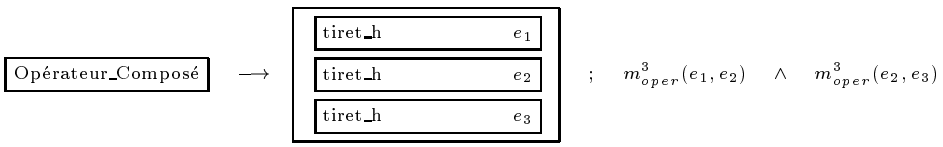
Généralement, dans l'aspect graphique d'une illustration, la légende est centrée par rapport aux illustrations.

5.6 Aspects graphiques élémentaires

Les aspects graphiques élémentaires, comme leur nom l'indique, désignent la structure physique des entités élémentaires dans la composition des microstructures : signes et symboles. Nous avons limité notre analyse aux structures les plus courantes couvrant les caractères, les symboles mathématiques et les composantes graphiques.

Signe : règles traduisant la structure graphique courante des caractères.





Règles 5.7: Structures graphiques élémentaires des signes.

Dans les règles 5.7, les prédicats métriques sont donnés, pour chaque type de signes, par une conjonction logique définie comme ci-après :

$$m^3_{ponct}(e_1,e_2,e_3) \quad = \quad m_{ponct}(e_1,e_2) \quad \wedge \quad m_{ponct}(e_2,e_3)$$

et

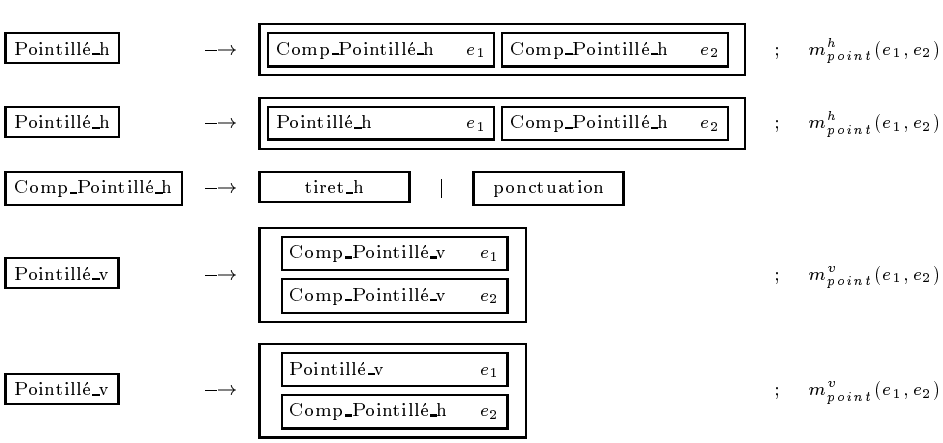
$$\begin{aligned}
m_{ponct}(e_1,e_2) &= (d_x(e_1,e_2) \leq d^c_x) \wedge (I_y(e_1,e_2) > 0) \wedge (x_2(e_1) \leq x_1(e_2)) \wedge (U_y(e_1,e_2) - I_y(e_1,e_2) \leq 0.2 \times U_y(e_1,e_2)) \\
m^2_{ponct}(e_1,e_2) &= (d_y(e_1,e_2) \leq d^d_y) \wedge (I_x(e_1,e_2) > 0) \wedge (y_2(e_1) \leq y_1(e_2)) \wedge (2 \times D_X(e_1,e_2) \leq I_x(e_1,e_2)) \\
m^1_p(e_1,e_2) &= (d_y(e_1,e_2) \leq d^d_y) \wedge (I_x(e_1,e_2) > 0) \wedge (y_2(e_1) \leq y_1(e_2)) \wedge (D_x(e_1,e_2) \leq I_x(e_1,e_2)) \\
m^1_{oper}(e_1,e_2) &= (d_y(e_1,e_2) \leq d^d_y) \wedge (I_x(e_1,e_2) > 0) \wedge (y_2(e_1) \leq y_1(e_2)) \wedge (I_x(e_1,e_2) \approx l(e_1)) \\
m^2_{oper}(e_1,e_2) &= (d_y(e_1,e_2) \leq d^d_y) \wedge (I_x(e_1,e_2) > 0) \wedge (y_2(e_1) \leq y_1(e_2)) \wedge (I_x(e_1,e_2) \approx l(e_2)) \\
m^3_{oper}(e_1,e_2) &= (x_1(e_1) == x_1(e_2)) \wedge (x_2(e_1) == x_2(e_2))
\end{aligned}$$

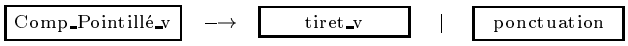
avec

$$\begin{aligned}
m^1_{letter}(e_1,e_2) &= m^1_{ponct}(e_1,e_2) \\
m_{diac}(e_1,e_2) &= m_{ponct}(e_1,e_2)
\end{aligned}$$

Idéalement, dans tout document, les profils verticaux des points "." composant "..." doivent parfaitement se recouvrir ($I_y == U_y$), ce qui n'est malheureusement jamais le cas dans la pratique à cause des bruits. Nous tenons compte de cette irrégularité dans le prédicat métrique m_{ponct} en admettant une tolérance au décalage empiriquement estimée à 20%. En revanche, dans les compositions du type ":", l'intersection du profil vertical des deux ".", devant parfaitement se recouvrir, est influencée par la fonte (roman ou italique). Nous prenons en compte cette particularité dans le prédicat métrique m^2_{ponct} en admettant une tolérance au décalage empiriquement estimée à 50%. Cette dernière observation est aussi valable pour les signes ayant un point diacritique.

Pointillé : Règles traduisant la structure graphique des pointillés.





Règles 5.8: Structures graphiques élémentaires des composantes graphiques.

Dans les règles 5.8, les prédicats métriques m^h_{point} et m^v_{point} sont respectivement les mêmes que ceux définis pour les signes du type "..." et ":" (cf. règles 5.7) :

$$m^h_{point}(e_1, e_2) = m_{ponct}(e_1, e_2)$$
$$m^v_{point}(e_1, e_2) = m^2_{ponct}(e_1, e_2)$$

Conclusion

Dans ce chapitre, nous avons présenté les règles régissant l'aspect graphique usuel des microstructures les plus courantes. Ces règles ont été fondées, d'une part, sur l'exploitation de la systématique des espaces régulant l'aspect graphique des microstructures et, d'autre part, sur le formatage conventionnel de ces dernières. Dans le chapitre 6, nous décrivons notre approche de reconnaissance des microstructures.

Chapitre 6

Reconnaissance de microstructures

Dans ce chapitre, nous traitons le problème de la reconnaissance des microstructures qui dans notre système commence par la classification des composantes pésentée à la section 6.1. Par la suite, nous présentons deux approches pour la reconnaissance des microstructures : (a) la première, présentée dans la section 6.2, suit une stratégie descendante et est basée sur l'analyse des rectangles structurants; (b) la seconde, présentée dans la section 6.3, a été motivée par les limites de la première. Cette dernière approche est guidée par l'ensemble des règles gouvernant l'aspect graphique usuel des microstructures. La section 6.4 est consacrée à l'estimation des lignes de base alors que la section 6.5 définit une notion de tri spatial sur l'arborescence des structures physiques.

6.1 Classification des composantes connexes

Dans cette section, nous présentons après l'extraction des composantes connexes, leur classification en fonction de leurs caractéristiques typographiques. Nous désignons dans ce qui suit une ou plusieurs composantes connexes par CCX.

6.1.1 Définition et objectifs

Définition

Soit $I[x, y]$ une image binaire : on désigne par composante connexe tout ensemble de pixels noirs c, dans l'image I tel que :

$$\forall I[x_i, y_i] \in c, \exists I[x_j, y_j] \in c \quad | \quad (x_i - x_j)^2 + (y_i - y_j)^2 \leq 2$$

Objectif

L'objectif de la classification des CCX est de déterminer les structures (symboles) terminales (cf. section 5.1.4) utilisées dans les règles gouvernant l'aspect graphique usuel des microstructures (cf. section 5.2 à 5.6). La classification est fondée sur une étude statistique de la taille et du gris typographique des composantes connexes. Elle consiste à répartir les CCX en 6 classes qui seront par la suite affinées (cf. section 6.1.5) :

1. la classe *lettre* est constituée de CCX de taille dominante; désignons cette classe par L;
2. la classe *ponctuation* est constituée de CCX de taille relativement plus petite par rapport aux lettres; désignons cette classe par P;

3. la classe *graphique* est constituée de CCX de taille beaucoup plus grande par rapport aux lettres; désignons cette classe par G;

4. la classe *symbole* est constituée de CCX de taille intermédiaire entre celle des lettres et celle des graphiques; désignons cette classe par S;

5. la classe *filet_h* est constituée de CCX dont la largeur est nettement supérieure à la hauteur qui, en général, est inférieure à la hauteur des lettres; désignons cette classe par F_h;

6. la classe *filet_v* est constituée de CCX dont la hauteur est nettement supérieure à la largeur qui, en général, est inférieure à la largeur des lettres; désignons cette classe par F_v;

Cette classification intuitive est fondée sur le fait qu'un document composite est généralement composé d'une forte proportion de texte conformément à notre définition.

6.1.2 Extraction de composantes connexes

L'extraction des CCX est réalisée au moyen d'un algorithme qui, balayant l'image ligne par ligne, fusionne les segments de pixels noirs adjacents d'une ligne à l'autre. La fusion d'un segment à une composante connexe en cours d'extraction a lieu dans une des cinq configurations de la figure 6.1.

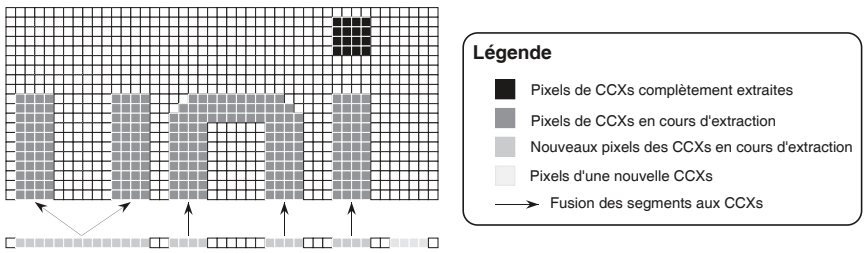


Figure 6.1: Processus d'extraction des composantes connexes.

L'extraction d'une composante connexe est terminée lorsque, après le balayage de la ligne courante, elle n'a subi aucune modification. Cette technique, quoique fiable et efficace, consomme plus de temps de calcul comparée aux autres primitives collaborant à la reconnaissance d'un document (cf. chapitre 10 consacré à l'évaluation des primitives de reconnaissance).

6.1.3 Classification par regroupement des plus proches voisins

La classification est réalisée au moyen d'une méthode statistique de reconnaissance basée sur la technique d'agglomération itérative [1]. Deux caractéristiques ont été retenues : il s'agit de la hauteur et de la largeur des composantes connexes. Après avoir expérimenté plusieurs algorithmes statistiques usuels dans le domaine de la classification dont, les k-plus proches voisins, l'optimisation itérative (cas particulier des k-plus proche voisins avec $k = 1$), la programmation dynamique et le regroupement hiérarchique, notre choix s'est porté sur une variante de l'optimisation itérative (cf. algorithme 6.1) qui, en plus de son efficacité, nous a semblé plus fiable pour la classification des CCX.

Soient
h_f : la hauteur la plus fréquente des CCX,
l_f : la largeur la plus fréquente des CCX,
N : le nombre de CCX,
$x_i = (h(CCX_i), l(CCX_i))^t$: vecteur caractéristique de la i-ème CCX,

[1] Ce genre de technique est connue sous le nom de *Clustering* en anglais

$X = \{x_i\}$: ensemble des vecteurs caractéristiques,
L : le nombre de classes,
$X_j = \{x_j\}$: ensemble des vecteurs attribués à la classe j,
N_j : le nombre de CCX attribuées à la classe j,
$c_j = (h_j, l_j)^t$: vecteur caractéristique moyen de la classe j,

Initialisation
L =?, (cf. section 6.1.4);
c_j =?, (cf. section 6.1.4);
$N_j = 0$, $j = 1 \ldots L$;

TANT QUE $X \neq \{\}$ FAIRE
Soient $(k \leq N)$ et $(l \leq L)$ Tels que
$\|x_k - c_l\| = \min \|x_i - c_j\|$, $i = 1 \ldots N$ et $j = 1 \ldots L$;
$X = X - \{x_k\}$;
$N = N - 1$;
$X_l = X_l + \{x_k\}$;
$N_l = N_l + 1$;
$c_l = c_l + (x_k - c_l)/N_l$;
FIN TANT QUE;

Algorithme 6.1: Classification des composantes connexes.

6.1.4 Initialisation des classes

Si l'on sait à ce stade comment extraire les composantes connexes et leurs caractéristiques, il reste deux problèmes non encore résolus avant que l'algorithme 6.1 ne devienne fonctionnel :

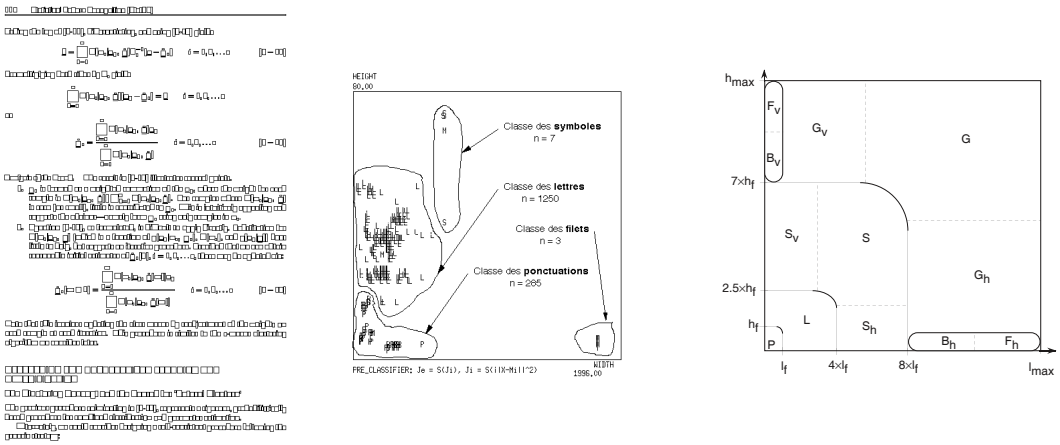
1. la détermination du nombre effectif de classes qui dépend de la variation de la taille des CCX composant le document spécifique en cours de reconnaissance,
2. l'initialisation des vecteurs caractéristiques moyens c_j pour chaque classe.

Ces deux problèmes, propres aux algorithmes classiques de classification non supervisée, ont trouvé une solution dans l'étude de la distribution des CCX dans l'espace des vecteurs caractéristiques. La figure 6.2.a, illustrant le résultat de cette classification sur le document (Doc. 3) de la figure 6.8, est assez représentative de la répartition observée dans ce genre de distribution. Il est important de noter, dans cette figure, que la hauteur et la largeur des CCX ne sont pas représentées à une même échelle dans un souci de clarté dans la distribution. L'étude réalisée sur un grand nombre de documents a révélé, pour la distribution de leurs CCX dans l'espace hauteur-largeur, la répartition suivante :

- une forte concentration autour du point $(h_h, 2.l_f)^t$: il s'agit de la classe des lettres, les documents composites étant constitués à dominance de blocs textuels;
- une autre concentration d'importance relative par rapport à la précédente et non loin de celle-ci : il s'agit des signes diacritiques et des signes de ponctuation;
- le reste de la distribution est constituée de CCX exotiques comprenant les symboles mathématiques, les blocs graphiques, les filets et les barres de fraction. Contrairement aux lettres et aux ponctuations, les CCX exotiques ne sont pas toujours présentes dans un document spécifique.

Initialisation du nombre de classes : L

Un prétraitement sur la taille des CCX permet de flairer, parmi elles, la présence de CCX exotiques. Par conséquent, l'initialisation du nombre de classes L est assujetie aussi bien à la hauteur qu'à la largeur maximales des CCX extraites du document à traiter. Ainsi, on évite de rechercher des blocs graphiques si l'on sait, à priori, que la taille maximale des CCX est assez proche de la taille la plus fréquente des CCX qui correspond à la taille des lettres minuscules. Cet ajustement automatique du nombre de classes L en fonction du contenu des documents a permis de consolider la fiabilité de la classification.



(a) (b) (c)

Figure 6.2: Partition initiale de l'espace des vecteurs caractéristiques.

Adaptation à la distance euclidienne

L'utilisation de la distance euclidienne pour déterminer la proximité des vecteurs x_i, par rapport aux vecteurs caractéristiques moyens c_j, provoque des recoupements non désirés entre certaines classes, par exemple, entre les classes S et L. Pour éviter cet inconvénient, nous avons réduit le champ d'action de la classe des symboles S en la subdivisant en trois sous-classes : (1) la classe des symboles horizontaux désignée par S_h, (2) celle des symboles verticaux désignée par S_v et (3) celle des autres symboles désignée par S (cf. figure 6.2.c). Il en a été de même pour la classe des graphiques qui a été décomposée en trois sous classes : G_h, G_v et G. Pour ce qui est de la classe des filets, à savoir *filet_v* et *filet_h*, chacune d'elle a été subdivisée en deux sous classes :

1. la classe *barre_h*, que nous désignons par B_h, est constituée de filet_h de taille moyenne; ces derniers sont susceptibles d'être des barres de fraction et peuvent également intervenir dans la composition des pointillés horizontaux;

2. la classe *barre_v*, que nous désignons par B_v, est constituée de filet_v de taille moyenne; ces derniers peuvent intervenir dans la composition des pointillés verticaux.

Initialisation des vecteurs caractéristiques moyens : c_j

L'initialisation de l'algorithme de classification consiste à attribuer à chacune des douze classes, ainsi définies, un vecteur caractéristique moyen initial. Contrairement aux techniques classiques dans lesquelles l'initialisation est généralement fondée sur une préclassification naïve, celle-ci est fondée sur une partition initiale de l'espace des vecteurs caractéristiques comme le montre la figure 6.2.c.

$$M_P = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad M_L = \begin{pmatrix} h_f \\ 2l_f \end{pmatrix} \quad M_S = \begin{pmatrix} 4.5h_f \\ 6l_f \end{pmatrix} \quad M_{S_v} = \begin{pmatrix} 4.5h_f \\ 2l_f \end{pmatrix}$$

$$M_{S_h} = \begin{pmatrix} h_f \\ 6l_f \end{pmatrix} \quad M_{B_h} = \begin{pmatrix} 0 \\ 10l_f \end{pmatrix} \quad M_{B_v} = \begin{pmatrix} 9h_f \\ 0 \end{pmatrix} \quad M_{F_h} = \begin{pmatrix} 0 \\ 35l_f \end{pmatrix}$$

$$M_{F_v} = \begin{pmatrix} 30h_f \\ 0 \end{pmatrix} \quad M_G = \begin{pmatrix} 30h_f \\ 35l_f \end{pmatrix} \quad M_{G_h} = \begin{pmatrix} 9h_f) \\ 22l_f \end{pmatrix} \quad M_{G_v} = \begin{pmatrix} 20h_f \\ 10l_f \end{pmatrix}$$

L'évaluation de cet algorithme a donné de très bons résultats qui ont profité à la fiabilité de l'étiquetage des microstructures (cf. section sect fusion hierarchique). L'efficacité de la technique a été considérablement améliorée par l'utilisation d'une table d'adjacence $T[i,j] = \|x_i - c_j\|$. Ainsi, après l'attribution du vecteur x_k à une classe X_l, seule la distance entre celle-ci et chacune des CCX non encore classées est recalculée, cette dernière étant la seule à subir une remise à jour de son vecteur caractéristique moyen c_l.

6.1.5 Raffinement de la classification

Dans l'optique d'augmenter la fiabilité de la reconnaissance des microstructures, guidée par des règles, il a été nécessaire d'affiner la classification des CCX étiquetées comme *lettres*. Le raffinement a consisté à répartir les *lettres* en *lettres minuscules*, *autres lettres*, *tirets horizontaux*, *tirets verticaux* et *opérateurs mathématiques*. Les lettres minuscules ont la particularité d'avoir pour hauteur la hauteur la plus fréquente des CCX, soit h_f. Les tirets sont par contre relativement allongés alors que les opérateurs mathématiques sont caractérisés par une graisse assez faible. Le raffinement a été réalisé au moyen de la classification hiérarchique illustrée à la figure 6.3. Une évaluation quantitative de la classification des CCX est donnée dans le chapitre 10.

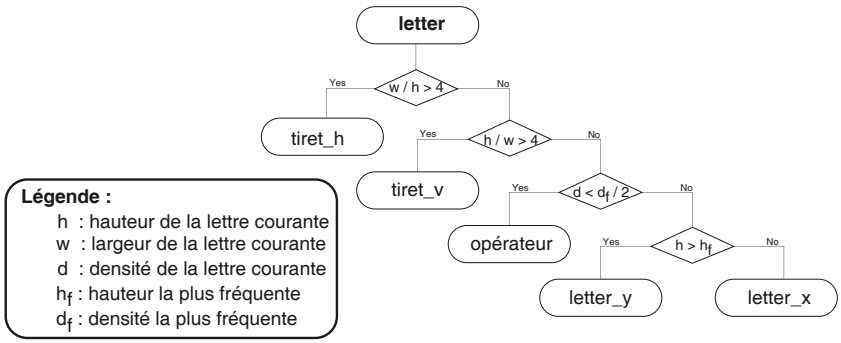


Figure 6.3: Raffinement de la classification des lettres.

6.2 Etiquetage suivant une approche de découpe hiérarchique

Dans cette approche, la reconnaissance des microstructures est réalisée au moyen des rectangles structurants extraits du document à traiter. Elle consiste, d'une part, à poursuivre la découpe hiérarchique jusqu'au niveau des mots et des signes et, d'autre part, à attribuer à chaque découpe hiérarchique, ainsi déterminée, une étiquette logique conformément au modèle décrit dans la figure 4.7. Dans ce qui suit, nous désignons par :

- SR un rectangle structurant,
- SRH un rectangle structurant horizontal,
- SRV un rectangle structurant vertical,
- E une fonction qui retourne soit l'enveloppe d'un RS, soit celle d'une entité physique.

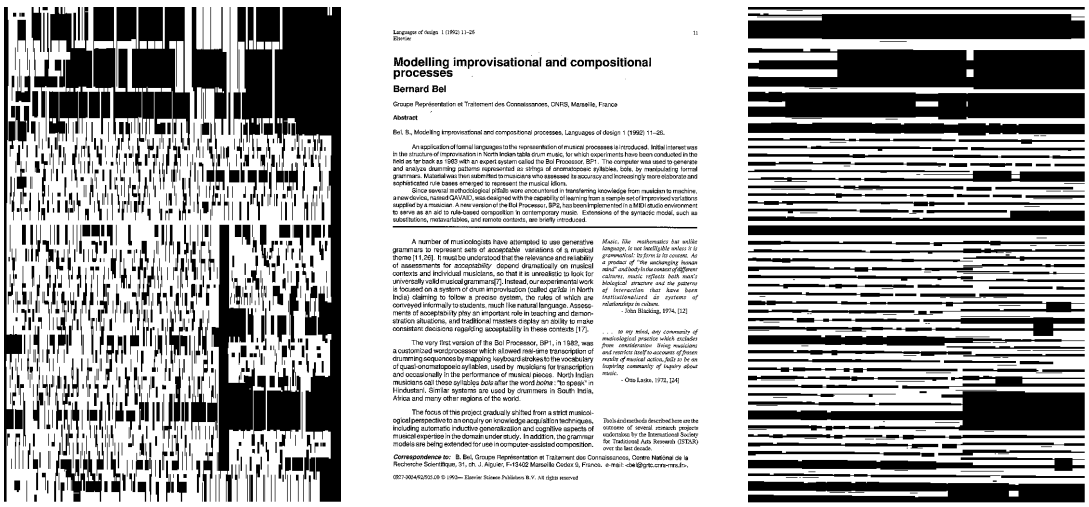
6.2.1 Extraction des rectangles structurants

L'extraction des rectangles structurants est réalisée en subdivisant de façon itérative l'enveloppe d'une page, par rapport à ses constituants, de sorte qu'il ne reste à la fin de l'itération que des espaces qui délimitent les objets imprimés. Soit p une page spécifique, $C = \{c_k\}$ l'ensemble des blocs

résultant des traitements antérieurs sur p (ex. composantes connexes, microstructures ou blocs de niveau hiérarchique intermédiaire) excepté les filets que nous supposons être des séparateurs lorsqu'ils ne désignent pas des barres de fraction. Soit $T = \{t_i\}$, l'ensemble courant des rectangles structurants initialement constitué de $E(p)$ enveloppe de p. L'extraction est concrètement réalisée au moyen de l'algorithme 6.2.

extraire_rectangles_structurants(p, h_{min}, l_{min}):
  $C = \{c_k\}$;
  $T = \{E(p)\}$;
  POUR CHAQUE $c_k \in C$ FAIRE
    $T' = \{\}$;
    POUR CHAQUE $t_i \in T$ FAIRE
      $T = T - \{t_i\}$;
      SI $E(p) \cap E(ti) \neq 0$ ALORS
        SI $c_k.y1 - t_i.y_1 > h_{min}$ ALORS
(a) $T' = T' + \{(t_i.x_1, t_i.x_2, t_i.y_1, c_k.y_1)\}$;
        FIN SI;
        SI $t_i.y_2 - c_k.y2 > h_{min}$ ALORS
          $T' = T' + \{(t_i.x_1, t_i.x_2, c_k.y_2, t_i.y_2)\}$;
        FIN SI;
        SI $c_k.x1 - t_i.x_1 > l_{min}$ ALORS
          $T' = T' + \{(t_i.x_1, c_k.x_1, t_i.y_1, t_i.y_2)\}$;
        FIN SI;
        SI $t_i.x_2 - c_k.x2 > l_{min}$ ALORS
          $T' = T' + \{(c_k.x_2, t_i.x_2, t_i.y_1, t_i.y_2)\}$;
        FIN SI;
      SINON $T' = T' + \{t_i\}$; FIN SI;
    FIN POUR CHAQUE;
    $T = T'$;
  FIN POUR CHAQUE;
FIN **extraire_rectangles_structurants**;

Algorithme 6.2: Extraction des rectangles structurants.



(a) : RSV (b) : Doc. 1 (c) : RSH

Figure 6.4: Rectangles structurants verticaux et horizontaux du document (Doc. 1).

La figure 6.4 montre des rectangles structurants extraits de l'image du document 6.4.b. Pour augmenter l'efficacité de l'algorithme, nous avons introduit deux améliorations.

1. La première amélioration consiste à paramétrer l'algorithme par deux seuils h_{min} et l_{min} représentant respectivement la hauteur et la largeur minimales des rectangles structurants

auxquels l'on s'intéresse, les autres étant systématiquement éliminés. Cette paramétrisation a pour conséquence de limiter la subdivision hiérarchique. Par exemple, au cours d'une segmentation en blocs, la connaissance du seuil maximum des espaces inter-mots d_x^m et des espaces interligne d_y^l permet de limiter l'extraction aux rectangles structurants horizontaux de hauteur $> d_y^l$ et aux rectangles structurants verticaux de largeur $> d_x^m$.

2. La seconde amélioration consiste à trier les éléments de C et à maintenir également triés les éléments de T durant leur extraction. Nous avons, pour ce qui nous concerne, trié les éléments $c_k \in C$ par rapport à leur première ordonnée (c.-à-d. $y_1(c_k)$) et ceux de T par rapport à leur deuxième ordonnée (c.-à-d. $y_2(t_i)$). Cette amélioration revient dans l'algorithme 6.2 à poser $T = T_{trié} + T_{nontrié}$ et à remplacer, d'une part, T par $T_{nontrié}$ et, d'autre part, dans la ligne etiquetée (a), T' par $T_{trié}$.

Les rectangles structurants, dont l'extraction ne dépend pas de la nature des éléments de C, ont été utilisés également dans la reconnaissance des macrostructures, d'une part, pour segmenter les documents en régions et, d'autre part, pour segmenter les régions en blocs (cf. chapitre 9).

6.2.2 Découpe hiérarchique des microstructures

La découpe hiérarchique d'un bloc b consiste à découper récursivement ce dernier jusqu'au niveau de ses signes. Soient b un bloc représenté par l'ensemble de ses composantes connexes $C = \{c_k\}$ et r son enveloppe. Soit maintenant $T = \{t_i\}$ l'ensemble des rectangles structurants extraits du bloc b au moyen de l'algorithme 6.2 dont les paramètres ont été définis comme suit : $l_{min} = d_x^c$ et $h_{min} = 0$. La découpe hiérarchique du bloc b est réalisée au moyen de l'algorithme 6.3.

---

découpe_microstructures(r, C, T):
  C_b : désigne l'ensemble des constituants du sous-bloc courant;
  $r_b = r$: où r_b désigne l'enveloppe du sous-bloc courant;
  $T_v = \{t_i \in T | (t_i.x_2 - t_i.x_1 > d_x^c) \wedge (t_i.x_1 > r.x_1) \wedge (t_i.x_2 < r.x_2) \wedge (t_i.y_1 < r.y_1) \wedge (t_i.y_2 > r.y_2)\}$
  SI $T_v \neq \{\}$ ALORS
    découpe_mots(r, C, T_v, T);
  SINON
    $T_h = \{t_i \in T | (t_i.y_1 > r.y_1) \wedge (t_i.y_2 < r.y_2) \wedge (t_i.x_1 < r.x_1) \wedge (t_i.x_2 > r.x_2)\}$
    SI $T_v \neq \{\}$ ALORS
      découpe_lignes(r, C, T_v, T);
    SINON (* Plus de découpe possible *) FIN SI;
  FIN SI;
FIN **découpe_microstructures**;

Algorithme 6.3: Découpe hiérarchique des microstructures.

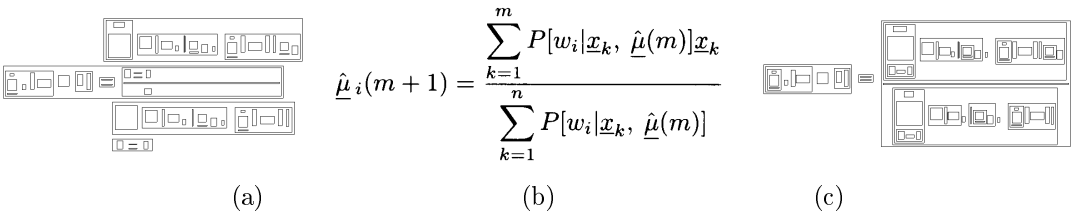
La découpe en lignes est réalisée au moyen d'un algorithme semblable à celui de la découpe en mots (cf. algorithme 6.4).

---

découpe_mots(r, C, T):
  Trier les éléments de T_v par rapport à la coordonnée x_1 de leur enveloppe;
  POUR CHAQUE i ALLANT DE 1 A $1 + N(T_v)$ FAIRE
    SI $i \leq N(T_v)$ ALORS $r_b.x_2 = T_v[i].x_1$;
    SINON $r_b.x_2 = r.x_2$ FIN SI;
    $C_b = \{c_k \in C | B(c_k) \subset r_b\}$;
    découpe_microstructures(r_b, C_b, T);
    $C = C - C_b + \{C_b\}$;
    SI $i \leq N(T_v)$ ALORS $r_b.x_1 = T_v[i].x_2$; FIN SI;
  FIN POUR CHAQUE;
FIN **découpe_mots**;

Algorithme 6.4: Découpe en mots des microstructures.

Dans l'agorithme 6.3, nous avons privilégié la découpe en colonnes par rapport à celle en lignes. Ce choix se prête mieux pour l'analyse des expressions mathématiques; les structures hiérarchiques (a) et (c) de la figure 6.5 sont là pour s'en convaincre : (a) est obtenue en commençant l'analyse du bloc (b) par une découpe en lignes et (c) est obtenue en commençant par une découpe en mots. En revanche, ce choix peut avoir un revers redoutable lors d'un mauvais formatage : celui d'interpréter comme séparateurs de colonnes les *ruisseaux*[2] provenant d'une mauvaise justification des lignes dans le formatage des blocs de texte.



$$\underline{\hat{\mu}}_i(m+1) = \frac{\sum_{k=1}^{m} P[w_i|\underline{x}_k,\ \underline{\hat{\mu}}(m)]\underline{x}_k}{\sum_{k=1}^{n} P[w_i|\underline{x}_k,\ \underline{\hat{\mu}}(m)]}$$

(a) (b) (c)

Figure 6.5: Structure hiérarchique d'une expression mathématique.

La pratique courante dans les techniques équivalentes présentées dans la littérature consiste à filtrer manuellement les expressions mathématiques du reste du document afin de les traiter séparément. Notre approche d'analyse a l'avantage dêtre uniforme quel que soit le contenu des documents traités.

6.2.3 Etiquetage de microstructures

L'étiquetage consiste à attribuer à une microstructure hiérarchique une étiquette logique pouvant être par exemple *texte*, *ligne*, *mot*, *terme*, *graphique* ou *photographie*, conformément au modèle de la figure 4.7. L'étiquetage d'une microstructure b est fondé sur une analyse combinée des attributs suivants :

- hauteur de b,
- largeur de b,
- densité de b,
- axe de découpe conduisant au premier niveau hiérarchique de b,
- étiquette de la structure englobante de b,

Cette approche d'étiquetage n'a pas été explorée davantage faute de résultats tangibles lors de nos premières expérimentations; les limites rencontrées font l'objet de la section 6.2.4.

6.2.4 Limites

Si la technique de découpe récursive est appropriée pour l'analyse de la structure hiérarchique des documents textuels (pour autant qu'ils ne soient pas inclinés) et dans une certaine mesure, pour celle des expressions mathématiques, elle est par contre très peu fiable pour ce qui est de l'analyse des blocs grahiques ou photographiques. En revanche, il ressort de notre expérimentation que la découpe récursive est très fiable pour la reconnaissance des tableaux présentée dans les sections 9.3.2 et 9.3.3. Pour pousser plus loin les limites montrées par l'approche présentée dans cette section, nous nous sommes intéressés à une autre approche de reconnaissance basée sur une stratégie plutôt mixte.

[2] Le terme typographique *ruisseaux* est appelé *rivers* en anglais.

6.3 Etiquetage suivant une approche mixte

Il s'agit d'une approche qui combine la technique de découpe hiérarchique présentée dans la section 6.2 et une technique de fusion hiérarchique guidée par les règles gouvernant l'aspect graphique usuel des microstructures (cf. chapitre 5). La fusion est réalisée au moyen d'un analyseur syntaxique paramétré par la règle décrivant le type de microstructures à reconnaître. L'étiquetage logique d'une microstructure est effectué au fur et à mesure que l'on progresse dans le regroupement de ses constituants. Le regroupement est réalisé au moyen d'un algorithme de la famille des *k-plus proches voisins*[3] avec $k = 2$, la fusion étant réalisée séquentiellement selon l'axe X ou Y. Le regroupement est contraint par les prédicats métriques exigés par la règle de production des microstructures à reconnaître.

6.3.1 Algorithme générique d'étiquetage par fusion

Le principe de cet algorithme consiste à rechercher parmi l'ensemble des blocs résultant des traitements antérieurs, les séquences dont l'aspect graphique est conforme à celle d'une microstructure X donnée.

Soient $C = \{c_k\}$ l'ensemble des constituants étiquetés d'un document spécifique, f une fonction décrivant la règle de production régissant l'aspect graphique usuel du type X de microstructures à reconnaître et g une fonction définissant un ordre de parcours sur C. Soient c_i et c_j deux éléments de C : la fonction f sert à valider la fusion de c_j avec c_i par le biais des contrôles suivants :

1. c_i et c_j sont de types autorisés dans la composition de X,
2. la fusion de c_j et c_i est valide conformément à la règle de production régissant X,
3. c_i et c_j vérifient le prédicat métrique de voisinage régissant la topologie de X.

Quant à la fonction g, elle sert à disposer l'un à côté de l'autre les éléments de C les plus proches selon un critère d'ordre défini par g. La réorganisation des éléments de C a pour but d'augmenter l'efficacité de l'algorithme 6.5 de fusion qui est de type ascendant. Cette réorganisation est fondée sur des critères d'ordre de parcours présentés à la section 6.5.2.

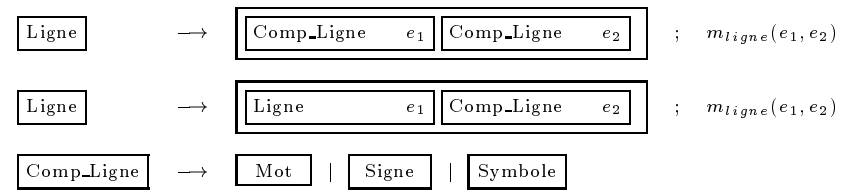
étiquetage_incrémental(C, f, g):
  c : microstructure en cours de fusion;
  $C' = \{\}$: nouvelle structure hiérarchique après fusion;
  trier_selon_critère(C, g);
  POUR CHAQUE i ALLANT DE 1 A $N(C)$ FAIRE
    $c = \{C[i]\}$;
    TANT QUE $(i < N(C)) \wedge f(c, C[i+1])$ FAIRE
      $i = i + 1$;
      $c = c + \{C[i]\}$;
    FIN TANT QUE;
    $C = C - c$;
    $C' = C' + \{c\}$;
  FIN POUR CHAQUE;
  $C = C'$
FIN **étiquetage_incrémental**;

Algorithme 6.5: Algorithme générique d'étiquetage par fusion.

Un exemple d'application : étiquetage des lignes de texte

Soient les règles régissant l'aspect graphique usuel d'une ligne de texte définies dans la section 5.2.2 et reprises ici dans la règle 6.1, pour les besoins de l'exemple.

[3] A ne pas confondre avec la technique d'agglomération itérative du même nom utilisée pour le *clustering*



Règles 6.1: Reprise des règles décrivant l'aspect graphique usuel des lignes.

Soient L_i une ligne de texte et c_k un élément de C ensemble des constituants d'un bloc textuel. La fonction f décrivant la règle 6.1 vérifie :

- d'une part, que L_i est bien une ligne de texte ou, comme c_k, un mot, un signe ou un symbole;
- et d'autre part, que L_i et c_k vérifient le prédicat métrique m_{ligne} donné comme suit :

$$\begin{aligned} m_{ligne}(e_1, e_2) \quad = \quad & (d_x^c < d_x(e_1, e_2) \leq c \times d_x^m) \\ & \wedge \quad (I_y(e_1, e_2) > 0) \\ & \wedge \quad ((D_Y(e_1, e_2) \leq I_y(e_1, e_2)) \\ & \quad \vee \quad ((h_{min} < h_f) \quad \wedge \quad (h_{min} < 2 \times I_y(e_1, e_2)))) \end{aligned}$$

où le facteur c est empiriquement estimé à $c = 6/5$ lorsque e_1 est une marque de ponctuation et 1 sinon.

L'ordre défini par la fonction g est celui de la lecture des mots dans un bloc textuel. Il est obtenu en triant les éléments de C une première fois par rapport à la coordonnée x_2 de leur enveloppe et une seconde fois par rapport au profil horizontal de leur enveloppe. L'estimation automatique du seuil métrique d_x^m est présentée dans la section 7.4.

Remarques : Le principe d'analyse étant le même pour toutes les microstructures, notre objectif dans la suite de cette section, est de définir pour chaque type de microstructures un ensemble C, une fonction f et une fonction g servant à leur analyse. Les sections 6.3.2 à 6.3.5 présentent une analyse des microstructures guidée par les règles résultant de l'étude systématique de leur aspect graphique usuel (cf. chapitre 5). Pour chaque type de microstructures, nous illustrons les résultats de reconnaissance sur deux documents types. Ces documents ont été numérotés "Doc. **" (ex. Doc. 3) pour pouvoir les référencer lors de l'évaluation quantitative que nous présentons au chapitre 10. Lorsque c'est nécessaire, nous focalisons l'illustration résultats sur une zone spécifique (au moyen du symbole $\bigoplus$) afin de mieux faire ressortir le détail de ce qui est reconnu. La focalisation privilégie la découpe hiérarchique de la zone en question.

6.3.2 Blocs textuels

La reconnaissance des blocs textuels suppose a priori et dans l'ordre d'abord la reconnaissance des signes, celle des mots puis celle des lignes de texte.

Signes

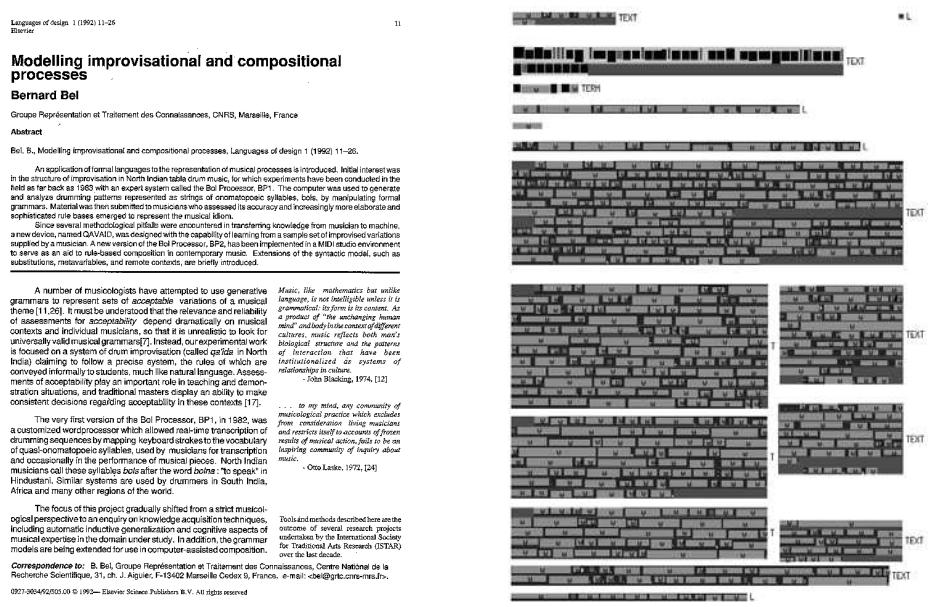
La reconnaissance des signes est basée sur une classification des composantes connexes (cf. section 6.1); celle-ci consiste à répartir les composantes connexes en fonction de leur taille, en plusieurs classes dont les *lettres* et les *ponctuations*. L'ensemble de constituants C utilisé pour l'analyse des signes est formé de lettres et de marques de ponctuation. La fonction f décrit les règles régissant l'aspect graphique usuel des signes (cf. règles 5.7). La fonction g définit un ordre qui consiste à positionner les composantes connexes étiquetées *Ponctuation* par rapport à leurs deux plus proches voisins suivant l'axe Y.

Mots

Pour la reconnaissance des mots, l'ensemble C est constitué de signes résultant du traitement précédent. La fonction f décrit les règles régissant l'aspect graphique usuel des mots (cf. règles 5.3) et la fonction g définit un ordre qui consiste à trier dans l'ordre de lecture les éléments de C. L'ordre de lecture est obtenu au moyen de deux tris successifs : le premier par rapport à la coordonnée x_2 de l'enveloppe des éléments de C et le second par rapport au profil horizontal des enveloppes.

Lignes

La reconnaissance des lignes de texte a fait l'objet de l'exemple présenté dans la section 6.3.1 illustrant le principe de l'algorithme de fusion.



$d_y^d = 6$ $d_x^c = 9$ $d_x^m = 58$ $d_y^b = 79$

Figure 6.6: Reconnaissance de documents textuels (Doc. 1).

Textes

Finalement, la reconnaissance des blocs textuels se base sur les entités provenant du résultat des étapes précédentes. L'ensemble C est alors constitué de *signes*, de *mots* et de *lignes*. La fonction f décrit les règles gouvernant l'aspect graphique usuel des blocs textuels (cf. règles 5.1). L'ordre décrit par la fonction g est celui de la lecture des lignes de texte. Il est obtenu au moyen de deux tris successifs : le premier par rapport à la coordonnée y_1 de l'enveloppe des éléments de C et le second par rapport au profil vertical des enveloppes.

Résultats : Les figures 6.6 et 6.7 illustrent le résultat de l'étiquetage des microstructures sur deux exemples de documents purement textuels. Dans le zoom illustré par $\oplus$, on note la finesse de l'approche dans le regroupement en signes des composantes connexes.

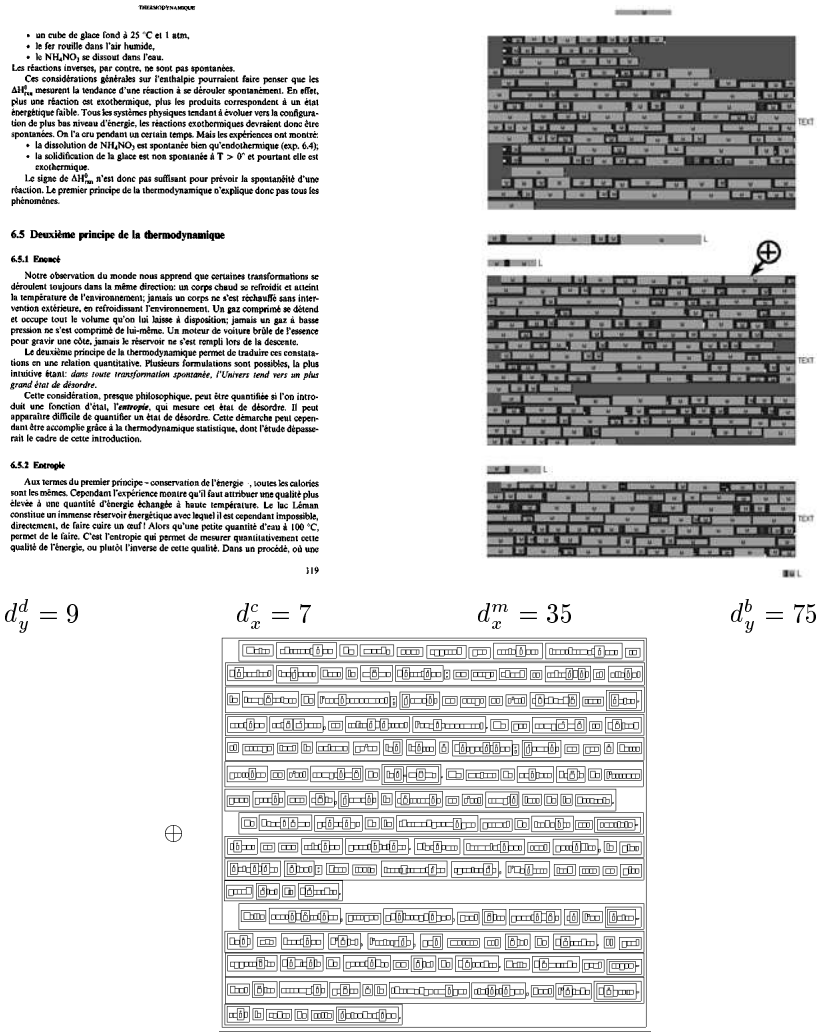


Figure 6.7: Reconnaissance de documents textuels (Doc. 9).

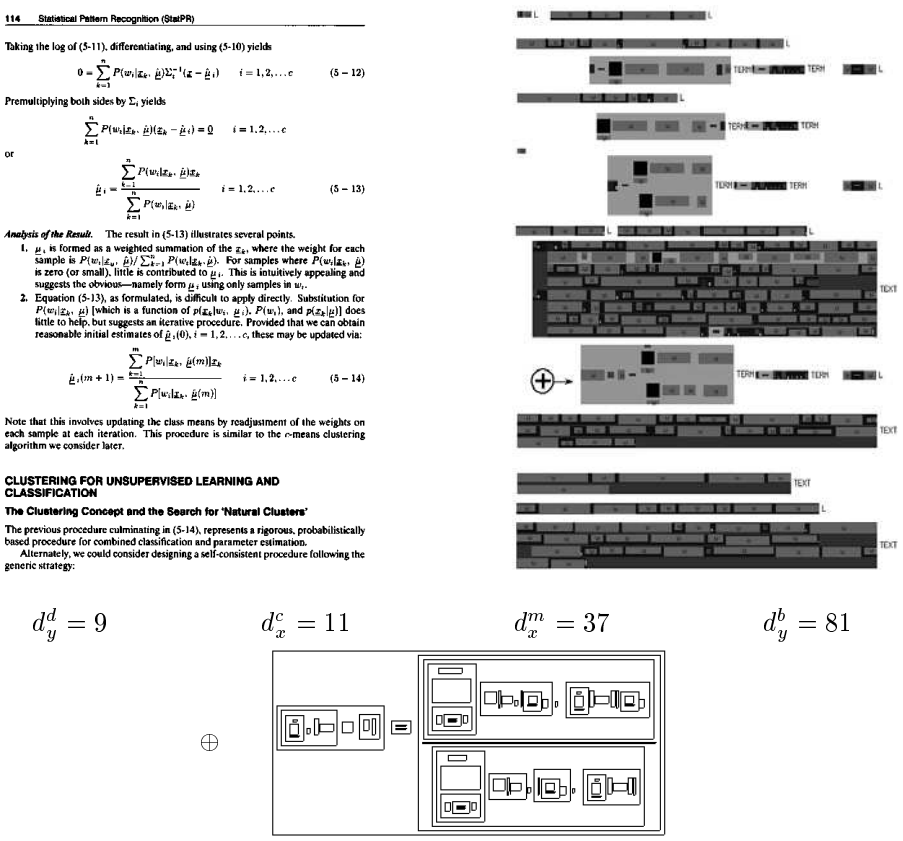
6.3.3 Expressions mathématiques

La reconnaissance des expressions mathématiques suppose à priori celle des *symboles* et des opérateurs mathématiques. Les symboles proviennent de la classification des composantes connexes alors que les opérateurs constituent une sous-classe des signes (ex. "="). La reconnaissance proprement dite est réalisée en alternant la reconnaissance des expressions fractionnaires, exponentielles, bornées, racines ou encore celle des expressions composées (cf. section 5.3). L'itération se poursuit jusqu'à ce qu'il ne soit plus possible de trouver une expression mathématique de niveau hiérarchique plus élévé.

Expressions fractionnaires

La reconnaissance des expressions fractionnaires (ex. $\frac{1}{2}$) est basée sur l'analyse d'un ensemble C constitué de termes, de lignes, de mots et de signes provenant des traitements précédents. La fonction f décrit la règle régissant l'aspect graphique usuel des expressions fractionnaires (cf.

règles 5.4). L'ordre décrit par la fonction g consiste à positionner les barres de fraction, provenant de la classification des composantes connexes, par rapport à leurs deux plus proches voisins suivant l'axe Y.



$d_y^d = 9$ $d_x^c = 11$ $d_x^m = 37$ $d_y^b = 81$

Figure 6.8: Reconnaissance d'expressions mathématiques (Doc. 3).

Expressions exponentielles

Pour la reconnaissance des expressions exponentielles (ex. $e^{\frac{1}{2}}$), l'ensemble C est constitué de termes, de lignes, de mots et de signes provenant des traitements précédents. La fonction f décrit la règle régissant l'aspect graphique des expressions fractionnaires (cf. règles 5.4) alors que la fonction g décrit un ordre de parcours sur les éléments de C.

Expressions bornées

Pour la reconnaissance des expressions bornées (ex. $\sum_{i=1}^{n}$), l'ensemble C est constitué d'entités résultant des traitements précédents. La fonction f décrit la règle gouvernant l'aspect graphique usuel des expressions bornées (cf. règles 5.4). L'ordre décrit par la fonction g consiste à positionner les symboles, provenant de la classification des composantes connexes, par rapport à leurs deux plus proches voisins suivant l'axe Y.

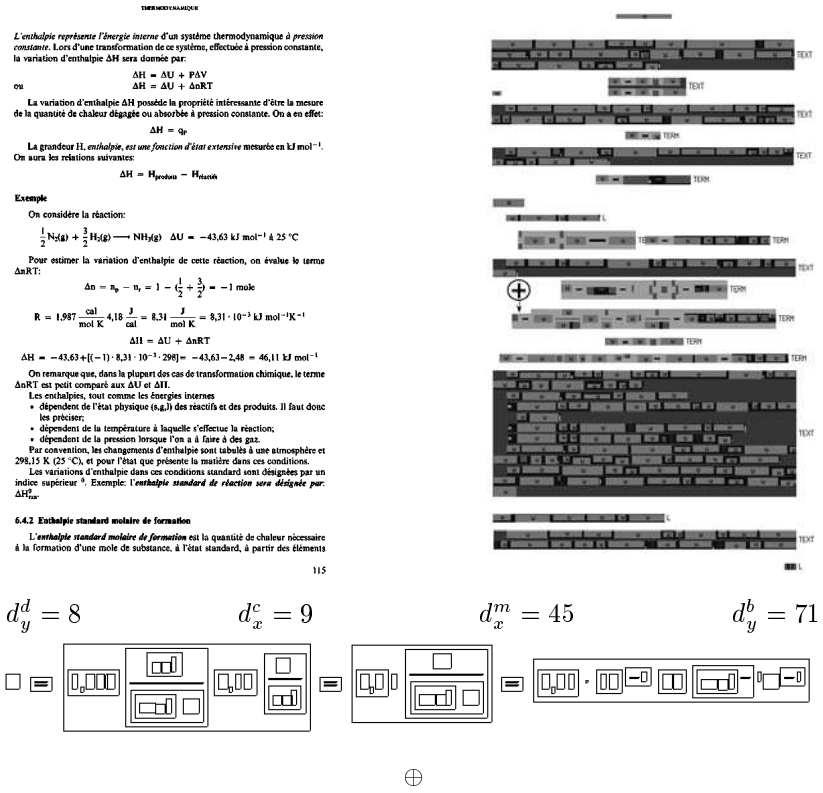


Figure 6.9: Reconnaissance d'expressions mathématiques (Doc. 4).

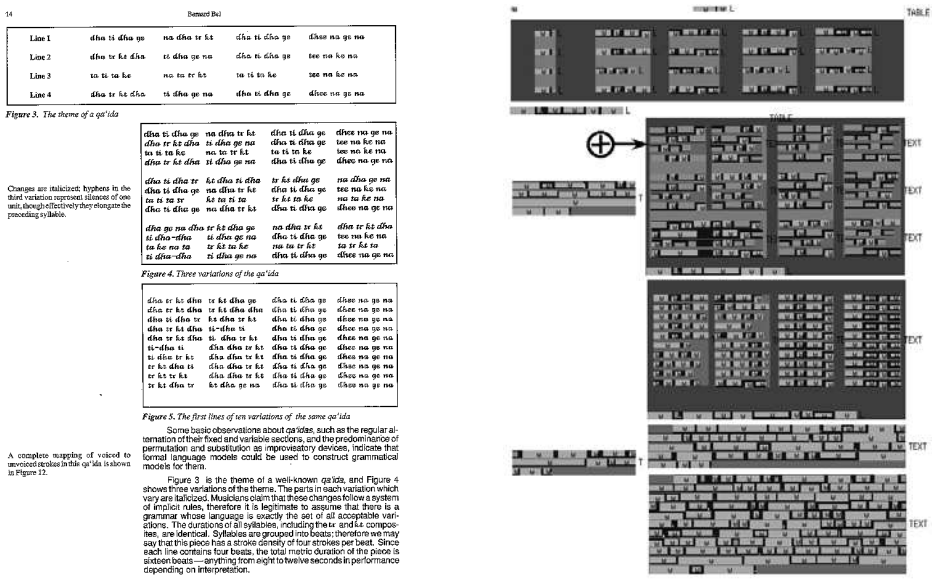
Expressions composées

La reconnaissance des expressions composées (ex. $h(x) = x + 1$) est effectuée en regroupant symboles et opérateurs mathématiques avec leurs deux plus proches voisins suivant l'axe X. L'ensemble C est constitué d'entités résultant des étapes de reconnaissance précédentes. La fonction f décrit la règle régissant l'aspect graphique usuel des expressions composées (cf. règles 5.4) alors que la fonction g décrit dans C un ordre de parcours sur les mots.

Expressions racines

La reconnaissance des expressions racines consiste à rechercher parmi les composantes connexes celles qui englobent une expression mathématique.

Resultats : Les figures 6.8 et 6.9 illustrent le résultat de l'étiquetage des microstructures sur deux exemples de documents contenant des expressions mathématiques. Dans les zooms illustrés par $\oplus$, on note une bonne découpe des expressions mathématiques. Il ne s'agit pas ici d'une découpe sémantique puisque celle-ci est fondée sur une analyse des espaces. Forts de notre expérimentation, nous pensons qu'une analyse plus fine des expressions mathématiques passe par une classification des délimiteurs d'expressions du style "(" et "}".



$d_y^d = 7$ $d_x^c = 10$ $d_x^m = 37$ $d_y^b = 68$

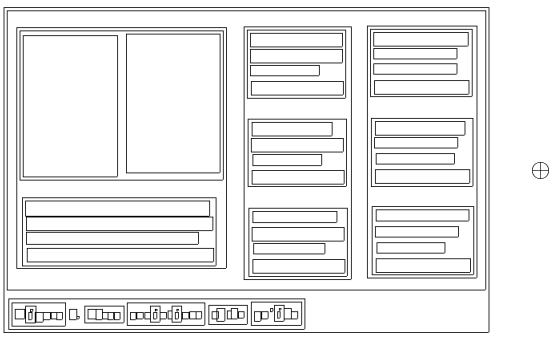


Figure 6.10: Reconnaissance de tableaux (Doc. 6).

6.3.4 Tableaux

La pratique usuelle lors de la saisie d'un tableau, aussi bien dans un système de traitement de texte (ex. Word) que dans un système de formatage (ex. LaTeX), consiste à définir le quadrillage reflétant la forme graphique finale du tableau puis seulement après, à compléter le quadrillage par le contenu proprement dit du tableau. Dès lors, il nous semble plus convenable pour augmenter la fiabilité de la reconnaissance des tableaux de procéder :

1. d'abord à la reconnaissance des cellules qui ne sont rien d'autres que des microstructures,

2. puis au regroupement en tableau des cellules.

Contrairement aux travaux présentés dans la littérature [66, 92, 67, 68], notre approche ne présuppose la présence d'aucun filet. En effet, la présence des filets dans l'aspect graphique des tableaux n'a pour seul but que d'augmenter la lisibilité des tableaux en renforçant la séparation entre les cellules.

L'étiquetage des tableaux est réalisée lors de la segmentation des régions en blocs au moyen de l'algorithme 9.3 procédant par découpe récursive (cf. section 9.3). L'algorithme, profitant de

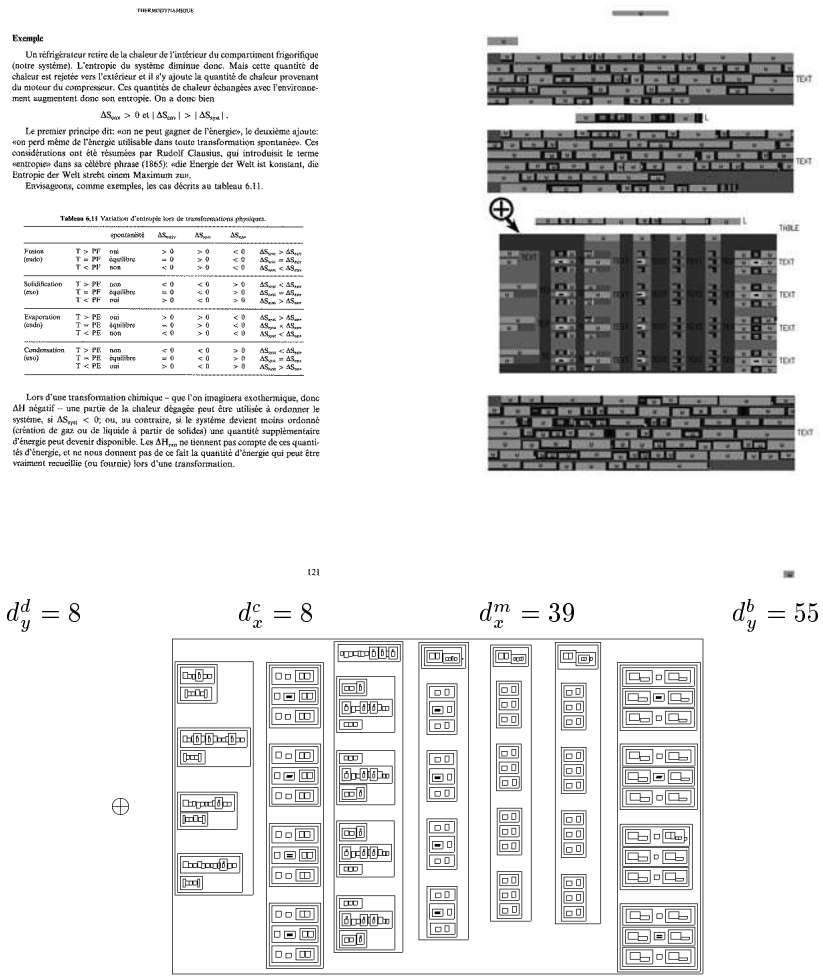


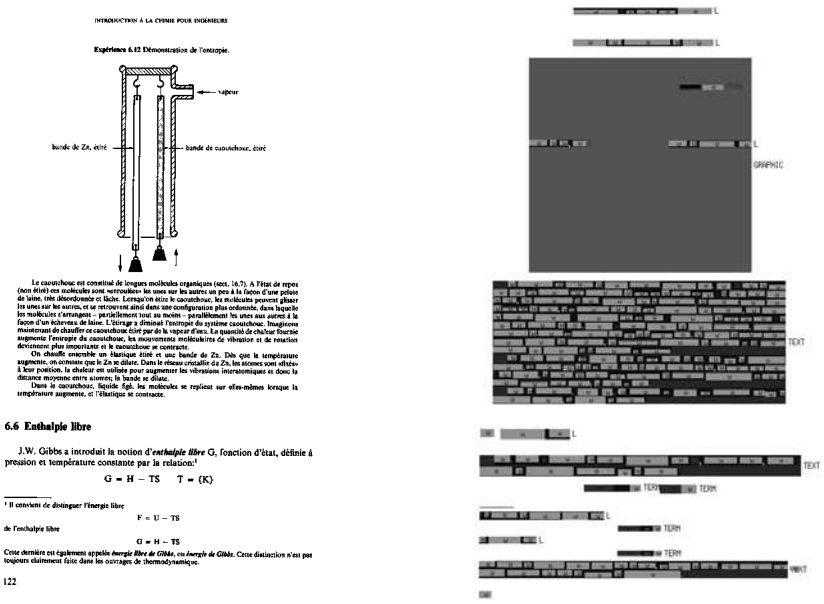
Figure 6.11: Reconnaissance de tableaux (Doc. 5).

la découpe hiérarchique en blocs, extrait et étiquette comme tel, tout groupe de blocs dont la topologie locale est conforme à celle d'un tableau. Il s'agit là, d'un exemple de collaboration entre les deux approches de la reconnaissance : la fusion hiérarchique ayant été utilisée pour déterminer les cellules et la découpe hiérarchique pour déterminer les tableaux.

Résultats : Les figures 6.11 et 6.10 illustrent le résultat de l'étiquetage des microstructures sur deux exemples de documents contenant des tableaux. Dans les zooms illustrés par $\oplus$, on note une bonne découpe en colonnes des tableaux. Quand bien même la découpe en rangées ne ressort pas, ayant privilégié la découpe en colonnes pour raison d'uniformité dans la représentation des entités physiques, elle peut, toutefois, être dérivée de la découpe en colonnes.

6.3.5 Blocs graphiques et photographiques

La reconnaissance des blocs graphiques et photographiques est fondée sur la classification des composantes connexes présentée à la section 6.1. Outre leur étiquetage en tant que tel, la reconnaissance se poursuit sur des blocs graphiques dans l'optique d'extraire la couche textuelle dont



$d_y^d = 8$ $d_x^c = 7$ $d_x^m = 32$ $d_y^b = 55$

Figure 6.12: Reconnaissance de blocs graphiques (Doc. 7).

ils sont composés. L'extraction de la couche textuelle est réalisée en même temps que l'analyse du contenu textuel principal en faisant abstraction de la couche graphique. L'écartement de la couche graphique est rendu possible grâce à la classification des composantes connexes. La faiblesse de cette technique réside dans son incapacité à extraire des informations textuelles de forte inclinaison. Ceci se justifie par notre hypothèse de travail qui suppose le contenu textuel des documents analysés non incliné.

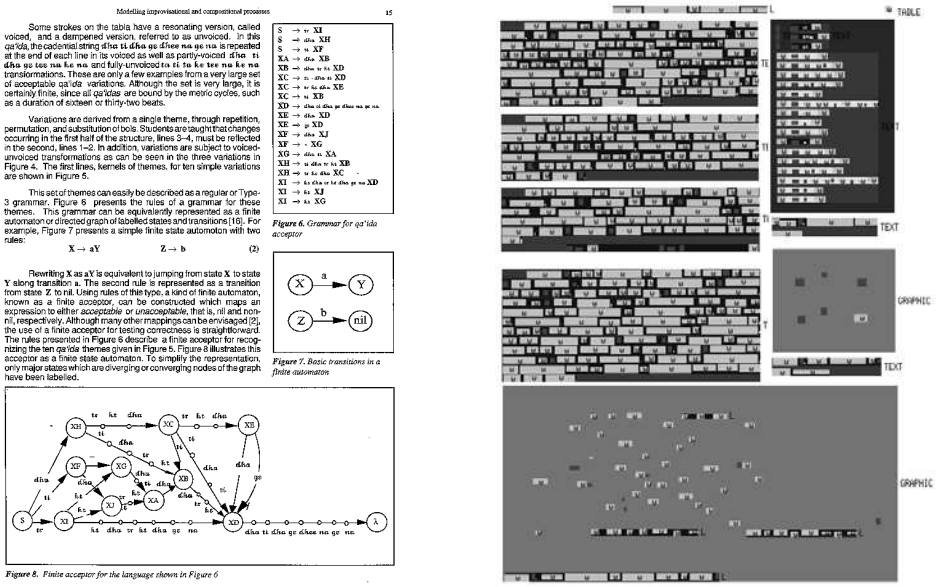
Résultats : Les figures 6.12 et 6.13 illustrent le résultat de l'étiquetage des microstructures sur deux exemples de documents contenant des blocs graphiques. Dans les blocs graphiques, seule la couche textuelle est représentée; le resultat est très satisfaisant.

6.4 Estimation des lignes de base

6.4.1 Définition et motivation

La *ligne de base*, comme l'indique la figure 4.2 qui décrit les attributs typographiques des lignes de texte, désigne la ligne virtuelle sur laquelle sont posées les lettres sans jambage. En dehors de l'intérêt que présente les lignes de base dans les sytèmes de reconnaissance de structures logiques, elles ont servi dans notre système à estimer l'interligne qui a servi de seuil métrique dans la fusion des lignes en blocs textuels.

Parmi les techniques classiques, les plus courantes procèdent par une analyse des profils de projection verticaux. Contrairement à ces dernières, la technique présentée ici s'affranchit de l'image et fonde l'estimation des lignes de base uniquement sur l'analyse des entités physiques étiquetées *Signe*.



$d_y^d = 7$ $d_x^c = 9$ $d_x^m = 61$ $d_y^b = 73$

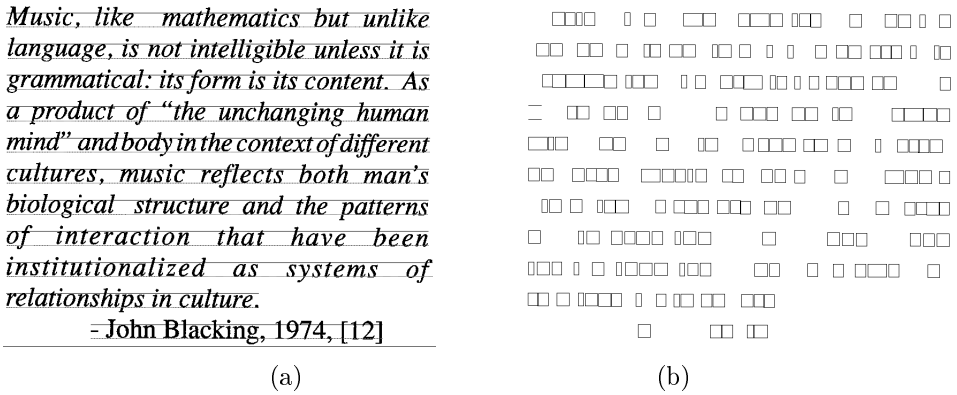
Figure 6.13: Reconnaissance de blocs graphiques (Doc. 8).

6.4.2 Estimation à partir des signes

Soient $L = \{c_k\}$ une ligne de texte constituée par un ensemble de signes et $\overline{L} \subset L$ défini comme suit :

$$\overline{L} = \{c_k \in L | h(c_k) = h_f\}$$

où h_f désigne la hauteur estimée des lettres minuscules. L'estimation de la ligne de base de l est réalisée en calculant la moyenne de la coordonnée y_2 de l'enveloppe des éléments de $\overline{L}$. La figure 6.14.a montre l'ensemble des signes ayant servi au calcul des lignes de base illustrées à la figure 6.14.b.



(a) (b)

Figure 6.14: Estimation des lignes de base à partir de composantes connexes.

L'estimation des lignes de base supérieures est réalisée de la même manière sur $\overline{L}$, mais en

prenant la moyenne de la coordonnée y_1 des enveloppes. Cette technique, comparée à la précédente est non seulement fiable, mais en plus très efficace.

6.5 Ordre de parcours dans les structures physiques

6.5.1 Objectif

Les notions d'ordre et de hiérarchie au sens large sont fondamentales en mathématiques et en informatique. Elle sont sous-jacentes à tout ce qui concerne les questions de structuration de données, de tri et de recherche. Notre système de reconnaissance de structures physiques repose à la fois sur une modélisation des documents et sur celle de l'organisation de leurs constituants. Par exemple, dans une stratégie d'analyse ascendante, il est important que les entités physiques soient disposées dans un ordre adéquat pour leur regroupement dans une entité englobante. En effet, une recherche systématique du plus proche voisin aurait pour conséquence de diminuer considérablement l'efficacité de la technique de regroupement; d'où l'intérêt de cette section.

6.5.2 Ordres de parcours

Soit $C_y(e_1, e_2)$ la fonction définie dans la section 5.1.2 par rapport à l'axe Y et qui retourne le type de relation locale entre les deux entités physiques e_1 et e_2 (cf. figure 5.1). Soit $C_x(e_1, e_2)$ la fonction analogue à $C_y(e_1, e_2)$, mais définie par rapport l'axe X.

Ordre de parcours des signes

L'ordre de parcours des signes, dans un bloc textuel, est obtenu au moyen de deux tris successifs :

1. suivant le critère R_1 défini par rapport à la coordonnée x_1 des enveloppes :

$$R_1(e_1, e_2) = x_1(e_1) \leq x_1(e_2)$$

2. suivant le critère R_2 défini par rapport au profil vertical des enveloppes :

$$R_2(e_1, e_2) = P_y(e_1, e_2) \leq 1$$

où la fonction P_y, définie dans la section 5.1.2, sert à comparer le profil vertical des enveloppes d'entités physiques.

La figure 6.15 illustre sur un exemple l'ordre de parcours des signes composant un bloc textuel suivant les deux critères d'ordre énumérés ci-dessus.

. . . to my mind, any community of musicological practice which excludes from consideration living musicians and restricts itself to accounts of frozen results of musical action, fails to be an inspiring community of inquiry about music.

- Otto Laske, 1972, [24]

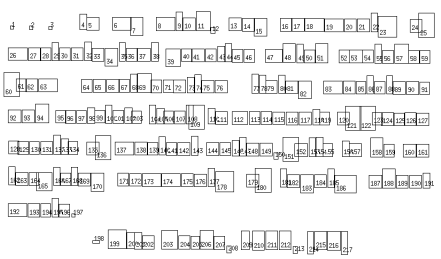


Figure 6.15: Ordre de parcours des signes dans un bloc textuel.

Ordre de parcours des blocs

L'ordre de parcours des blocs, dans une page spécifique, est obtenu au moyen de deux tris successifs :

1. suivant le critère R_1 défini par rapport à la coordonnée y_1 des enveloppes :

$$R_1(e_1, e_2) = y_1(e_1) \leq y_1(e_2)$$

2. suivant le critère R_2 défini par rapport au profil horizontal des enveloppes :

$$R_2(e_1, e_2) = P_x(e_1, e_2) \leq 1$$

où la fonction P_x, définie dans la section 5.1.2, sert à comparer le profil horizontal des enveloppes d'entités physiques.

La figure 6.16 illustre sur un exemple l'ordre de parcours des blocs composant une page de revue scientifique suivant les deux critères d'ordre énumérés ci-dessus.

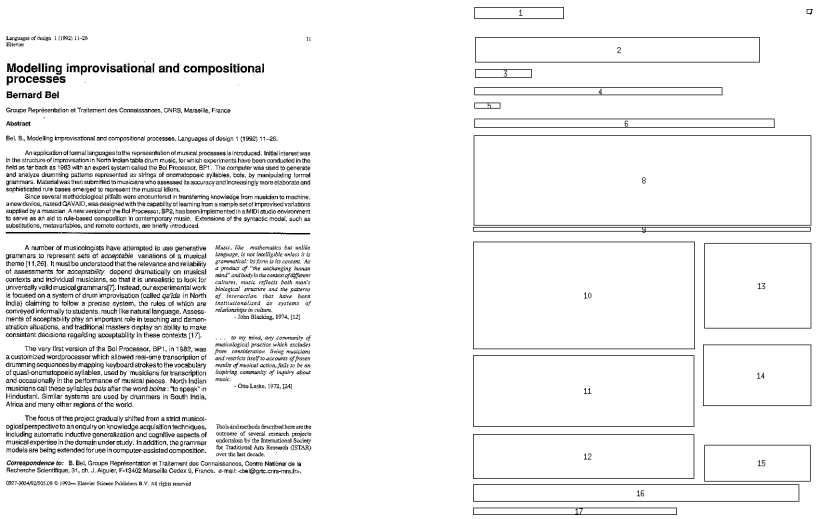


Figure 6.16: Ordre de parcours des blocs dans une page.

Observations

Si dans ce qui précède, la relation spatiale R_1 vérifie toutes les conditions d'une relation d'ordre totale, il n'en est pas de même pour R_2 qui ne vérifie même pas toujours les conditions d'une relation d'ordre partielle, condition *sine qua non* pour un tri. Pour preuve, considérons l'image de la figure 6.17 composée de trois entités physiques e_1, e_2 et e_3. On peut montrer que R_2 ne vérifie pas toujours la relation de transitivité. Toutefois, comme nous venons de le montrer dans les deux exemples de tri ci-dessus, l'ordre de parcours désiré pour notre analyse peut être obtenu après deux tris successifs.

Conclusion

Dans ce chapitre, nous avons traité le problème de la reconnaissance des microstructures en commençant par la classification des composantes. Le résultat de cette reconnaissance, guidée par

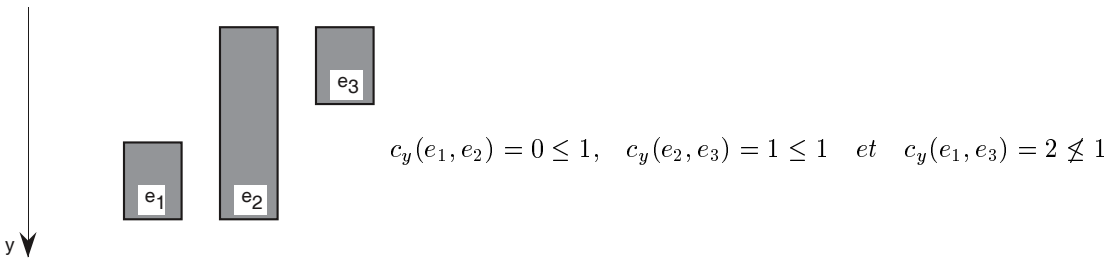


Figure 6.17: La comparaison de profils n'est pas toujours une relation d'ordre.

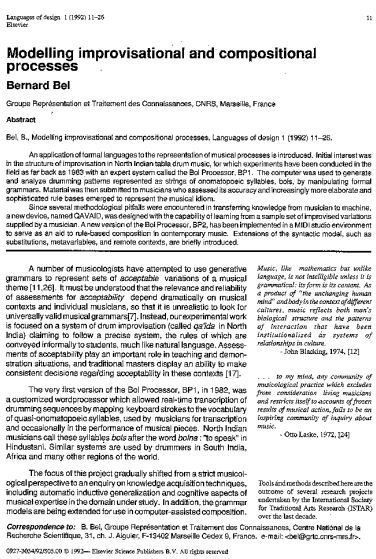
l'ensemble des règles gouvernant l'aspect graphique usuel des microstructures, est illustré sur des exemples de pages types. La qualité des résultats obtenus est due non seulement à l'approche de reconnaissance mais aussi et surtout à la fiabilité des seuils meétriques de segmentation utilisés.

Dans le chapitre 7, nous développons une méthode générale pour l'estimation automatique des seuils métriques ayant servi aussi bien dans les techniques de découpe que dans les techniques de fusion décrites dans ce chapitre.

Chapitre 7

Estimation des seuils métriques

Dans ce chapitre, nous abordons de front un problème souvent passé sous silence dans les systèmes de reconnaissance de documents : celui de la sélection automatisée des seuils métriques nécessaires à une bonne partition des documents segmentés. Notre objectif est d'estimer les seuils métriques servant aussi bien à la découpe hiérarchique dans une stratégie d'analyse descendante, qu'à la fusion hiérarchique dans une stratégie d'analyse ascendante. L'estimation est réalisée par apprentissage automatique au cours du traitement des documents au moyen d'une analyse statistique des espaces inoccupés dans les documents.



Languages of design 1 (1992) 11–26
Elsevier

11

Modelling improvisational and compositional processes

Bernard Bel

Groupe Représentation et Traitement des Connaissances, CNRS, Marseille, France

Abstract

Bel, B., Modelling improvisational and compositional processes, Languages of design 1 (1992) 11–26.

An application of formal languages to the representation of musical processes is introduced. Initial interest was in the structure of improvisation in North Indian tabla drum music, for which experiments have been conducted in the field as far back as 1985 with an expert system called the Bol Processor, BP1. The computer was used to generate and analyze drumming patterns represented as strings of onomatopoeic syllables, bols, by manipulating formal grammars. Material was then submitted to musicians who assessed its accuracy and increasingly more elaborate and sophisticated rule bases emerged to represent the musical idioms.

Since several methodological pitfalls were encountered in transferring knowledge from musician to machine, a new device, named QAVAID, was designed with the capability of learning from a sample set of improvised variations supplied by a musician. A new version of the Bol Processor, BP2, has been implemented in a MIDI studio environment to serve as an aid to rule-based composition in contemporary music. Extensions of the syntactic model, such as substitutions, metavariables, and remote contexts, are briefly introduced.

A number of musicologists have attempted to use generative grammars to represent sets of *acceptable* variations of a musical theme [11,26]. It must be understood that the relevance and reliability of assessments for *acceptability* depend dramatically on musical contexts and individual musicians, so that it is unrealistic to look for universally valid musical grammars[7]. Instead, our experimental work is focused on a system of drum improvisation (called *qa'ida* in North India) claiming to follow a precise system, the rules of which are conveyed informally to students, much like natural language. Assessments of acceptability play an important role in teaching and demonstration situations, and traditional masters display an ability to make consistent decisions regarding acceptability in these contexts [17].

Music, like mathematics but unlike language, is not intelligible unless it is grammatical: its form is its content. As a product of "the unchanging human mind" and body-in-the-context of different cultures, music reflects both man's biological structure and the patterns of interaction that have been institutionalized as systems of relationships in culture.
- John Blacking, 1974, [12]

. . . to my mind, any community of musicological practice which excludes from consideration living musicians and reinvests itself in accounts of frozen results of musical action, fails to be an inspiring community of inquiry about music.
- Otto Laske, 1972, [24]

The very first version of the Bol Processor, BP1, in 1982, was a customized wordprocessor which allowed real-time transcription of drumming sequences by mapping keyboard strokes to the vocabulary of quasi-onomatopoeic syllables, used by musicians for transcription and occasionally in the performance of musical pieces. North Indian musicians call these syllables *bols* after the word *bolna* : "to speak" in Hindustani. Similar systems are used by drummers in South India, Africa and many other regions of the world.

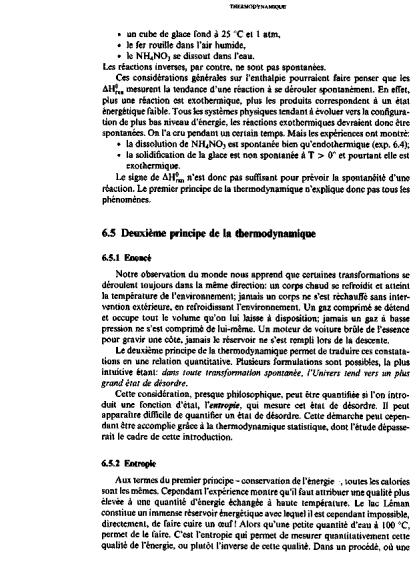
The focus of this project gradually shifted from a strict musicological perspective to an enquiry on knowledge acquisition techniques, including automatic inductive generalization and cognitive aspects of musical expertise in the domain under study. In addition, the grammar models are being extended for use in computer-assisted composition.

Tools and methods described here are the outcome of several research projects undertaken by the International Society for Traditional Arts Research (ISTAR) over the last decade.

Correspondence to: B. Bel, Groupe Représentation et Traitement des Connaissances, Centre National de la Recherche Scientifique, 31, ch. J. Aiguier, F-13402 Marseille Cedex 9, France. e-mail: <bel@grtc.cnrs-mrs.fr>.

0927-3604/92/$05.00 © 1992— Elsevier Science Publishers B.V. All rights reserved

(Doc. 1)



THERMODYNAMIQUE

- un cube de glace fond à 25 °C et 1 atm,
- le fer rouille dans l'air humide,
- le NH_4NO_3 se dissout dans l'eau.

Les réactions inverses, par contre, ne sont pas spontanées.

Ces considérations générales sur l'enthalpie pourraient faire penser que les ΔH^0_m mesurent la tendance d'une réaction à se dérouler spontanément. En effet, plus une réaction est exothermique, plus les produits correspondent à un état énergétique faible. Tous les systèmes physiques tendant à évoluer vers la configuration de plus bas niveau d'énergie, les réactions exothermiques devraient donc être spontanées. On l'a cru pendant un certain temps. Mais les expériences ont montré:

- la dissolution de NH_4NO_3 est spontanée bien qu'endothermique (exp. 6.4);
- la solidification de la glace est non spontanée à T > 0° et pourtant elle est exothermique.

Le signe de ΔH^0_m n'est donc pas suffisant pour prévoir la spontanéité d'une réaction. Le premier principe de la thermodynamique n'explique donc pas tous les phénomènes.

6.5 Deuxième principe de la thermodynamique

6.5.1 Enoncé

Notre observation du monde nous apprend que certaines transformations se déroulent toujours dans la même direction: un corps chaud se refroidit et atteint la température de l'environnement; jamais un corps ne s'est réchauffé sans intervention extérieure, en refroidissant l'environnement. Un gaz comprimé se détend et occupe tout le volume qu'on lui laisse à disposition; jamais un gaz à basse pression ne s'est comprimé de lui-même. Un moteur de voiture brûle de l'essence pour gravir une côte, jamais le réservoir ne s'est rempli lors de la descente.

Le deuxième principe de la thermodynamique permet de traduire ces constatations en une relation quantitative. Plusieurs formulations sont possibles, la plus intuitive étant: *dans toute transformation spontanée, l'Univers tend vers un plus grand état de désordre.*

Cette considération, presque philosophique, peut être quantifiée si l'on introduit une fonction d'état, l'***entropie***, qui mesure cet état de désordre. Il peut apparaître difficile de quantifier un état de désordre. Cette démarche peut cependant être accomplie grâce à la thermodynamique statistique, dont l'étude dépasserait le cadre de cette introduction.

6.5.2 Entropie

Aux termes du premier principe - conservation de l'énergie -, toutes les calories sont les mêmes. Cependant l'expérience montre qu'il faut attribuer une qualité plus élevée à une quantité d'énergie échangée à haute température. Le lac Léman constitue un immense réservoir énergétique avec lequel il est cependant impossible, directement, de faire cuire un œuf ! Alors qu'une petite quantité d'eau à 100 °C, permet de le faire. C'est l'entropie qui permet de mesurer quantitativement cette qualité de l'énergie, ou plutôt l'inverse de cette qualité. Dans un procédé, où une

119

(Doc. 9)

Figure 7.1: Documents servant à illustrer l'estimation des seuils métriques.

Tout au long de ce chapitre, l'estimation des seuils métriques est illustrée sur les deux exemples de documents spécifiques de la figure 7.1, l'un en anglais et l'autre en français. Ces deux documents sont caractérisés par un contenu purement textuel, justifié par le fait que les seuils recherchés ont été essentiellement définis, dans la section 5.1.3, par rapport aux blocs textuels. Il s'agit de :

- d_y^d : seuil maximal des espaces verticaux entre signes diacritiques et leur corps,

- d_x^c : seuil maximal des espaces horizontaux entre caractères d'un même mot,
- d_x^m : seuil maximal des espaces horizontaux entre mots d'une même ligne,
- d_y^l : seuil maximal des espace verticaux entre lignes d'un même bloc textuel,
- d_y^b : espace vertical séparant deux lignes de base consécutives (l'*interligne*).

Dans la section 7.1, nous présentons une fonction d'estimation paramétrique pour l'évaluation de ces seuils métriques. Les sections 7.2 à 7.5 présentent, pour chaque seuil métrique, les paramètres effectifs conduisant à une bonne évaluation.

7.1 Un estimateur paramétrique des seuils métriques

7.1.1 Fondement

Suite à une étude empirique de la distribution des espaces, nous avons pu établir une fonction d'estimation générale valable pour tous les seuils métriques utilisés dans les diverses étapes de segmentation.

Considérons par exemple l'ensemble des lettres obtenues après la classification des composantes connexes $C = \{c_k\}$ extraites des documents illustrés à la figure 7.1. Les histogrammes de la figure 7.2, illustrant la distribution des espaces entre les lettres de C, sont assez représentifs de l'allure générale de ce genre de distribution :

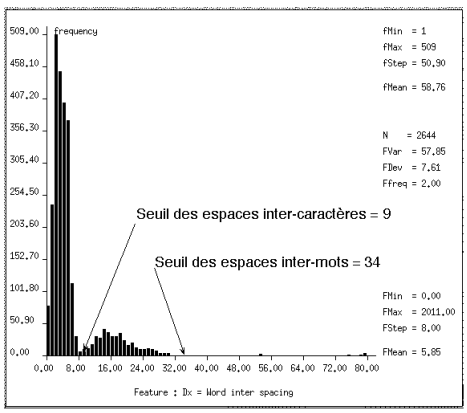
- pour les plus petits espaces, on observe une forte concentration incluant le maximum global; elle correspond à la plage de fréquences des espaces inter-caractères dans un mot;
- a la suite de cette première plage, on en trouve une autre, moins importante et un peu plus étalée que la première; elle correspond à la plage de fréquences des espaces inter-mots dans une ligne de texte;
- le reste de la distribution (plus ou moins plate) est inintéressante pour notre analyse; elle correspond entre autre, à la fréquence des espaces séparant des blocs.

On observe dans cette distribution que le seuil maximal des espaces inter-caractères d_x^c est situé entre les deux premières plages de fréquences indiquées ci-dessus alors que celui des espaces inter-mots d_x^m est situé à l'extrémité de la deuxième plage (cf. figure 7.2). Pour l'estimation des seuils, nous avons défini une fonction d'estimation Σ paramétrique qui, intuitivement, sert à localiser un minimum local dans la distribution des espaces.

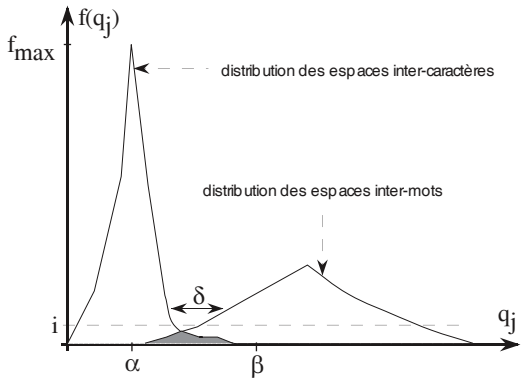
7.1.2 L'estimateur Σ

Soient X une variable aléatoire discrète relative aux espaces inoccupés dans un document, $Q = \{q_i\}$ une distribution de X dans un document spécifique p et f une fonction de fréquence définie sur Q. Nous définissons par :

- $f(q_i)$: fréquence de l'espace q_i dans le document p,
- f_{min} : fréquence minimale,
- f_{max} : fréquence maximale,
- q_f : espace de fréquence maximale,
- f_0 : fréquence minimale des espaces significatifs,
- $[\alpha, \beta]$: la plage d'intérêt pour la recherche du seuil s,



(Doc. 1)



(Doc. 9)

Figure 7.2: Distribution des espaces inter-caractères.

- l_s : longueur de l'intervalle de variation (tolérance) du seuil s.

Le principe général utilisé pour l'estimation d'un seuil s consiste à trouver, dans l'intervalle $[\alpha, \beta]$, le premier minimum local $\overline{q}$ tel que :

- $f(\overline{q}) < f_0$,
- $l_s \geq \delta$: où δ est une longueur minimale donnée.

Nous déterminons le seuil s par la formule :

$$s = \overline{q} + \epsilon \quad \text{où } \epsilon \text{ est un seuil de tolérance}$$

Lorsqu'il n'est pas possible d'obtenir un tel minimum local, on itère la recherche en incrémentant f_0, tous les autres paramètres restant inchangés. La fréquence minimale des espaces significatifs f_0 est initialisée à h_{min}. Cette évaluation des seuils est concrètement réalisée au moyen d'un estimateur Σ défini sur des distributions d'espaces Q et paramétré par α, β, δ et ϵ. Cet estimateur est formellement défini par la fonction itérative :

$$\Sigma_j(Q_i) = \begin{cases} q_j + \epsilon & \text{si } (q_{j+1} \leq \beta) \wedge (q_{j+1} - q_j \geq \delta) \\ \Sigma_{j+1}(Q_i) & \text{si } q_{j+1} \leq \beta \\ \Sigma_{j_0}(Q_{i+1}) & \text{si } i < f_{max} \\ \alpha & \text{sinon} \end{cases}$$

dans laquelle j_0, i_0 et Q_{f_0} sont définis comme suit :

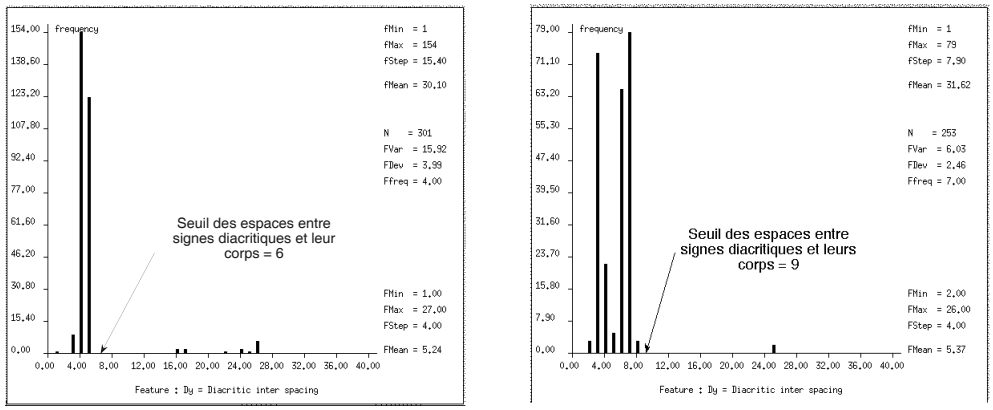
$$\begin{array}{lcl} j_0 & = & \alpha \\ i_0 & = & f_0 \\ Q_i & = & \{q_j\} = \{q_k \in Q \mid f(q_k) \geq i\} \end{array}$$

Le choix d'une plage d'intérêt pour la recherche des seuils est motivé par le fait que les espaces qui nous intéressent sont en général proportionnels au contenu des documents. Par exemple, nous avons pu observer que le seuil des espaces inter-caractères est en général inférieur à la moitié de la largeur la plus fréquente l_f des CCX.

Dans les sections 7.2 à 7.5, nous présentons, pour chaque seuil métrique, la variable aléatoire discrète X d'une part et, les paramètres α, β, δ et ϵ de Σ conduisant à une bonne estimation du seuil s recherché d'autre part.

7.2 Seuil des espaces entre signes diacritiques et lettres

Il s'agit ici de l'estimation du seuil d_y^d utilisé dans notre méthode de fusion hiérarchique pour regrouper les composantes connexes en signes. La méthode de découpe hiérarchique n'est pas concernée par ce seuil puisqu'elle s'arrête au niveau des mots, faute de pouvoir toujours extraire les rectangles structurants entre les caractères d'un même mot.



(Doc. 1) (Doc. 9)

Figure 7.3: Distribution des espaces entre les signes diacritiques et les lettres.

L'estimation du seuil d_y^d est réalisée au moyen de l'estimateur Σ, à partir de la distribution Q des espaces verticaux entre les signes diacritiques et leur corps (ex. espace vertical entre "." et "ı" dans "i"). Dans l'étude de telles distributions (cf. figure 7.3) nous avons observé que, quelle que soit la fonte utilisée, le seuil d_y^d des espaces entre un signe diacritique et son corps est :

- situé au delà de l'espace le plus fréquent q_f : ainsi $\alpha = q_f$,
- inférieur à la hauteur la plus fréquente h_f des CCX : ainsi $\beta = h_f$.

Généralement pour une fonte donnée, la valeur théorique de d_y^d est constante ce qui, normalement, implique une variation nulle. Toutefois, pour tenir compte d'éventuels bruits et de variations de fontes rencontrés dans la pratique, nous avons défini de façon empirique la longueur δ de l'intervalle de variation de d_y^d et le seuil ϵ de tolérance, en fonction de la résolution res des documents traités, comme suit :

- $\delta = res/2$,
- $\epsilon = 1$.

7.3 Seuil des espaces inter-caractères

Il s'agit ici de l'estimation du seuil d_x^c utilisé dans notre méthode de fusion hiérarchique pour regrouper les signes en mots. Ce même seuil est utilisé dans notre approche de découpe hiérarchique pour segmenter les lignes de texte en mots. Son estimation est réalisée, au moyen de l'estimateur Σ, à partir de la distribution Q des espaces inter-lettres profitant ainsi de la classification des CCX. Dans l'étude de telles distributions (cf. figure 7.2) nous avons observé que, quelle que soit la fonte utilisée, le seuil d_x^c des espaces inter-caractères est :

- situé après l'espace le plus fréquent q_f; ainsi $\alpha = q_f$,

- inférieur à la moitié de la largeur la plus fréquente l_f des CCX; ainsi $\beta = l_f/2$.

En typographie, l'espace entre les caractères d'un même mot est régi avant tout par le *crénage* qui varie en fonction des fontes. Par rapport aux espaces inter-mots, ces espaces varient très peu. Pour minimiser les risques de sous segmentation, par exemple la fusion de plusieurs mots en un, nous avons défini de façon empirique δ, longueur de l'intervalle de variation de d_x^c, et ϵ son seuil de tolérance, en fonction de la résolution res des documents traités :

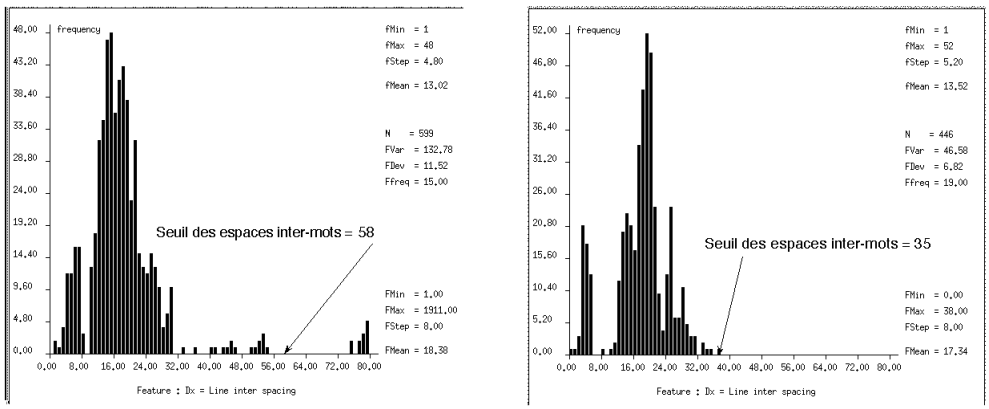
1. $\delta = 1$,
2. $\epsilon = res/2$.

La valeur de β, que nous avons estimée inférieure à la moitié de la largeur la plus fréquente des caractères contenus dans les documents traités, s'explique par le fait que les espaces inter-caractères sont assez petits (nuls, voir négatifs) quelle que soit la fonte utilisée. Quant à la valeur de δ, elle est justifiée par la faible différence qu'il y a entre le seuil supérieur des espaces inter-caractères et la seuil inférieur des espaces inter-mots. La valeur de ϵ permet de tenir compte de la résolution des documents traités.

Lorsqu'il n'est pas possible de bénéficier d'une classification des composantes connexes, typiquement dans une approche procédant par découpe, l'estimation de d_x^c est réalisée à partir de la distribution des espaces inter-CCX. Cette dernière estimation est moins fiable que celle réalisée à partir des espaces inter-caractères.

7.4 Seuil des espaces inter-mots

Il s'agit ici de l'estimation du seuil d_x^m utilisé dans notre méthode de fusion hiérarchique pour regrouper les mots en lignes de texte. Ce même seuil est utilisé dans notre approche de découpe hiérarchique pour segmenter les blocs textuels en lignes.



(Doc. 1) (Doc. 9)

Figure 7.4: Distribution des espaces inter-mots.

L'estimation du seuil d_x^m est réalisée au moyen de l'estimateur Σ, à partir de la distribution Q des espaces inter-mots dans une approche d'analyse ascendante. Dans l'étude de telles distributions (cf. figure 7.4) nous avons observé que, quelle que soit la fonte utilisée, le seuil d_x^m des espaces inter-mots est :

- situé après l'espace le plus fréquent q_f; ainsi $\alpha = q_f$,

- inférieur à 3 fois la largeur la plus fréquente l_f des CCX; ainsi $\beta = 3 \times l_f$.

Nous avons observé que, quel que soit le mode de justification utilisé dans les blocs textuels, la frontière entre les espaces inter-mots et les espaces inter-colonnes est relativement importante. Par conséquent, nous avons défini de façon empirique δ, longueur de l'intervalle de variation de d_x^m, et ϵ son seuil de tolérance, dans le but d'adapter l'estimation de d_x^m à la résolution res des documents traités :

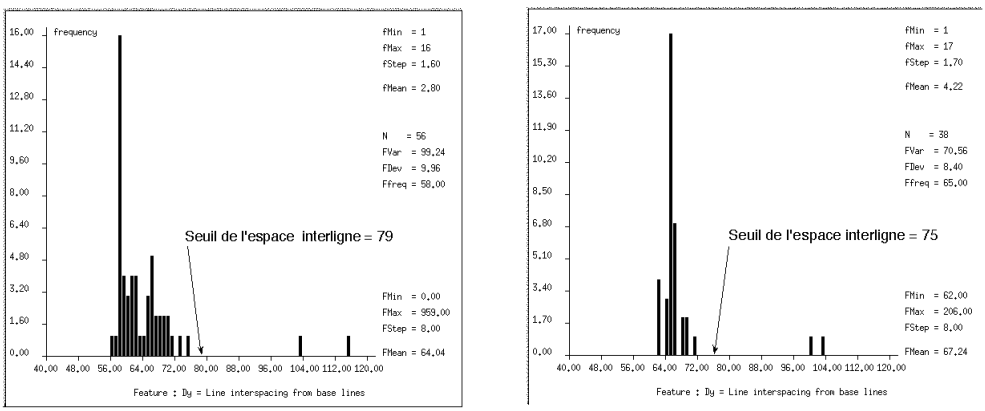
1. $\delta = res + 1$,
2. $\epsilon = res$.

Lorsqu'on ne dispose pas encore de la segmentation en mots, typiquement dans une approche procédant par découpe, l'estimation de d_x^m est réalisée à partir de la distribution des espaces inter-CCX. Cette dernière estimation est moins fiable que celle réalisée à partir des espaces inter-mots. De notre expérimentation, il ressort que la distribution de la largeur des rectangles structurants verticaux, déjà peu fiable pour l'estimation du seuil d_x^c, est encore moins appropriée pour l'estimation de d_x^m.

7.5 Seuil des espaces inter-lignes

7.5.1 L'interligne

Si en typographie, *interligne* désigne l'espace entre deux lignes de texte consécutives, en informatique, il désigne généralement l'espace entre deux lignes de base consécutives. Cette dernière définition de l'interligne présente une propriété intéressante : celle d'être constante dans un document monofonte. L'interligne, ainsi défini, est estimé par le seuil d_y^b utilisé dans notre méthode de fusion hiérarchique pour regrouper les lignes en blocs textuels.



(Doc. 1) (Doc. 9)

Figure 7.5: Distribution des espaces entre lignes de base consécutives.

L'estimation du seuil d_y^b est réalisée au moyen de l'estimateur Σ, à partir de la distribution Q des espaces entre lignes de base consécutives dans une approche d'analyse ascendante. Dans l'étude de telles distributions (cf. figure 7.5) nous avons noté que, quelle que soit la fonte utilisée, le seuil d_y^b de l'interligne est :

- situé après l'espace le plus fréquent q_f; ainsi $\alpha = q_f$,

- inférieur à 4 fois la hauteur la plus fréquente h_f des CCX; ainsi $\beta = 4 \times h_f$.

Pour tenir compte de la variation des fontes et de la résolution res des documents traités, nous avons défini de façon empirique δ, longueur de l'intervalle de variation de d_y^b, et ϵ son seuil de tolérance :

1. $\delta = 2 \times res$,
2. $\epsilon = res$.

7.5.2 Espace inter-lignes effectif

Malheureusement, il n'est pas toujours possible de profiter de l'avantage que procure l'inter-ligne dans une stratégie d'analyse descendante, notamment dans notre méthode de découpe hiérarchique. Notre solution palliative a été d'estimer le seuil maximal des espaces effectifs séparant deux lignes de texte consécutives : il s'agit du seuil métrique d_y^l dont l'estimation est réalisée à partir de la distribution Q de la hauteur des rectangles structurants horizontaux.

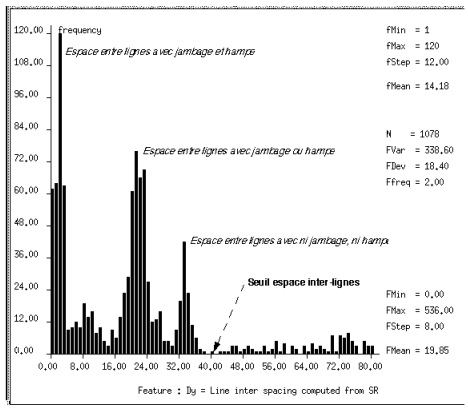


Figure 7.6: Distribution de la largeur des rectangles structurants horizontaux (Doc. 1).

Dans de telles distributions (cf. la figure 7.6) on observe trois pics :

1. distribution des espaces inter-lignes impliquant jambages et hampes,
2. distribution des espaces inter-lignes n'impliquant exclusivement que des jambages ou que des hampes,
3. distribution des espaces inter-lignes n'impliquant ni jambages, ni hampes.

De notre étude il ressort que, quelle que soit la fonte en vigueur dans le document traité, le seuil d_y^l des espaces inter-lignes est :

- situé après l'espace le plus fréquent q_f; ainsi $\alpha = q_f$,
- inférieur à 2 fois la hauteur la plus fréquente h_f des CCX; ainsi $\beta = 2 \times h_f$.

Pour tenir compte de la variation, d'une part, des espaces inter-lignes comme illustrée dans la figure 7.6 et, d'autre part, de la résolution res des documents traités, nous avons défini de façon empirique δ, longueur de l'intervalle de variation de d_y^b, et ϵ son seuil de tolérance :

1. $\delta = 2 \times res$,
2. $\epsilon = (q_{j+1} - q_j)/2 + res$.

Conclusion

Pour juger de l'efficacité de ces estimations, le lecteur peut se rapporter aux résultats de segmentation présentés, dans le chapitre 6 et dans l'annexe B illustrant d'autres résultats de reconnaissance. De tous les paramètres de l'estimateur générique Σ, le plus sensible reste δ; c'est avant tout de ce dernier que dépend la fiabilité des valeurs estimées. Par exemple, pour un δ trop petit, le seuil estimé est en général plus petit que celui attendu. C'est le cas lorsque l'estimation échoue dans le premier minimum local de largeur δ.

En général, il n'a pas été pas nécessaire de réestimer les seuils métriques pour chaque page spécifique traitée puisque l'on a pu profiter des connaissances acquises lors des traitements antérieurs. Pour des documents composés de plusieurs pages, il n'est donc pas nécessaire d'extraire systématiquement les seuils sur chacune des pages. Notre approche consiste à les estimer par apprentissage automatique à partir de quelques pages typiques constituées, en général, deux ou trois pages. Nous considérons comme typique une page contenant une forte proportion de texte. Sur l'ensemble des documents traités, notre évaluation des seuils estimés (cf. section 10.3) prouve aussi bien la fiabilité que l'efficacité de l'estimateur Σ que nous avons défini dans ce chapitre.

Dans le chapitre 8, nous présentons le nouveau langage que nous avons défini pour décrire la macrostructure générique des documents composites. De telles descriptions, fondées sur une exploitation des espaces inoccupés, sert à guider la reconnaissance des macrostructures spécifiques puisque, contrairement aux microstructures, il n'existe aucune convention universelle régissant l'aspect graphique de ces dernières.

Chapitre 8

Langage de description de macrostructures physiques génériques

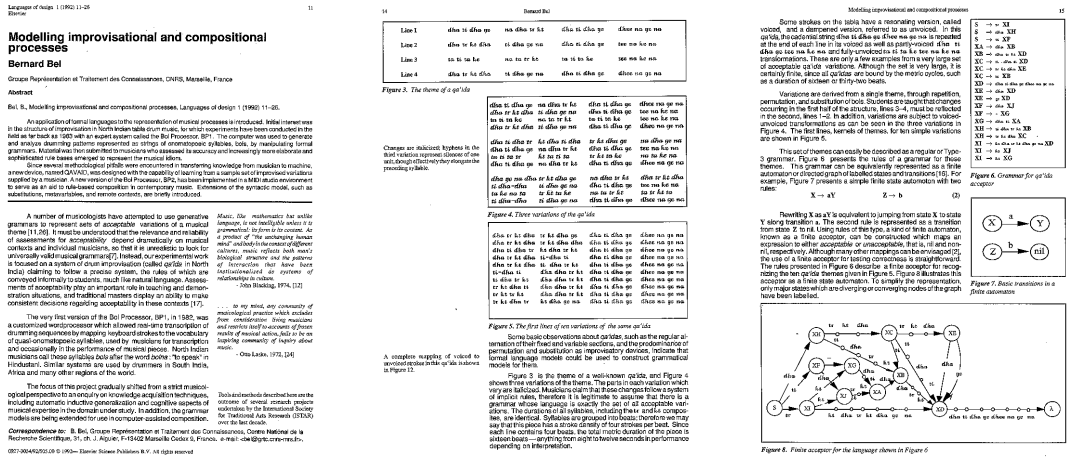
8.1 Motivation et originalité

Le formatage des microstructures, comme nous le montrons dans le chapitre 5, est régi par des conventions universelles, éventuellement dépendantes de la culture linguistique. Si nous avons pu nous baser sur la régularité de la distribution des espaces pour établir les règles régissant l'aspect graphique des microstructures, il n'en est pas de même pour ce qui concerne l'aspect graphique des macrostructures qui généralement diffère d'un document à l'autre. Toutefois, on observe que les documents d'une même publication présentent globalement une certaine uniformité dans leurs aspects graphiques. Pour ce faire, nous avons défini un nouveau langage servant à décrire partiellement la mocrostructure générique des documents [93]. Par opposition aux langages de description de page classiques [43, 49, 53, 55, 52]), ce nouveau langage tient son originalité du fait qu'il permet de décrire la structure globale des espaces inoccupés[1] dans un document. En effet, Pour la reconnaissance des macrostructures, nous pensons qu'il est plus facile d'analyser la structure globale des espaces inoccupés que les relations de voisinage entre les blocs.

Tout au long de ce chapitre, nous allons illustrer les principaux concepts du langage au moyen d'un exemple de classe de documents composites constituée des papiers scientifiques publiés dans la revue intitulée *"Language of Design Review"* que nous désignons par LDR. Les trois images de la figure 8.1 sont trois exemples de pages typiques, du point de vue de leur aspect graphique global, des articles publiés dans LDR : la première image (a) illustre la structure graphique des pages de garde, la seconde (b) celle des pages paires et la troisième (c) celle des pages impaires.

Le langage présenté dans ce chapitre se compose de deux parties, la première partie est dédiée à la description de la macrostructure générique des documents appartenant à une même classe alors que la seconde partie permet d'associer à un document spécifique la macrostructure générique (classe) qui régit son aspect graphique. Les sections 8.2 à 8.5 présentent les différents concepts du langage alors que la section 8.6 présente la manière de spécifier un document spécifique. Dans l'annexe A.1, nous donnons la grammaire complète du langage sous la forme de règles EBNF.

[1] L'espace inoccupé désigne ici ce qui reste du *fond*, en anglais *background*, d'un document après formatage



(a) (b) (c)

Figure 8.1: Pages typiques des articles publiés dans la revue *LRD*.

8.2 Description des classes de documents

A défaut d'une convention universelle régissant l'aspect graphique de la macrostructure de tous les documents composites, nous avons introduit, entre la structure physique générique des documents composites (voir figure 4.7) et la structure physique spécifique d'un document composite particulier, un niveau d'abstraction intermédiaire, regroupant dans une *classe* des documents qui globalement ont un même aspect graphique (voir figure 8.2). Notre objectif, ici, est de définir les règles qui serviront à décrire l'aspect graphique générique des documents appartenant à une même classe.

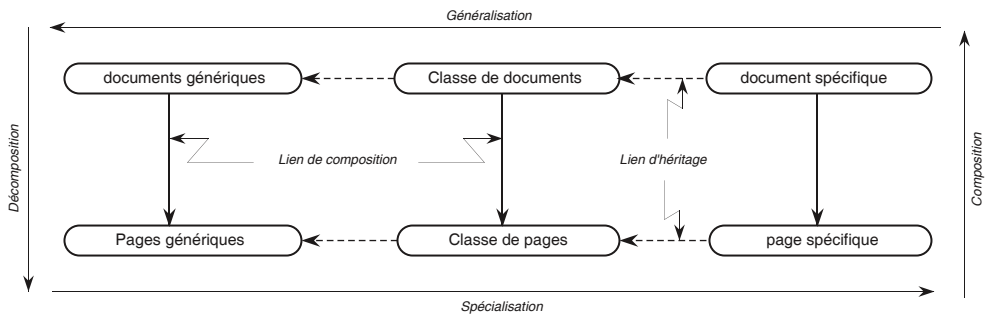
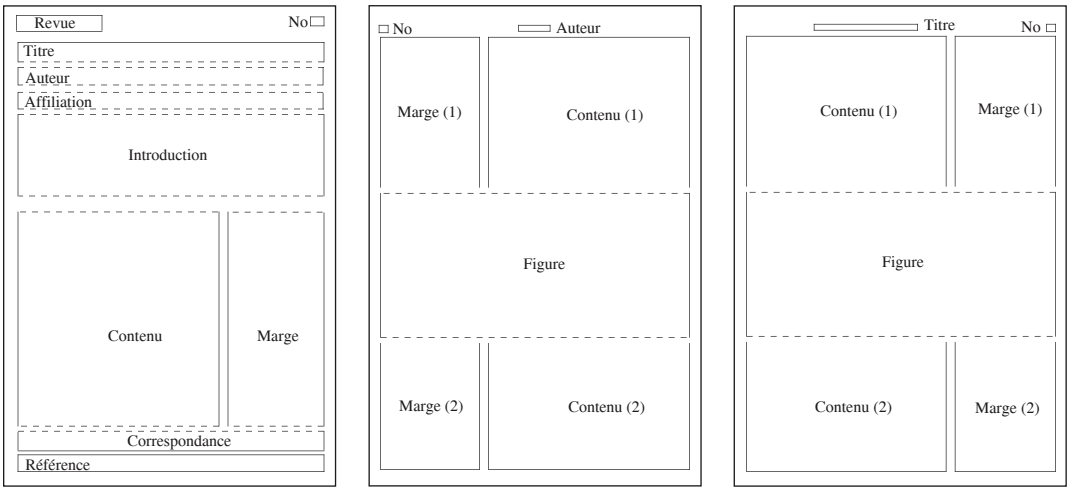


Figure 8.2: Raffinement des documents composites.

Un document spécifique est souvent composé de plusieurs pages dont certaines ont globalement un même aspect graphique. Nous regroupons de telles pages dans un concept de classe de pages désigné par PAGE. L'ensemble des classes de pages régissant une classe de documents est décrit au moyen d'une entité désignée par VOLUME. Par exemple, pour couvrir l'ensemble des articles publiés dans la revue LDR nous avons défini une classe de documents appelée LDR_articles et composée de trois classes de pages illutrer à la figure 8.3 : (a) traduit la macrostructure des pages de garde LDR_garde, (b) celle des pages paires LDR_paire et (c) celle des pages impaires LDR_impaire. La description de la classe LDR_articles est donnée, dans ce nouveau langage, par

la description 8.1.



a : LDR_garde b : LDR_paire c : LDR_impaire

Figure 8.3: Macrostructures génériques des pages typiques de la figure 8.1.

```
VOLUME LDR_articles IS
  UNIT = MM
  WIDTH = 160
  HEIGHT = 240
  LANGUAGE = ENGLISH
  PAGE LDR_garde IS    ...    END
  PAGE LDR_paire IS    ...    END
  PAGE LDR_impaire IS    ...    END
END
```

Description 8.1: Description partielle de la classe de documents LDR_articles.

La description d'une classe de documents comprend, en plus de la description de l'ensemble de ses pages typiques (cf. scetion 8.3), la spécification de quatre attributs de classe :

1. l'*unité de mesure* par défaut utilisée pour positionner les séparateurs,
2. la *hauteur* des pages spécifiques,
3. la *largeur* des pages spécifiques,
4. la *langue* de rédaction par défaut des documents spécifiques.

La connaissance de la langue de rédaction d'un document peut être prépondérante dans le choix d'algorithmes adéquats pour sa reconnaissance. A titre d'exemple, on observe dans des documents rédigés en français beaucoup de signes diacritiques contrairement à ceux qui sont rédigés en anglais et qui n'en comportent presqu'aucun (à l'exception par exemple des "." dans "i" ou "j" que nous traitons dans notre modèle comme des points diacritiques au même titre que "´" dans "é"). De tous les attributs définis dans une classe de documents, seule la langue de rédaction peut être redéfinie au moment d'associer un document spécifique à sa classe (cf. section 8.6). En effet, il peut arriver qu'un rapport technique, d'ordinaire rédigé en anglais, soit par exemple rédigé en français pour des raisons linguistiques ou politiques. Dans la description 8.1, LDR_garde, LDR_paire et LDR_impaire sont les entités décrivant les trois pages types de la classe LDR_articles.

8.3 Description des classes de pages

La pratique dans l'art de la mise en page consiste à créer, pour chaque classe de pages, une maquette servant à guider le formatage des pages spécifiques conformes à cette dernières. Une maquette définit une partition élémentaire des pages devant posséder globalement un même aspect graphique. Cette pratique favorise une utilisation cohérente et consistante des espaces, permettant ainsi d'accroître l'efficacité du processus de formatage. Nous désignons par *classe de pages* un ensemble de pages ayant un aspect graphique conforme à une même maquette. Lorsqu'on observe l'aspect graphique des pages spécifiques d'un document, on note que cette dernière est souvent perturbée par la présence d'illustrations (graphiques ou tableaux) ne faisant pas partie du contenu textuel principal. Pour tenir compte de ce genre de perturbation, une classe de pages est définie par une superposition de couches transparentes désignées par LAYER et dédiées à des contenus bien spécifiques.

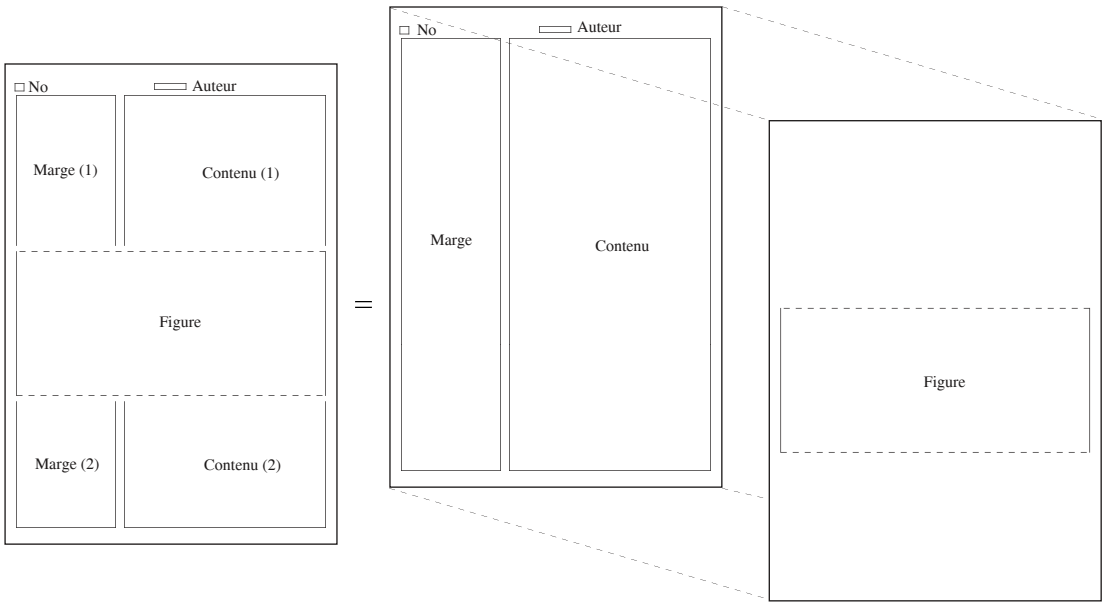


Figure 8.4: Décomposition en couches de la macrostructure 8.3.b.

Par exemple, la classe de pages LDR_paire (cf. figure 8.4.a) peut être représentée au moyen de deux couches, comme l'illustre la figure 8.4 dans laquelle la première couche (cf. figure 8.4.b) est dédiée au contenu textuel principal et la seconde (cf. figure 8.4.c) au reste du contenu. Cette description de la classe LDR_paire est partiellement traduite, dans ce nouveau langage, par la description 8.2 complétée dans les sections suivantes. Dans la description d'une classe de pages, il est possible de définir des classes de séparateurs génériques de régions (cf. section 8.4) aussi bien au niveau de la page que localement à une couche.

```
PAGE LDR_paire IS
  HSEP    ...    IS    ...
  LAYER Principale IS
    VSEP    ...    IS    ...
    REGION Main IS    ...
    REGION Margin IS    ...
    REGION No IS    ...
    REGION Author IS    ...
```

```
  END
  LAYER Secondaire IS
    HSEP    ...    IS    ...
    REGION Figure IS    ...
  END
END
```

Description 8.2: Description partielle de la classe de pages LDR_paire.

8.4 Description des séparateurs génériques

8.4.1 Séparateur

Dans cette section, il s'agit de définir l'entité séparateur générique servant à délimiter le contenu des pages. Comme nous le présentons dans la section 4.4, les séparateurs servent à structurer l'espace inoccupé dans une page de document. Un séparateur est défini par sa position relative, sa taille minimale et les deux séparateurs qui le délimitent de chaque côté. La position relative d'un séparateur statique est donnée par une valeur numérique alors que celle d'un séparateur flottant est donnée par un intervalle définissant son domaine de valeurs. Quatre séparateurs génériques ont été prédéfinis pour délimiter le contenu d'une page spécifique : TOP, BOTTOM, LEFT et RIGHT. Le langage permet de spécifier, au moyen d'un attribut, si le séparateur est espace ou un filet caractérisé par :

- sa nature pouvant être soit une *simple ligne*, soit une *double ligne* ou un *pointillé*;
- sa position relative aux deux séparateurs de délimitation et pouvant être soit centré, soit situé sur la gauche ou sur la droite.

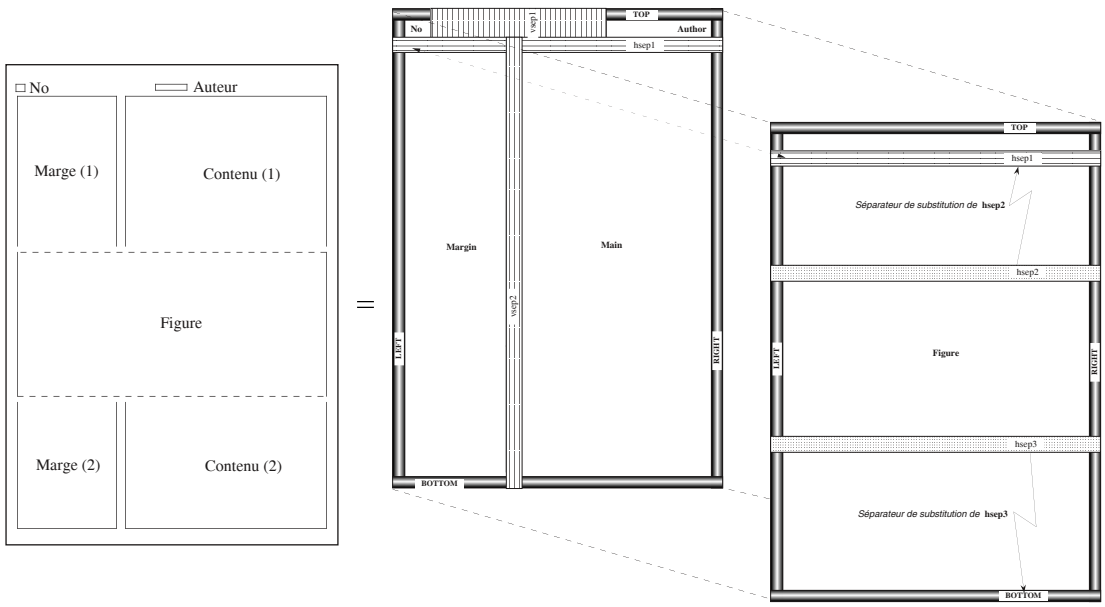
Dans la figure 8.7.b, *hsep*1 et *vsep*1 sont deux exemples de séparateurs statiques, *hsep*2 et *hsep*5 sont deux exemples de séparateurs flottants et *vsep*2 un exemple de séparateur élastique.
La figure 8.5 montre l'aspect graphique des deux couches de la classe de pages LDR_paire dans lequel les séparateurs génériques délimitant les régions ont été mis en évidence. Cette description est partiellement traduite, dans ce nouveau langage, par la description 8.3 qui complète la description de LDR_paire commencée avec la description 8.2.

```
PAGE LDR_paire IS
  HSEP hsep1 IS (4, 3, LEFT, RIGHT, SPACE)
  LAYER Principale IS
    VSEP vsep1 IS (40, 65, TOP, hsep1, SPACE)
    VSEP vsep2 IS (53, 4, hsep1, BOTTOM, SPACE)
    REGION Main IS    ...
    REGION Margin IS    ...
    REGION No IS    ...
    REGION Author IS    ...
  END
  LAYER Secondaire IS
    HSEP hsep2 IS (-[10, 220], 2, LEFT, RIGHT, SPACE) SUBSTITUTE hsep1
    HSEP hsep3 IS ([20, 240], 2, LEFT, RIGHT, SPACE) SUBSTITUTE BOTTOM
    REGION Figure IS    ...
  END
END
```

Description 8.3: Description des séparateurs génériques de la classe de pages LDR_paire.

8.4.2 Séparateur de substitution

Supposons P une classe de pages composée d'une seule couche qui, à son tour, est composée de deux régions r_1 et r_2 séparées par un séparateur flottant horizontal s (voir figure 8.6). Soit maintenant p une page spécifique conforme à P, mais composée uniquement d'une seule région r; de deux choses l'une :



a : LDR_paire b : Principale c : Superposée

Figure 8.5: Description de la macrostructure 8.4 au moyen des séparateurs génériques.

- soit s est substitué par le séparateur prédéfini BOTTOM et $r = r_1$,
- soit s est substitué par le séparateur prédéfini TOP et $r = r_2$.

Dans le but de lever de telles ambiguïtés, il peut être spécifié pour chaque séparateur flottant un séparateur de substitution; cela revient à définir une des deux régions r_1 et r_2 optionnelle. Par exemple, une région dédiée aux notes de bas de page sera définie, en général, optionnelle puisqu'elle peut être absente d'une page spécifique. Dans la description de LDR_paire donnée par la description 8.3, les séparateurs génériques *hsep*2 et *hsep*3 sont définis avec un séparateur de substitution, respectivement *hsep*1 et BOTTOM.

8.5 Description des régions génériques

Nous désignons par région, une portion de page rectangulaire dédiée à un contenu spécifique et dont la position est quasi invariable à travers les pages du document. Au lieu de décrire la macrostructure des pages directement par rapport à leurs régions comme dans les langages classiques de description de pages (voir figure 8.7.a), nous avons plutôt décrit celle-ci par rapport à l'espace inoccupé dans les pages. A cet effet, l'espace inoccupé est structuré en classes de séparateurs génériques pour délimiter les régions génériques (voir figure 8.7.b) d'une classe de pages, transformant ainsi la macrostructure des classes de pages en un réseau de séparateurs génériques.

```
PAGE LDR_paire IS
  HSEP hsep1 IS (4, 3, LEFT, RIGHT, SPACE)
  LAYER Principale IS
    VSEP vsep1 IS (40, 65, TOP, hsep1, SPACE)
    VSEP vsep2 IS (53, 4, hsep1, BOTTOM, SPACE)
    REGION Main IS (vsep2, RIGHT, hsep1, BOTTOM)
    REGION Margin IS (LEFT, vsep2, hsep1, BOTTOM, ANY, SMALL, 1)
```

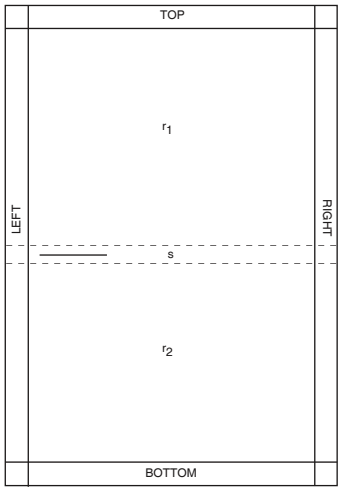


Figure 8.6: Schéma de substitution d'un séparateur flottant.

```
      REGION No IS (LEFT, vsep1, TOP, hsep1, TEXT, VSMALL)
      REGION Author IS (vsep1, RIGHT, TOP, hsep1, TEXT, VSMALL)
    END
    LAYER Secondaire IS
      HSEP hsep2 IS (-[10, 220], 2, LEFT, RIGHT, SPACE) SUBSTITUTE hsep1
      HSEP hsep3 IS ([20, 240], 2, LEFT, RIGHT, SPACE) SUBSTITUTE BOTTOM
      REGION Figure IS (LEFT, RIGHT, hsep2, hsep3, {TABLE, GRAPHIC}, SMALL)
    END
END
```

Description 8.4: Description des régions de la classe de pages LDR_paire.

Dans la description 8.4 complètant celle de la classe LDR_paire, la description d'une région générique peut être enrichie de trois attributs de classe optionnels, en rapport avec le contenu des régions spécifiques.

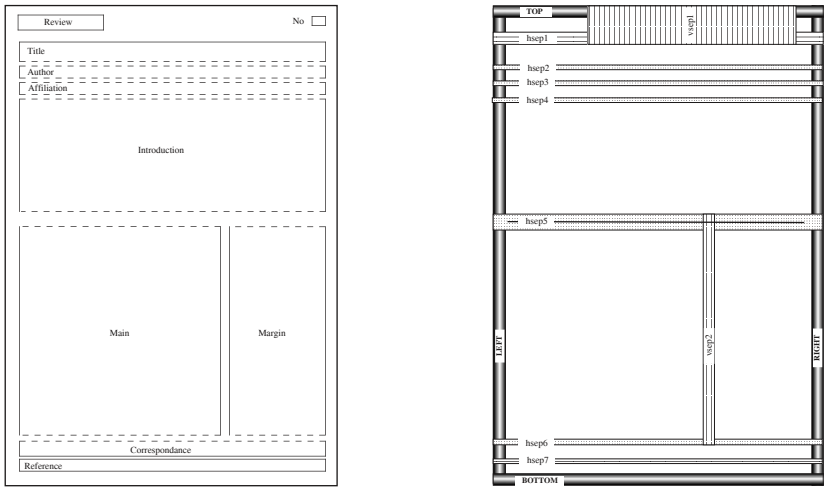
1. La nature des blocs autorisés dans les régions spécifiques; elle est donnée au moyen d'un sous-ensemble de types prédéfinis : bloc de texte, formule, graphique, tableau et photographie. Par exemple, dans la description 8.4, les régions spécifiques conformes à la région générique *Figure* ne peuvent être composées que de tableaux et de graphiques.

2. La taille dominante des caractères appartenant aux régions spécifiques; elle est donnée par rapport à la taille la plus fréquente des caractères contenus dans le document spécifique en cours de traitement. La taille est qualifiée soit de normale, soit de très large, soit de large, soit de petite, soit de très petite. Généralement, la taille des caractères contenus dans une région dédiée au contenu principal, sera qualifiée de normale. Par exemple, dans la description 8.4, les régions spécifiques conformes à la région générique *Figure* ont une taille qualifiée de petite.

3. La structure grossière des régions spécifiques qui peut être :

   - soit une structure de colonnes que l'on peut spécifier en donnant le nombre de colonnes;
   - soit une structure mosaïque désignant les régions que l'on ne peut modéliser au moyen d'un arbre XY pur.

   Par exemple, toutes les régions génériques définies dans la description 8.4 ont une structure de colonnes, en l'occurence une colonne.



a : Régions b : Séparateurs

Figure 8.7: Macrostructure de la classe de pages 8.3.a.

Par analogie avec les séparateurs, on dira d'une région r, définie par ses quatre délimiteurs $(left, right, top, bottom)$, qu'elle est statique si ses 4 séparateurs sont statiques, flottante si les séparateurs $left$ et $right$ ou top et $bottom$ sont flottants et pour finir élastique si un et un seul de ses 4 séparateurs est élastique. Dans la figure 8.7.a, *Review* est un exemple de région statique, *Introduction* un exemple de région flottante et, pour finir, *Title* un exemple de région élastique. Dans l'annexe A.3, nous donnons la description complète de la classe *LDR_articles* qui régit la macrostructure des articles scientifiques publiés dans la revue intitulée *Language of Design*.

8.6 Description de documents spécifiques

Nous avons présenté dans les sections 8.2 à 8.5 comment définir une description de classe de documents. Dans cette section, nous présentons la manière d'établir la correspondance entre un document spécifique et sa classe. Dans un certain nombre de documents, les pages sont structurées en sous-ensembles de pages hiérarchiques. Par exemple, une thèse peut être organisée en parties, une partie en chapitres qui, à leur tour, constituent des ensembles de pages. L'objectif principal de cette partie du langage est d'associer à chaque page du document une des classes de pages définies dans la classe du document spécifique. Nous illustrons ce mécanisme au moyen de la description 8.5 d'un d'article spécifique conforme à la classe LDR_articles. Cet article, intitulé *Modeling improvisational on compositional processes* et que nous désignons par MICP_article, est tiré du numéro Vol. 1 (1) 1-104 (1992) de la revue intitulée *Language of Design Review* (LDR).

```
VOLUME MICP_article IS
  DIRECTORY = Catalogue/MICP_article
  INSTANCE OF = LDR_articles
  LANGUAGE = ENGLISH
  SET article IS
    (page_1.400 , first)
    (page_2.400 , even, LAYER Superposée = NULL)
    (page_3.400 , odd )
    (page_4.400 , even)
    (page_5.400 , odd )
    (page_6.400 , even)
    (page_7.400 , odd )
    (page_8.400 , even)
```

```
      (page_9.400 , odd )
      (page_10.400, even)
      (page_11.400, odd )
      (page_12.400, even)
      (page_13.400, odd )
      (page_14.400, even)
      (page_15.400, odd )
      (page_16.400, even)
    END
  END
```

Description 8.5: Spécification d'un document spécifique conforme à la classe LDR_articles.

Au cours de la spécification d'un document particulier, il est possible de modifier la langue de rédaction par défaut spécifiée dans la classe du document. En effet, il peut arriver qu'un rapport technique, d'ordinaire rédigé en anglais, soit par exemple rédigé en français pour des raisons linguistiques ou politiques. Cette seconde partie du langage permet aussi :

- de situer sur le disque, l'emplacement des images de pages composant le document particulier,
- optionnellement, de notifier lors de l'association des classes de pages avec les pages spécifiques, l'abscence dans la page spécifique d'une des couches décrites dans sa classe.

Cette dernière possibilité est illustrée, dans la description 8.5, sur la page spécifique *page_2.400*. Dans l'annexe A.2, nous donnons, sous la forme de règles EBNF, la grammaire spécifiant la manière d'établir la correspondance entre un document spécifique et sa classe.

Conclusion

Dans ce chapitre, nous avons présenté un nouveau langage motivé par le constat qu'il n'existe aucune règle universelle qui régit la formatage des régions. Ce langage permet de décrire partiellement la macrostructure commune aux documents appartenant à une même classe. Il tient son originalité, par rapport aux langages classiques, du fait qu'au lieu de décrire un document directement par rapport à ses blocs, ce dernier est décrit au moyen d'un réseau de séparateurs servant à structurer les espaces inoccupés.

Lors d'une description, la donnée des attributs optionnels (génériques et spécifiques), peut être prépondérante dans le choix d'algorithmes de segmentation plus adéquats et donc plus performants. Le fait de considérer, au cours d'un traitement, toutes les pages composant un document offre l'avantage de pouvoir estimer, au plus près, les paramètres de segmentation puisqu'on peut tirer profit de la quantité appréciable d'échantillons disponibles.

Dans le chapitre 9, présentant la reconnaissance des macrostructures, la segmentation en régions (cf. section 9) est guidée par une description de la classe du document à traiter.

Chapitre 9

Reconnaissance des macrostructures

Dans ce chapitre, nous traitons le problème de la reconnaissance de la macrostructure physique des documents composites. Cette reconnaissance est guidée par une description de la macrostructure générique du document spécifique à traiter [93]. L'interprétation de la description, donnée dans le langage que nous avons défini à cet effet dans le chapitre 8, est présentée dans la section 9.1. Dans la section 9.2 nous dércivons notre méthode de segmentation en régions puis, dans la section 9.3, la poursuite de la segmentation des régions en blocs.

9.1 Analyseurs des descriptions de macrostructures

Pour chacune des deux parties que comporte le langage de description de macrostructures physiques, nous avons réalisé un analyseur : un analyseur de descriptions génériques et un analyseur de descriptions spécifiques.

9.1.1 Analyseur de descriptions génériques

L'interprétation de la description d'une classe de documents consiste à générer, pour chacune des pages génériques décrites dans la classe, une structure physique partiellement constructe et qui servira de point de départ pour la reconnaissance de ses pages spécifiques. Nous désignons par *maquette* une telle structure partielle dans laquelle la position exacte des séparateurs reste à définir.

9.1.2 Analyseur de descriptions spécifiques

L'interprétation de la spécification d'un document particulier consiste à dériver, de l'ensemble des maquettes de pages constituant sa classe, une macrostructure partielle du document particulier. Pour ce faire, nous initialisons la macrostructure de chaque page spécifique avec la maquette résultant de l'interprétation de la description de la page générique qui, dans la classe du document, régit la page en question. La structure partielle, ainsi dérivée pour chaque page spécifique, sera complétée par la détermination de la position exacte des séparateurs spécifiques qui délimitent les régions. Il ne s'agit pas là, d'une instantiation à proprement dit pour deux raisons au moins :

1. le regroupement hiérarchique des pages spécifiques (ex. les chapitres d'une partie de thèse) est donné uniquement dans la description du document spécifique et n'est donc pas connu de sa classe;

2. la description d'une macrostructure générique ne tient pas compte de l'ordre des pages dans les documents spécifiques; elle ne définit que, et dans un ordre quelconque, les différents types de pages que peuvent contenir un document spécifique.

En résumé, la segmentation en régions d'un document spécifique revient dans un premier temps, à générer une structure physique partielle à partir de l'interprétation de sa description générique puis, dans un second temps, à déterminer la position exacte des séparateurs délimitant les régions dans la structure partielle déterminée précédemment. La reconnaissance des macrostructures se poursuit avec la segmentation des régions en blocs en se servant des rectangles structurants.

9.2 Segmentation en régions

Tout au long de cette section, nous illustrons notre approche de segmentation en régions à l'aide d'un document composite que nous supposons composé d'une page et conforme à un modèle composé de plusieurs couches. Nous supposons également que l'on dispose d'un ensemble de blocs constituant la page, les blocs pouvant être :

- soit des microstructures (feuilles de macrostructures),
- soit des blocs de niveau hiérarchique intermédiaire dans les microstructures,
- soit des composantes connexes.

Cette dernière supposition constitue un avantage certain, puisque la segmentation en régions peut ainsi intervenir à tout moment sur une structure partiellement déterminée au cours d'une reconnaissance. La segmentation en régions consiste à déterminer la position exacte des séparateurs, délimitant les régions, en se basant sur une analyse des rectangles structurants.

9.2.1 Détermination des régions

La segmentation d'une page en régions commence par la segmentation de la couche dédiée à son contenu principal. Une page est représentée au moyen d'une liste chaînée de couches avec en queue, la couche dédiée au contenu principal et en tête, la dernière couche superposée. Soient l la couche courante dans le processus de segmentation d'une page spécifique p, $C = \{c_k\}$ l'ensemble des blocs résultant des traitements antérieurs sur p, $T = \{t_i\}$ celui de ses rectangles structurants. On dit de la segmentation de p qu'elle est réussie si, après avoir terminé avec succès la segmentation de la couche l, il ne reste plus un seul bloc de p qui ne soit attribué à l'une ou l'autre des régions de p. Ceci peut intervenir dans l'un des deux cas suivants :

1. la couche courante l est la dernière couche superposée, donc la tête de liste;
2. les couches suivantes superposées à la couche courante l, dans le modèle de la page p, sont absentes de la page p.

Autrement, tant qu'il reste une partie $C' \subset C$ des constituants de p non encore attribuée à une région r_j, on poursuit la segmentation avec la couche l' immédiatement superposée à l (la couche suivante dans la liste), le reste des constituants C' et l'ensemble des rectangles structurants T. La segmentation d'une couche l est dite réussie lorsque toutes ses régions sont complètement déterminées, c'est-à-dire lorsque pour toute région r, la position exacte de chacun de ses quatre séparateurs $(left, right, top, bottom)$ est connue.

Dans le but de déterminer, en premier, les régions ayant plus de séparateurs statiques ou de séparateurs déjà déterminés, nous avons attribué à chaque région une priorité d'analyse. Cette priorité est définie, par rapport à la priorité d'analyse de ses quatre séparateurs, de sorte que

la région ayant la priorité la plus élevée soit la suivante à déterminer parmi celles non encore analysées. Soit F, une fonction définissant la priorité d'analyse :

$$F(r) = f(r.top) + f(r.bottom) + f(r.left) + f(r.right)$$

où f désigne la fonction de priorité définie sur les séparateurs :

$$f(s) = \begin{cases} 2 & \text{si la position exacte de } s \text{ est déjà déterminée} \\ 1 & \text{si } s \text{ est un séparateur statique non encore extrait} \\ 0 & \text{sinon} \end{cases}$$

De part cette définition, une région sera qualifiée de complètement déterminée quand sa priorité est égale 8, en d'autres termes, quand la position de chacun de ses quatre séparateurs est déterminée. Cette approche de segmentation est traduite par l'algorithme 9.1.

---

segmenter(p, C):
  l = couche_principale(p);
  T = extraire_rectangles_structurants(C);
  FAIRE
    r = region_prioritaire(l);
    TANT QUE une telle region r existe FAIRE
      SI PAS déterminé($r.top$) ALORS déterminer($r.top$, C, T); FIN SI;
      SI PAS déterminé($r.bottom$) ALORS déterminer($r.bottom$, C, T); FIN SI;
      SI PAS déterminé($r.left$) ALORS déterminer($r.left$, C, T); FIN SI;
      SI PAS déterminé($r.right$) ALORS déterminer($r.right$, C, T); FIN SI;
      $C_r = \{c_k \in C | E(c_k) \subset E(r)\}$;
      $C = C - C_r$;
      r = region_prioritaire(l);
    FIN TANT QUE;
    l = couche_suivante(l);
  JUSQU'A CE QUE ($C == \{\}$) $\vee$ il n'existe plus de couche supérieure;
  $succès = (C == \{\})$;
FIN **segmenter**

Algorithme 9.1: Segmentation en régions d'une page spécifique.

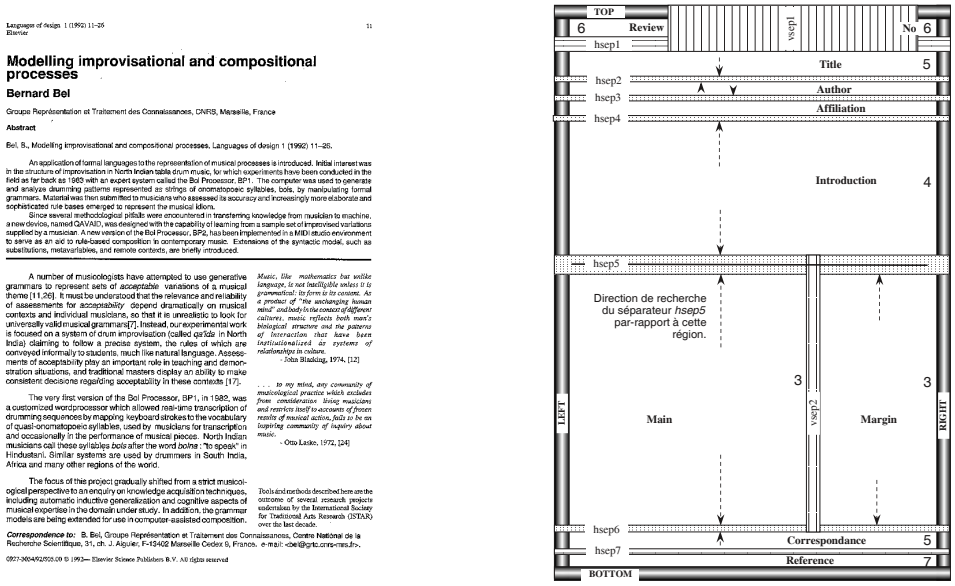
Au cours de la segmentation, la priorité des régions non encore déterminées augmente au fur et à mesure que l'on connaît la position exacte des séparateurs qu'elles partagent avec d'autres régions qui, elles, sont déjà déterminées. Sur la figure 9.1.b, on peut lire la priorité initiale de chacune des régions de la figure 9.1.a. Dans cette figure, la région nommée *Reference* a la priorité initiale la plus élevée; par conséquent, elle sera déterminée en premier lieu.

9.2.2 Extraction des séparateurs

Soit $s = (sep1, sep2, pos1, pos2, size)$ un séparateur défini comme le montre la figure 9.2 par les deux séparateurs $sep1$ et $sep2$ qui le délimitent aux extrémités, par son intervalle de positionnement $[pos1, pos2]$ et pour finir par sa taille minimale $size$. La position relative d'un séparateur statique ou élastique est donnée par un nombre entier (cf. section 8.4 consacrée à la description des séparateurs), ce qui revient à poser $pos1 = pos2$. Le séparateur s est déterminé par les coordonnées du rectangle structurant $t \in T$ dont la position et la taille concordent le mieux avec la position et la taille de s. Au lieu de rechercher un tel rectangle structurant parmi tous les éléments de T, nous avons limité la recherche à un domaine rectangulaire $d(s) = (x_1, x_2, y_1, y_2)$ que nous désignons par *domaine de recherche* (voir figure 9.2) et qui est défini comme suit :

$$d(s) = \begin{cases} (s.sep_1.pos_2, s.sep_2.pos_1, s.pos_1 - s.size, s.pos_2 + s.size) & \text{si } s \text{ est horizontal} \\ (s.pos_1 - s.size, s.pos_2 + s.size, s.sep_1.pos_2, s.sep_2.pos_1) & \text{si } s \text{ est vertical} \end{cases}$$

La position exacte d'un séparateur horizontal s est calculée par les coordonnées y_1 et y_2 du rectangle structurant $t \in T$ déterminé au moyen de l'algorithme 9.2.



(a) (b)

Figure 9.1: Priorités initiales d'analyse et directions de recherche des séparateurs flottants.

---

déterminer(s, C, T):
SI PAS déterminé($s.sep1$) ALORS déterminer($s.sep1$, C, T); FIN SI;
SI PAS déterminé($s.sep2$) ALORS détermine($s.sep2$, C, T); FIN SI;
(* à ce stade, s est forcément un séparateur statique ou flottant,
les séparateurs élastiques sont devenus statiques *);
$d = d(s)$;
SI static(s) ALORS
Chercher un rectangle structurant $t \in T$ tel que
$(t.x_1 < d.x_1) \wedge (t.x_2 > d.x_2) \wedge$
$(t.y_1 < d.y_2) \wedge (t.y_2 > d.y_1) \wedge ((t.y_2 - t.y_1) > s.size)$
ET $\forall t_i \neq t$, $(t.y_2 - t.y_1) > (t_i.y_2 - t_i.y_1)$
SI PAS trouvé un tel $t \in T$ ALORS
(* s est recouvert par les constituants de la couche suppérieure. *)
$C' = \{c_k \in C | E(c_k) \cap E(s) \neq \{\}\}$
déterminer(s, C', T');
FIN SI;
SINON (* s est flottant *)

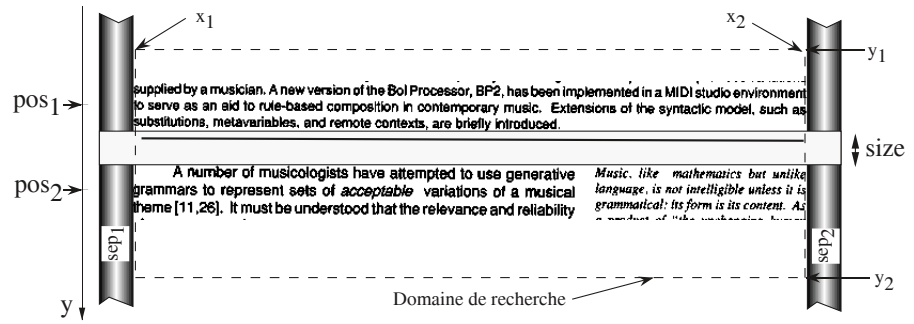
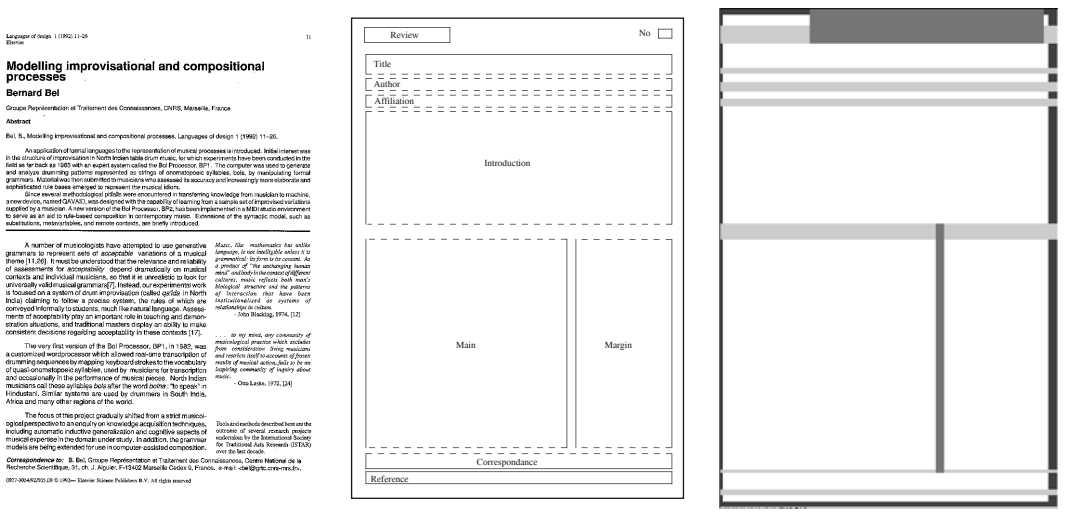


Figure 9.2: Attributs d'un séparateur horizontal.

Rechercher le premier rectangle structurant t tel que
$(t.x_1 < d.x_1) \wedge (t.x_2 > d.x_2) \wedge$
$(t.y_1 < d.y_2) \wedge (t.y_2 > d.y_1) \wedge ((t.y_2 - t.y_1) > s.size)$
SI PAS trouvé un tel $t \in T \wedge s$ possède un séparateur alternatif s_1 ALORS
$s = s_1$
déterminer(s, C, T);
FIN SI;
FIN SI;
SI l'on a trouvé un t qui concorde le mieux avec s ALORS
$s.pos1 = t.y1$;
$s.pos2 = t.y2$;
SINON (* Echec de l'extraction *) FIN SI;
FIN **déterminer**;

Algorithme 9.2: Détermination d'un séparateur horizontal.

S'agissant d'un séparateur flottant s, la direction de recherche des rectangles structurants concordants est orientée en fonction du bord (nord, sud, est ou ouest) que délimite s dans la région en cours d'analyse. La direction de recherche des séparateurs flottants de la figure 9.1 est mise en évidence par des pointillés orientés. Dans l'optique de pouvoir reviser le résultat d'une segmentation, tous les rectangles structurants dont la taille concorde avec $s.size$ et dont la position se situe dans l'intervalle $[s.pos_1, s.pos_2]$ sont également retenus comme d'éventuels alternatifs. Dans les figures 9.3 et 9.4 : (c) illustre, par rapport à la maquette (b), les séparateurs déterminés à partir de la page (a).

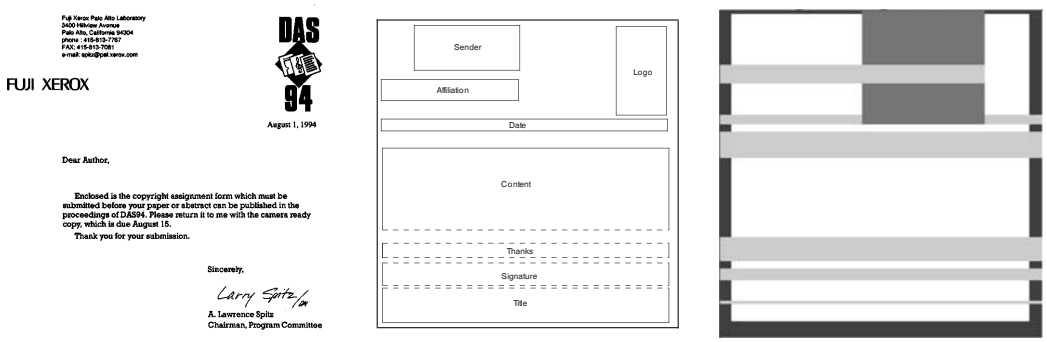


(a) (b) (c)

Figure 9.3: Illustration de la détermination des séparateurs (Doc. 1).

9.3 Segmentation en blocs et identification des tableaux

La reconnaissance des macrostructures, commencée avec la segmentation en régions (voir section 9.2) s'achève ici avec la segmentation des régions en blocs. L'objectif consiste à découper une région en blocs, récursivement et de façon alternative selon les axes X et Y, de sorte à isoler à chaque découpe les blocs dont les distances par rapport aux voisins sont supérieures à un seuil métrique donné. Ce seuil correspond à d_x^m (seuil maximal des espaces inter-mots) pour la découpe selon l'axe X et à d_y^l (seuil maximal des espaces inter-lignes) pour la découpe selon l'axe Y. On ne s'intéresse pas ici à l'étiquetage logique des blocs qui demande une analyse plus fine de leur



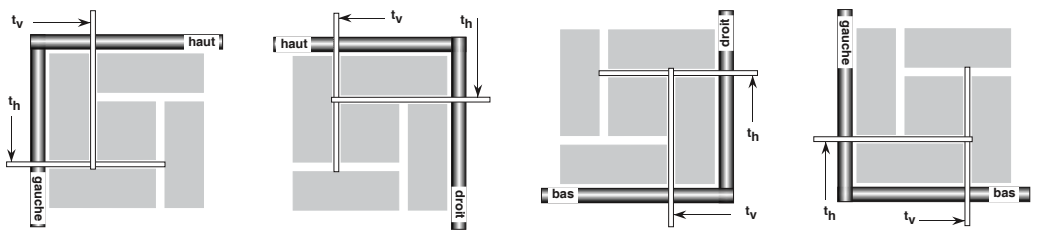
(a) (b) (c)

Figure 9.4: Illustration de la détermination des séparateurs (Doc. 2).

structure interne (cf. chapitre 6). La segmentation des régions en blocs est réalisée au moyen d'une technique de découpe récursive à laquelle nous avons apporté deux adaptations.

1. La recherche de structures mosaïques lorsqu'aucune des découpes classiques (en colonnes ou en rangées) n'est possible. Pour ce faire, nous recherchons un rectangle structurant horizontal t_h et un autre vertical t_v tels que la topologie locale des deux vérifie avec le bord du bloc en question l'une ou l'autre des quatre configurations de la figure 9.5.

2. Le regroupement en tableaux, pendant la découpe hiérarchique en blocs, des blocs dont la topologie de voisinage locale vérifie une structure matricielle (cf. sections 9.3.2 et 9.3.3).



(a) (b) (c) (d)

Figure 9.5: Types de structures mosaïques.

9.3.1 Découpe hiérarchique

La découpe hiérarchique d'un bloc consiste à déterminer, au moyen d'une analyse des rectangles structurants, la découpe adéquate de son image. Dans notre analyse, la découpe peut se faire en colonnes, en rangées ou suivant une structure mosaïque. Nous avons réalisé pour chacune de ces trois types de découpe, un algorithme approprié (cf. sections 9.3.2 à 9.3.3).

Soient p une page représentée par l'ensemble de ses constituants $C = \{c_k\}$ et r l'enveloppe rectangulaire de p. Soit maintenant $T = \{t_i\}$ l'ensemble des rectangles structurants extraits de l'image de p au moyen de l'algorithme 6.2 dont les paramètres sont définis comme suit :

1. $l_{min} = d_x^m$,

2. $h_{min} = d_y^l$.

d_x^m et d_y^l, définis respectivement le seuil métrique maximal des espaces inter-mots et celui des espaces inter-lignes en vigueur dans la page p (cf. section 5.1.3). Les rectangles structurants verticaux (RSV), servant soit à découper la page p en colonnes ou suivant une structure mosaïque ont été répartis en quatre catégories comme le montre la figure 9.6 :

- T_c : ensemble des RSV traversant p de haut en bas,
- T_c^h : ensemble des RSV traversant p dans sa partie supérieure,
- T_c^b : ensemble des RSV traversant p dans sa partie inférieure,
- T_c^i : ensemble des RSV situés à l'intérieur de p,

Figure 9.6: Catégories de rectangles structurants verticaux.

Nous avons défini de façon analogue quatre catégories de rectangles structurants horizontaux (RSH) T_r, T_r^g, T_r^d et T_r^i servant soit une découpe en rangées ou suivant une structure mosaïque. La structure hiérarchique d'un bloc défini par son enveloppe r, par l'ensemble C des ses composantes et par l'ensemble T de ses rectangles structurants est déterminée au moyen de l'algorithme 9.3 qui se termine lorsqu'il n'est plus possible de poursuivre la découpe en blocs; c'est le cas lorsqu'on se trouve en présence d'une microstructure.

---

découpe_hiérarchique(r, C, T):
$T_c = \{t_i \in T | (t_i.x_1 > r.x_1) \wedge (t_i.x_2 < r.x_2) \wedge (t_i.y_1 < r.y_1) \wedge (t_i.y_2 > r.y_2)\}$
$T_r = \{t_i \in T | (t_i.y_1 > r.y_1) \wedge (t_i.y_2 < r.y_2) \wedge (t_i.x_1 < r.x_1) \wedge (t_i.x_2 > r.x_2)\}$
SI $T_c \neq \{\}$ ALORS
découpe_en_colonnes(r, C, T, T_c, T_r);
SINON
$T_c^h = \{t_i \in T | (t_i.x_1 > r.x_1) \wedge (t_i.x_2 < r.x_2) \wedge (t_i.y_1 < r.y_1) \wedge (t_i.y_2 > r.y_1)\}$
$T_c^b = \{t_i \in T | (t_i.x_1 > r.x_1) \wedge (t_i.x_2 < r.x_2) \wedge (t_i.y_1 < r.y_2) \wedge (t_i.y_2 > r.y_2)\}$
$T_c^i = \{t_i \in T | (t_i.x_1 > r.x_1) \wedge (t_i.x_2 < r.x_2) \wedge (t_i.y_1 > r.y_1) \wedge (t_i.y_2 < r.y_2)\}$
SI $T_r \neq \{\}$ ALORS
découpe_en_rangées(r, C, T, T_r, T_c^h, T_c^b, T_c^i);
SINON
$T_r^g = \{t_i \in T | (t_i.y_1 > r.y_1) \wedge (t_i.y_2 < r.y_2) \wedge (t_i.x_1 < r.x_1) \wedge (t_i.x_2 > r.x_1)\}$
$T_r^d = \{t_i \in T | (t_i.y_1 > r.y_1) \wedge (t_i.x_2 < r.y_2) \wedge (t_i.x_1 < r.x_2) \wedge (t_i.x_2 > r.x_2)\}$
SI $T_c^h \neq \{\} \wedge T_r^g \neq \{\}$ ALORS
succès = découpe_mosaïque_haut_gauche(r, C, T, T_c^h, T_r^g);
SI PAS succès ET $T_c^h \neq \{\} \wedge T_r^d \neq \{\}$ ALORS
succès = découpe_mosaïque_haut_droit(r, C, T, T_c^h, T_r^d);
SI PAS succès ET $T_c^b \neq \{\} \wedge T_r^g \neq \{\}$ ALORS
succès = découpe_mosaïque_bas_gauche(r, C, T, T_c^b, T_r^g);
SI PAS succès ET $T_c^b \neq \{\} \wedge T_r^d \neq \{\}$ ALORS
succès = découpe_mosaïque_bas_droit(r, C, T, T_c^b, T_r^d);
SINON
(* il s'agit d'une microstructure cf. algorithme 6.3*)
FIN SI
FIN SI
FIN SI

FIN **découpe_hiérarchique**;

Algorithme 9.3: Segmentation en blocs.

9.3.2 Découpe en colonnes

La découpe en colonnes est réalisée au moyen de l'algorithme 9.4 fondé sur une analyse des rectangles structurants verticaux T_c préalablement triés selon la coordonnée x_1 de leur enveloppe. Sur chaque colonne C_c, ainsi extraite, on réapplique l'algorithme de découpe hiérarchique 9.3.

découpe_en_colonnes(r, C, T, T_c, T_r):
  C_c : désigne l'ensemble des constituants de la colonne courante;
  $r_c = r$: où r_c désigne l'enveloppe de la colonne courante;
  trier les éléments de T_c par rapport à la coordonnée x_1 de leur enveloppe;
  POUR CHAQUE i ALLANT DE 1 A $N(T_c) + 1$ FAIRE
    SI $i \leq N(T_c)$ ALORS $r_c.x_2 = T_c[i].x_1$;
    SINON $r_c.x_2 = r.x_2$ FIN SI;
    $C_c = \{c_k \in C | B(c_k) \subset r_c\}$;
    découpe_hiérarchique(r_c, C_c, T);
    $C = C - C_c + \{C_c\}$;
    SI $i \leq N(T_c)$ ALORS $r_c.x_1 = T_c[i].x_2$; FIN SI;
  FIN POUR CHAQUE;
  SI $(N(T_r) \geq 2) \vee (N(T_c) \geq 2)$ ALORS attribuer à C l'étiquette $TABLEAU$ FIN SI
FIN **découpe_en_colonnes**;

Algorithme 9.4: Découpe récursive en colonnes.

Lorsqu'il est également possible de subdiviser le même bloc C en rangées, au moyen de l'ensemble des rectangles structurants horizontaux T_r, on décide alors qu'on est en présence d'une structure matricielle et le bloc C est étiqueté comme tel, c.-à-d. TABLEAU.

9.3.3 Découpe en rangées

La découpe en rangées est réalisée au moyen de l'algorithme 9.5 fondé sur une analyse des rectangles structurants horizontaux T_r préalablement triés selon la coordonnée y_1 de leur enveloppe. On profite de la connaissance des rectangles structurants verticaux donnés par T_c^h, T_c^b et T_c^i pour déterminer la présence éventuelle de structures matricielles. Notre objectif est d'éviter la découpe en rangées d'éventuels tableaux puisque nous avons choisi de privilégier la structure en colonnes (cf. règles 5.5).

découpe_en_rangées(r, C, T, T_r, T_c^h, T_c^b, T_c^i):
  y_1 : désigne l'ordonnée y_2 du RSH précédant le rectangle courant;
  y_2 : désigne l'ordonnée y_1 du RSH succédant au rectangle courant;
  C_r : désigne l'ensemble des constituants de la rangée courante;
  $r_r = r$: où r_r désigne l'enveloppe de la rangée courante;
  Trier les éléments de T_r par rapport à la coordonnée y_1 de leur enveloppe;
  POUR CHAQUE i ALLANT DE 1 A $N(T_r) + 1$ FAIRE
    $j = i$;
    TANT QUE ($j \leq N(T_r)$ FAIRE
      SI $i == 1$ ALORS $y_1 = r.y_1$;
      SINON $y_1 = T_r[i-1].y_2$ FIN SI;
      SI $i == N(T_r)$ ALORS $y_2 = r.y_2$;
      SINON $y_2 = T_r[i+1].y_1$ FIN SI;
      SI $\exists\ t_c^1,\ t_c^2 \in T_c^h \cup T_c^b \cup T_c^i$ TELQUE
        $(t_c^1.y_1 \leq y_1) \wedge (t_c^1.y_2 \geq y_2) \wedge (t_c^2.y_1 \leq y_1) \wedge (t_c^2.y_2 \geq y_2)$
      FAIRE
        $j = j + 1$;
      SINON sortir de TANT QUE; FIN SI;
    FIN TANT QUE;
    SI $j \leq N(T_r)$ ALORS $r_r.y_2 = T_r[j].y_1$;
    SINON $r_r.y_2 = r.y_2$ FIN SI;
    $C_r = \{c_k \in C | B(c_k) \subset r_r\}$;
    découpe_hiérarchique(r_r, C_r, T);

SI $i < j$ ALORS attribuer à C_r l'étiquette $TABLEAU$ FIN SI
$C = C - C_r + \{C_r\}$;
SI $j \leq N(T_r)$ ALORS $r_r.y_1 = T_r[j].y_2$; FIN SI;
FIN POUR CHAQUE;
FIN **découpe_en_rangées**;

Algorithme 9.5: Découpe récursive en rangées.

Au cours de la découpe en rangées, on élimine de l'ensemble T_r tout RSH t_h pour lequel il existe au moins deux RSV t_v^1 et $t_v^2 \in T_c^h \cup T_c^b \cup T_c^i$ tels que aussi bien t_v^1 que t_v^2 traversent non seulement t_h mais aussi le RSH t_h^1 précédant t_h dans T_r et le RSH t_h^2 lui succédant (voir figure 9.7). Lorsque t_h est le premier élément de T_r alors t_h^1 devient le bord supérieur du bloc C et lorsqu'il est le dernier élément de T_r, t_h^2 devient le bord inférieur du bloc C. Cette élimination a pour but de laisser groupés les blocs dont la topologie de voisinage locale reflète une structure matricielle; de tels blocs sont étiquetés TABLEAU. La segmentation se poursuit sur chaque bloc résultant de la découpe en rangées par l'application de l'algorithme 9.3.

(a) (b) (c)

Figure 9.7: Rectangles structurants vérifiant une structure matricielle.

9.3.4 Découpe des blocs mosaïques

Le principe de l'algorithme 9.6 de découpe suivant une structure mosaïque consiste à rechercher parmi les couples de rectangles structurants $(t_v, t_h) \in T_c^h \times T_r^g$, vérifiant la configuration de la figure 9.5.a, celui dont la diagonale est la plus grande. La segmentation se poursuit sur chaque bloc résultant de la découpe du bloc mosaïque par l'application de l'algorithme 9.3. Pour l'analyse des autres formes de mosaïques, l'algorithme est analogue, à quelques changements près, à l'algorithme 9.6.

découpe_mosaïque_haut_gauche(r, C, T, T_c^h, T_r^g):
C_b : désigne l'ensemble des constituants du bloc haut gauche dans la structure mosaïque C;
$r_b = r$: où r_b désigne l'enveloppe d'un bloc;
$U = \{(t_v, t_h) \in T_c^h X T_r^g | (t_h.x_1 < t_v.x_2) \wedge (t_v.y_1 < t_h.y_2)\}$
Si $\exists (t_v, t_h) \in U \mid \forall (t_1, t_2) \in U,\ (t_v.x_1 - r.x_1)^2 + (t_h.y_1 - r.y_1)^2 \geq (t_1.x_1 - r.x_1)^2 + (t_2.y_1 - r.y_1)^2$ ALORS
$r.x_2 = t_v.x_1$;
$r.y_2 = t_h.y_1$;
$C_b = \{c_k \in C | B(c_k) \subset r_b\}$;
découpe_hiérarchique(r_b, C_b, T);

```
    C = C − C_b;
    découpe_hiérarchique(r, C, T);
    C = C + {C_b};
    retourner VRAI; (* c.-à-d. la découpe a été possible *)
  SINON retourner FAUX; (* c.-à-d. découpe pas possible *) FIN SI;
FIN découpe_mosaïque_haut_gauche;
```

Algorithme 9.6: Découpe récursive de blocs mosaïques de type haut gauche.

9.3.5 Analyse des tableaux

Dans cette section, nous avons profité de l'analyse des rectangles structurants, lors de la découpe en colonnes (cf. section 9.3.2) ainsi que lors de la découpe en rangées (cf. section 9.3.3), pour déterminer la pésence de structures matricielles que nous avons étiquetées TABLEAU. A l'opposé des techniques présentées dans la littérature [66, 92, 67, 68] pour la reconnaissance des tableaux, notre approche ne présuppose la présence d'aucun filet pour guider la reconnaissance de ces derniers. Ainsi, pour la reconnaissance des tableaux, nous avons fait abstraction des filets présents dans les documents traités en basant notre analyse uniquement sur les espaces. En effet, comme nous le présentons dans la section 4.4 consacrée aux délimiteurs :

- le quadrillage des tableaux varie d'un document à l'autre,
- la présence de filets dans l'aspect graphique des tableaux n'a pour seul but que d'augmenter leur lisibilité en renforçant la séparation entre les cellules.

Notre approche, qui s'affranchit de la particularité du quadrillage des tableaux, a donné de très bons résultats sur l'ensemble des documents traités (cf. section 6.3.4 dans le chapitre 6 consacré à la reconnaissance des microstructures).

Conclusion

Dans l'annexe B, nous avons illustré le résulat de la reconnaissance des (macro et micro) structures physiques sur un échantillon de 13 documents représentatifs de la famille des documents composites. L'utilisation des rectangles structurants pour la segmentation en blocs a permis d'atteindre des résultats au moins équivalents à ceux obtenus par les techniques de projection de profil récursive. En mieux, cette méthode permet d'atteindre :

- une meilleure efficacité grâce à la diminution de la quantité d'information traitée; en effet, dans une image, l'analyse des rectangles structurants coûte moins chère que celle des pixels (cf. section 6.3);
- une meilleure fiabilité grâce à une estimation plus fiable des seuils métriques de découpe par une analyse statistique de la taille des rectangles structurants signifiants (cf. section 7).

Dans le chapitre 10, nous présentons une évaluation aussi bien qualitative que quantitative des différentes méthodes que nous avons mises au point dans cette thèse. A cet effet, nous avons défini dans l'espace des structures physiques une distance métrique servant à comparer entre-elles deux structures physiques.

Chapitre 10

Evaluation

Dans ce chapitre, notre objectif est celui d'évaluer les différentes primitives contribuant à la reconnaissance de structures physiques et développées dans cette thèses. Dans la section 10.1, nous définissons une distance métrique dans l'espace des structures arborescentes en général. Dans la section 10.2, nous définissons une distance métrique, variante de la première, dans l'espace des structures physiques en vue de comparer entre-elles les structures physiques. Dans la section 10.3, nous évaluons les primitives de reconnaissance de structures physiques que nous avons développées en comparant, au moyen de la distance définie dans la section 10.2, les résultats produits avec les structures physiques escomptées.

10.1 Distance dans l'espace des structures arborescentes

10.1.1 Rappel

Soit d une fonction définie sur des paires de structures arborescentes; d est qualifiée de distance métrique si et seulement si elle vérifie les trois axiomes suivants :

1. $\forall\ T_1,\ T_2\ d(T_1, T_2) \geq 0$ et $= 0$ si $T_1 = T_2$,
2. $\forall\ T_1,\ T_2\ d(T_1, T_2) = d(T_2, T_1)$,
3. $\forall\ T_1,\ T_2,\ T_3\ d(T_1, T_3) \leq d(T_1, T_2) + d(T_2, T_3)$.

10.1.2 Coût de la transformation d'un arbre en un autre

Désignons par $E(t)$ l'ensemble des feuilles d'un arbre (ou sous-arbre) binaire de racine t. Soient T_1, T_2 deux arbres ayant chacun n feuilles étiquetées de 1 à n, $\Omega(T_1) = \{E(t_1^1) \ldots E(t_1^{n-1})\}$, $\Omega(T_2) = \{E(t_2^1) \ldots E(t_2^{n-1})\}$ dans lesquels $E(t_i)$ est répété $(e-1)$ fois, où e désigne le nombre d'arêtes partant du père de t_i. Soit d_c une distance définie comme suit :

$$d_c(T_1, T_2) = \min_f \sum_{i=1}^{n-1} \delta(E(t_1^i), E(t_2^{f(i)})) \tag{10.1}$$

Dans la fonction 10.1, f désigne une permutation des $(n-1)$ premiers entiers et $\delta(E(t_1^i), E(t_2^j))$ calcule le nombre de non-correspondances une à une entre les feuilles de t_1^i et celles de t_2^j. La distance d_c sert à calculer le nombre de déplacements (retraits + ajouts) nécessaires pour transformer un arbre T_1 en T_2.

Exemple Soient T_1 et T_2 les deux arbres de la figure 10.1 : le coût de la transformation de T_1 en T_2 est calculé comme suit :

- $\Omega(T_1) = \{E_1^1 = \{a, b\}, E_1^2 = \{a, b, c\}, E_1^3 = \{a, b, c, d\}\}$,
- $\Omega(T_2) = \{E_2^1 = \{a, b\}, E_2^2 = \{c, d\}, E_2^3 = \{a, b, c, d\}\}$,
- $d_c(T_1, T_2) = 3$,
- avec par exemple : $\delta(E(t_1^1), E(t_2^3)) = 2$ et $\delta(E(t_1^2), E(t_2^2)) = 3$.

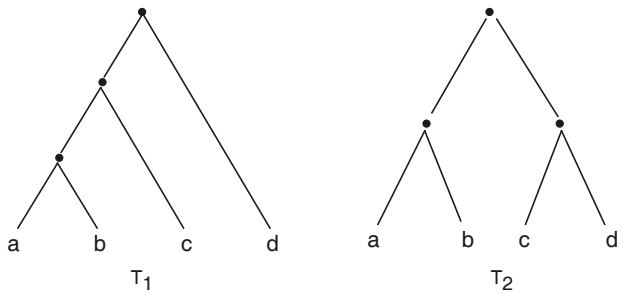


Figure 10.1: Distance entre deux arbres : $d_c(T_1, T_2) = 3$.

Le lecteur désireux d'en savoir plus sur la métrique des structures arborescentes et des treillis peut se rapporter aux travaux de S. A. Boorman et D. C Olivier présentés dans [94].

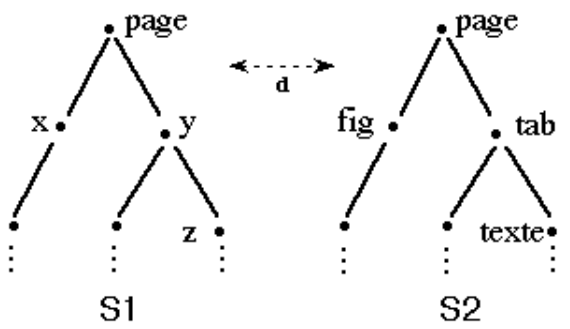
10.2 Distance dans l'espace des structures physiques

Dans l'optique de pouvoir calculer le degré de ressemblance entre des couples de structures physiques, nous avons établi une distance métrique d qui une variante de la distance d_c définie dans la section 10.1. La différence avec cette dernière se situe, d'une part, dans la définition de la fonction δ et, d'autre part, dans le fait que la distance d a été normalisée pour en faciliter l'interprétation; ces valeurs sont comprises entre 0 et 1. On dira de deux structures physiques S_1 et S_2 qu'elles sont :

- totalement semblables si et seulement si $d(S_1, S_2) = 0$,
- totalement disjointes si et seulement si $d(S_1, S_2) = 1$,
- et partiellement semblables (ou disjointes) si et seulement si $d(S_1, S_2) < 1$.

Nous rappelons ci-dessous les primitives définies dans la section 5.1.2 et qui serviront, ici, à établir la distance d :

- $etiq(e)$: étiquette logique associée à e,
- $N(e)$: le nombre de descendants directs de e,
- $I_y(e_1, e_2)$: longueur de l'intersection sur l'axe Y des profils horizontaux,
- $U_y(e_1, e_2)$: longueur de l'union sur l'axe Y des profils horizontaux,
- $I_x(e_1, e_2)$: longueur de l'intersection sur l'axe X des profils horizontaux,
- $U_x(e_1, e_2)$: longueur de l'union sur l'axe X des profils horizontaux,

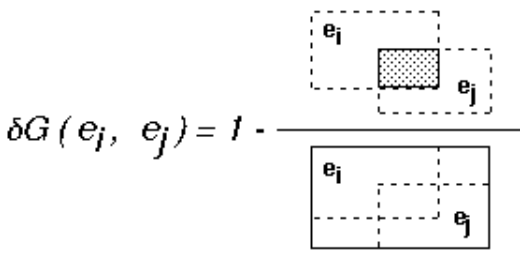


Notre objectif, en établissant la distance d, est de pouvoir estimer aussi bien la correspondance spatiale que la correspondance structurelle entre une structure physique déterminée avec celle escomptée :

$$d(S_1,S_2) = \begin{cases} \delta(S_1,S_2) & \text{si } N(S_1)=0 \text{ où } N(S_2)=0 \\ c\times\delta(S_1,S_2) + \frac{1-c}{2\times N(S_1)} \sum\limits_{e_i\in S_1} \min\limits_{e_j\in S_2}(d(e_i,e_j)) & \\ \qquad + \frac{1-c}{2\times N(S_2)} \sum\limits_{e_j\in S_2} \min\limits_{e_i\in S_1}(d(e_i,e_j)) & \text{sinon} \end{cases}$$

avec :

- $c \in]0,1[$ est un facteur de pondération indiquant le poids de la confrontation directe entre S_1 et S_2 alors que $(1-c)$ indique le poids de la confrontation cumulée de leurs descendants.
- $\delta(e_i,e_j) = k\times\delta_G(e_i,e_j) + (1-k)\times\delta_S(e_i,e_j)$
  - $\delta_G(e_i,e_j) = 1 - \frac{\max(0,I_x(e_i,e_j))\max(0,I_y(e_i,e_j))}{U_x(e_i,e_j)U_y(e_i,e_j)}$ une distance géométrique,



  - $\delta_S(e_i,e_j) = \begin{cases} 0 & \text{si } etiq(e_i)=etiq(e_j) \\ 1 & \text{sinon} \end{cases}$ une distance structurelle,
  - $k \in]0,1[$ est un facteur de pondération indiquant la part prise, dans la confrontation de e_i avec e_j, par la distance géométrique δ_G et $(1-k)$ la part prise par distance structurelle δ_S :
    - lorsque $k=1$, la distance d est qualifiée de géométrique, elle mesure alors le degré de ressemblance spatiale entre couples de structures physiques;
    - lorsque $k=0$, la distance d est qualifiée de structurelle, elle mesure alors le degré de ressemblance structurelle entre couples de structures physiques.

10.3 Evaluation des techniques de reconnaissance

10.3.1 Une architecture à la carte

L'architecture de notre prototype de reconnaissance se présente sous la forme d'un environnement utilisateur interactif offrant des primitives de reconnaissance de structures physiques. Toutes ces primitives sont indépendantes les unes des autres ce qui constitue un avantage certain quant à leur :

- combinaison dans des systèmes de reconnaissance particuliers,
- intégration dans une plateforme multi-agents ou parallèle.

Les résultats présentés aussi bien dans le chapitre 6 que dans l'annexe B ont été obtenus par le cycle de traitements suivant :

1. classification des CCX,
2. reconnaissance des signes,
3. reconnaissance des expressions fractionnaires,
4. reconnaissance des mots,
5. reconnaissance des expressions exponentielles,
6. reconnaissance des expressions fractionnaires,
7. reconnaissance des lignes,
8. reconnaissance des expressions fractionnaires,
9. reconnaissance des lignes,
10. reconnaissance des expressions bornées,
11. reconnaissance des lignes,
12. reconnaissance des expressions fractionnaires,
13. reconnaissance des expressions composées,
14. reconnaissance des lignes,
15. reconnaissance des textes,
16. extraction des rectangles structurants,
17. segmentation en régions,
18. segmentation en blocs et identification des tableaux.

Un cycle de traitements bien ciblés peut être conçu en fonction du type particulier de documents à reconnaître. Par exemple, pour des documents purement textuels, le cycle ci-dessus peut être considérablement allégé, les primitives d'analyse d'expressions mathématiques ou de blocs graphiques n'étant plus nécessaires.

10.3.2 Environnement d'évaluation

Nos développements ont été réalisés dans le langage de programmation C sur une machine SUN SPARC 5. Pour l'évaluation, nous avons retenu des documents constitués d'une page et tirés de diverses classes : (a) cinq pages d'un article tiré de la revue scientifique*Language of Design Review* (LDR), (b) huit pages d'un chapitre tiré d'un livre scientifique traitant de la thermodynamique, (c) une page représentative des correspondances bureautiques et (d) une page caractérisée par une forte proportion d'expressions mathématiques (cf. annexe B).

Dans les tableaux de synthèse présentés dans cette section, nous référençons de façon systématique les figures (résultat) auxquelles se rapportent les chiffres. La plupart des documents traités ont été digitalisés à 400 *dpi* pour assurer un traitement qui soit peu sensible aux bruits. Toutefois, pour tester la robustesse des primitives, nous avons également utilisé des documents digitalisés à 300 et 200 *dpi*.

10.3.3 Classification des composantes connexes

Pour ce qui est de l'évaluation de notre technique de classification des CCX, nous avons choisi de vérifier le postulat émis dans cette thèse et selon lequel il y aurait plus de signes diacritiques dans un document rédigé en français que dans un document rédigé en anglais, les signes diacritiques étant limités dans ce dernier au "." dans les lettres comme "i" et "j". A cet effet, nous avons calculé en fonction de la langue de rédaction, le pourcentage des CCX étiquetées *ponctuation* (signes diacritiques compris) par rapport à celles étiquetées *lettre* dans les documents traités. Dans le tableau 10.1 résumant ces pourcentages, la faible différence observée entre la proportion des composantes connexes étiquetées *ponctuation* et celles étiquetées *lettre* provient des bruits ainsi que des petites composantes connexes, courantes dans la texture des photographies et étiquetés comme *ponctuation*. Un autre résultat intéressant révélé par ce tableau réside dans la proportion des lettres sans hampe ni jambage (*lettre_a*) qui représentent en général les 70.00% de l'ensemble des lettres.

Tableau 10.1: Rapport en % des différentes classes de CCX calculées.

Doc.	Fig.	Langue	CCX	Ponct.	Lettre			Autres
					Total	Lettre_a	Lettre_b	
1	B.2	anglais	3917	10.79%	88.10%	44.74%	54.76%	1.09%
2	B.4	anglais	2166	9.04%	90.81%	71.63%	28.06%	0.13%
3	B.5	anglais	2486	12.26%	87.57%	69.54%	28.84%	0.16%
4	B.6	anglais	2166	10.48%	88.91%	66.14%	32.96%	0.60%
Moyenne			**2684**	**10.64%**	**88.85%**	**63.01%**	**36.16%**	**0.60%**
5	6.9	français	1682	16.58%	83.23%	63.78%	35.64%	0.17%
6	B.12	français	1839	20.82%	78.84%	66.89%	33.10%	0.32%
7	B.13	français	1715	11.95%	87.81%	65.67%	33.86%	0.23%
8	B.10	français	2647	13.90%	86.09%	70.99%	28.96%	0.00%
9	B.7	français	981	18.75%	80.63%	71.68%	27.68%	0.61%
10	B.8	français	1888	22.35%	76.43%	74.15%	25.01%	1.21%
11	B.9	français	1515	14.38%	84.88%	61.97%	37.40%	0.72%
12	B.11	français	2179	17.30%	82.51%	70.30%	29.64%	0.18%
Moyenne			**1806**	**17.00%**	**82.55%**	**68.18%**	**31.41%**	**0.43%**

La fiabilité de la classification des composantes connexes a été déterminante dans la reconnaissance des microstructures. Il ressort de nos évaluations qu'elle est relativement plus efficace que l'extraction des composantes connexes (cf. tableau 10.4).

10.3.4 Estimation des seuils métriques

Le tableau 10.2 permet de comparer les seuils métriques d_y^d, d_x^c, d_x^m, d_y^l et d_y^b estimés à partir d'une analyse statistique des espaces séparant les entités physiques. L'estimation du seuil d_x^m, relatif aux espaces séparant les mots, est réalisée de deux manières différentes : (a) une première fois à partir de la distribution des espaces inter-lettres et (b) une seconde fois à partir de la distribution des espaces inter-mots.

Tableau 10.2: Seuils métriques estimés en *pixels*.

Doc.	Fig.	CCX	h_f	l_f	d_y^d	d_x^c	d_x^m (a)	d_x^m (b)	d_y^l	d_y^b
1	B.2	3917	28	25	6	9	34	58	39	79
2	B.4	2166	32	27	7	10	37	37	33	68
3	B.5	2486	32	27	7	9	48	61	40	73
4	B.3	1835	32	6	7	5	7	20	48	71
5	B.6	2166	32	27	6	9	39	44	38	69
6	6.9	1682	27	22	8	9	33	45	35	71
7	B.12	1839	27	22	8	8	39	39	32	55
8	B.13	1715	21	17	8	7	28	32	22	55
9	B.10	2647	27	22	9	7	9	35	35	75
10	B.7	981	27	22	8	6	35	35	28	52
11	B.8	1888	27	22	8	8	33	42	24	73
12	B.9	1515	27	22	8	9	40	43	25	73
13	B.11	2179	27	22	8	10	35	36	28	74
14	B.1	425	27	20	6	8	9	24	49	49
15	6.8	1545	28	27	9	11	32	37	31	81

Dans le tableau 10.2, on observe, pour chaque catégorie de seuils métriques et pour les pages d'un même document, des différences dans les valeurs estimées qui semblent à première vue importantes. Cette observation a peu de conséquences en raison de la marge qui existe entre les espaces séparant des entités d'une catégorie donnée et les espaces séparant les entités de la catégorie hiérarchique supérieure. Par exemple, c'est grâce à une telle marge entre les espaces inter-caractères et les espaces inter-mots, qu'il ne nous arrive pas plus souvent de confondre une ligne de texte avec un mot. Deux autres remarques s'imposent :

1. quand bien même le seuil d_x^m, estimé à partir de la distribution (a), est souvent assez proche (sinon égal) de celui estimé à partir de la distribution (b), nous avons opté pour la seconde distribution pour assurer une plus grande fiabilité;

2. pour les pages d'un même document, la variation des valeurs estimées s'explique par la variation de leur contenu et en particulier par la variation des fontes.

Dans la pratique, les seuils métriques ont été estimés par apprentissage à partir de pages ou de blocs typiques, caractérisés par une forte proportion de texte. Cette estimation a donné de bons résultats sans lesquels notre approche de reconnaissance n'aurait pas été concluante.

10.3.5 Fiabilité

Pour l'évaluation de la reconnaissance des macrostructures, nous avons réalisé une base de données constituée des macrostructures d'une vingtaine de documents que nous avons étiquetés à la main. Ainsi, la macrostructure reconnue pour chaque document a pu être confrontée à la macrostructure attendue, pour juger de la qualité du résultat. Cette confrontation consiste à calculer la distance

entre les deux macrostructures au moyen de la distance métrique d définie dans la section 10.2. En ce qui concerne les microstructures, la constitution d'une base de données est beaucoup plus fastidieuse et prendrait trop de temps dans le cadre de cette thèse. Toutefois, pour évaluer la fiabilité de la reconnaissance des microstructures, nous avons attribué aux nœuds des macrostructures escomptées, une étiquette logique servant à valider la reconnaissance des microstructures.

Il ressort de notre expérimentation que la reconnaissance des macrostructures, lorsqu'elle a lieu après la reconnaissance des microstructures, est à la fois fiable et efficace. Cet observation est justifiée par une réduction des rectangles structurants ayant servi à déterminer la position exacte des séparateurs de régions. L'efficacité provient du fait qu'il n'a plus été nécessaire de réestimer les seuils métriques utilisés pour la segmentation des régions en blocs, ces derniers ayant été déjà déterminés lors de la reconnaissance des microstructures. Du tableau 10.3, résumant la fiabilité de cette approche de reconnaissance, trois observations s'imposent :

Tableau 10.3: Distance entre les structures physiques calculées et celles escomptées.

		d avec $c = 1/3$		
Doc.	Fig.	$k = 1$	$k = 0$	$k = 1/2$
1	B.2	0.0000	0.0000	0.0000
2	B.4	0.0121	0.0691	0.0118
3	B.5	0.0089	0.0543	0.0083
4	B.3	0.0101	0.0574	0.0091
5	B.6	0.0089	0.0514	0.0082
6	6.9	0.0000	0.0311	0.0000
7	B.12	0.0039	0.0825	0.0066
8	B.13	0.0131	0.1300	0.0106
9	B.10	0.0000	0.0000	0.0000
10	B.7	0.0225	0.1529	0.0263
11	B.8	0.0064	0.0791	0.0069
12	B.9	0.0119	0.1082	0.0080
13	B.11	0.0020	0.0533	0.0020
14	B.1	0.0273	0.0416	0.0206
15	6.8	0.0303	0.4760	0.0151
Moyenne		**0.0105**	**0.0925**	**0.0089**

1. La fiabilité de la découpe hiérarchique des documents traités est traduite par une distance géométrique ($k = 1$) très faible, parfois nulle, avec la découpe escomptée. Si pour des documents purement textuels la reconnaissance est presque parfaite, pour des documents de contenu varié, on note une ou deux erreurs locales de sous ou sur-segmentation.

2. La distance structurelle ($k = 0$) est en revanche relativement plus élevée par rapport à la distance géométrique ($k = 1$). Ce n'est ni la découpe, ni l'étiquetage qui est en cause, mais plutôt l'ordre de lecture des blocs qui, parfois, n'est pas conforme à l'ordre escompté. Ceci montre les limites du tri spatial des blocs.

3. La distance corrélée d avec $k = 1/2$ confirme l'argument que nous donnons au point 2. Comme l'indique les distances moyennes, lorsque l'on tient compte de l'étiquetage des entités lors de la confrontation spatiale, la distance d entre les structures physiques calculées et celles escomptées, en général, baisse. Il s'agit là d'une preuve que l'étiquetage des découpes hiérarchiques est lui aussi fiable.

Plus généralement, nous avons noté que la fiabilité de l'étiquetage dépend, avant tout, de la fiabilité de la découpe hiérarchique. La reconnaissance des structures physiques est :

- parfaite lorsque $d(S_1, S_2) = 0$,
- bonne lorsque $0 < d(S_1, S_2) \leq 0.02$ ce qui entraine une correction manuelle limitée à une ou deux entités,
- acceptable lorsque $0.02 < d(S_1, S_2) \leq 0.05$, dans ce cas, une correction manuelle est encore envisageable,
- inacceptable sinon.

Les erreurs de segmentation ne sont pas liées à nos primitives en tant que telles. Elles proviennent essentiellement du formatage des expressions mathématiques. En effet, la seule utilisation des espaces ne permet pas de regrouper une expression mathématique avec sa référence qui généralement est assez éloignée de l'expression elle-même (cf. figure 10.2) alors que, dans notre base de données, nous avons regroupé ces deux choses. Cette façon de faire est "peut-être" fausse, lorsque l'on se place uniquement du point de vue de la segmentation qui, elle, se base sur une analyse des espaces, comme c'est le cas dans notre approche.

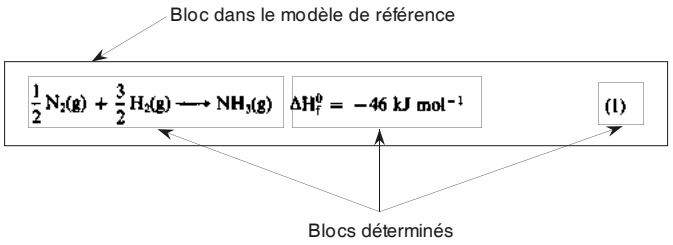


Figure 10.2: Illustration de l'éloignement d'un bloc calculé par rapport au bloc de référence.

Dans une approche d'analyse où plusieurs plusieurs techniques coopèrent pour la reconnaissance d'un document, la distances d peut servir à juger de la concordance des résultats produits par des techniques différentes.

10.3.6 Efficacité

Le temps d'exécution des différentes primitives évaluées a été estimé au moyen de la fonction *time* de la librairie standard *stdlib* du langage de programmation C. L'estimation est donnée par la différence du temps à l'entrée et à la sortie des primitives. L'unité de temps étant la seconde, de nombreux temps d'exécution ont été récupérés arrondis à 0 seconde ce qui, à notre avis, est plutôt signe d'efficacité. Il est également important de noter que l'évaluation a été réalisée sur une machine en réseau qui pouvait, à tout moment, être sollicitée par d'autres jobs. Par conséquent, le temps d'exécution estimé correspond au temps réel d'attente qui est toujours supérieur au temps CPU effectif.

Dans cette évaluation, nous avons choisi de mettre en évidence la proportion du temps passé dans les primitives importantes par rapport au temps d'attente nécessité par l'analyse complète d'un document. Dans le tableau 10.4, récapitulatif de cette évaluation, H et L désignent en *pixels* la hauteur et la largeur de l'image des documents et N, le nombre de CCX extraites. En moyenne, 52.80% du temps total de calcul est consacré à l'extraction des CCX, 32.90% à la classification des CCX et seulement 14.00% à la reconnaissance proprement dite alors que celui de l'estimation des seuils métriques est de 0.06%.

- Parmi les traitements incompressibles, celui qui nécessite le plus de temps de calcul est l'extraction des composantes connexes. Ceci s'explique par le balayage pixel par pixel de l'image des documents qui ont été pour la plupart digitalisés à 400 *dpi*.

Tableau 10.4: Rapports en % du temps passé dans les primitives de reconnaissance.

Doc.	Fig.	H	L	CCX	Temps Total	Extraction CCX	Class.	Seuils	Reconnaissance Micro	Reconnaissance Macro
1	B.2	3882	2624	3917	116 *s*	25.00%	56.89%	0.00%	18.10%	0.00%
2	B.4	3738	2666	2166	43 *s*	62.79%	23.25%	0.00%	13.95%	0.00%
3	B.5	3861	2640	2486	63 *s*	42.85%	44.44%	0.00%	12.69%	0.00%
4	B.3	3760	2650	1835	38 *s*	68.42%	15.78%	0.00%	15.78%	0.00%
5	B.6	3605	2624	2166	59 *s*	42.37%	45.76%	0.00%	11.86%	0.00%
6	6.9	3466	1914	1682	29 *s*	62.06%	20.68%	0.00%	17.24%	0.00%
7	B.12	3466	1898	1839	35 *s*	48.57%	34.28%	0.00%	17.14%	0.00%
8	B.13	3450	1935	1715	40 *s*	42.50%	45.00%	0.00%	12.50%	0.00%
9	B.10	3466	1909	2647	39 *s*	46.15%	30.76%	0.00%	23.07%	0.00%
10	B.7	2821	1962	981	21 *s*	66.66%	23.80%	0.00%	9.52%	0.00%
11	B.8	3440	1893	1888	43 *s*	41.86%	46.51%	0.00%	11.62%	0.00%
12	B.9	3440	1914	1515	31 *s*	58.06%	29.03%	0.00%	12.90%	0.00%
13	B.11	3455	1914	2179	41 *s*	43.90%	39.02%	0.00%	17.07%	0.00%
14	B.1	2160	2123	425	14 *s*	85.71%	7.14%	0.00%	7.14%	0.00%
15	6.8	3324	2256	1545	34 *s*	55.88%	32.35%	0.00%	11.76%	0.00%
Moyenne		**3422**	**2195**	**1932**	**43.07** *s*	**52.80%**	**32.90%**	**0.06%**	**14.16%**	**0.08%**

- L'essentiel du temps de calcul est passé dans la classification des composantes connexes. Dans notre système, ceci n'est pas trop important puisque la classification est réalisée par apprentissage sur deux ou trois pages ou blocs typiques des documents traités.

- Pour ce qui est de l'estimation des seuils métriques, le temps de calcul est insignifiant comparé aux autres primitives. Ils ont été, pour la plupart, estimé à 0 seconde dans notre environnement de test. Aussi comme pour la classification des CCX, les seuils peuvent être estimés une fois pour toute par un apprentissage.

- Le temps de calcul pour la reconnaissance des microstructures comprend celui de l'estimation des seuils métriques réalisée au cours des différents traitements. Néanmoins, ce temps reste raisonnable comparé à celui de l'extraction des composantes connexes qui, comme la reconnaissance des microstructures, est un traitement incompressible.

- Le temps de calcul de la reconnaissance des macrostructures comprend celui de l'extraction des rectangles structurants, de la segmentation en régions et de la découpe en blocs. Ce temps souvent estimé à 0 seconde est insignifiant comparé à celui de la reconnaissance des microstructures. Cela s'explique essentiellement par le fait que l'extraction des rectangles structurants a eu lieu à partir de l'ensemble des microstructures, et non des composantes connexes.

Pour accroître sensiblement l'efficacité globale de reconnaissance, les chiffres de ce tableau indiquent qu'il faudra commencer par améliorer la méthode d'extraction de CCX puis celle de classification.

Conclusion

En résumé, plus qu'un système fermé de reconnaissance automatique, notre prototype de reconnaissance se présente sous la forme d'un environnement de reconnaissance de structures physiques

constitué d'outils de segmentation variés. Il s'agit des primitives de découpe hiérarchique, d'étiquetage d'entités physiques, de classification des composantes connexes et d'estimation statistique des paramètres nécessaires à l'analyse d'un document. Il ressort de l'évaluation présentée dans ce chapitre que, sur l'ensemble des documents traités, les primitives que nous avons développées ont donné de bons résultats.

Chapitre 11

Conclusion générale

11.1 Bilan

Notre objectif dans cette thèse a été de prouver par des réalisations qu'il est possible d'effectuer la reconnaissance de la structure physique des documents composites au moyen d'une approche uniforme, fondée sur l'analyse des espaces et ceci, sans impliquer un système de reconnaissance optique de caractères. En effet, les techniques de segmentation usuelles rencontrées dans la littérature présentent, à notre avis, deux lacunes importantes.

1. S'il était possible de trouver des techniques spécialisées pour traiter des documents textuels, des expressions mathématiques, des tableaux ou encore des blocs graphiques, il était en revanche plus difficile d'intégrer celles-ci dans une approche d'analyse uniforme pour la reconnaissance des documents composites. En effet, l'utilisation de ces différentes techniques exigeait un filtrage manuel des blocs pouvant perturber le bon fonctionnement des techniques utilisées.

2. La fiabilité des techniques dépendait avant tout d'une bonne estimation des seuils métriques servant soit à la découpe d'une entité en sous-entités plus homogènes, soit à la fusion des entités en une entité de niveau hiérarchique plus élevé. Le problème de cette estimation qui nécessite un minimum de connaissances typographiques, n'était pas traité.

Ces deux limites ont été à l'origine de la thèse que nous avons défendue dans ce document et qui devrait contribuer à la résolution du problème posé. A cet effet, nous avons commencé par une étude des règles typographiques en vigueur dans le formatage des documents. Cette étude a débouché, d'une part, sur un ensemble de règles de production pour guider la reconnaissance des microstructures et, d'autre part, sur un langage de description de macrostructures génériques servant à la reconnaissance des macrostructures spécifiques.

11.1.1 Classification des composantes connexes

Une part de la fiabilité de notre système de reconnaissance des microstructures est due à une classification préalable des composantes connexes. Cette dernière a été réalisé au moyen d'une technique d'agglomération itérative. Le classement est guidé par les modèles appris lors d'un apprentissage automatique des composantes connexes contenues soit dans les documents d'une même classe, soit dans les pages d'un même document spécifique. Cet apprentissage n'a de sens que s'il y a une constante dans les fontes utilisées. Lorsqu'il est possible, l'apprentissage permet de diminuer considérablement le temps de calcul nécessaire à l'analyse des documents dont les composantes connexes sont conformes aux modèles de composantes connexes appris.

11.1.2 Reconnaissance des microstructures

L'étude systématique que nous avons menée sur l'aspect graphique usuel des microstructures nous a conduit à établir des règles de production traduisant la structure graphique usuelle de ces dernières. Deux approches ont été développées pour la reconnaissance des microstructures. La première, basée sur une technique de découpe hiérarchique, est guidée par des rectangles structurants. Cette approche a montré trois limites importantes : (1) une forte sensibilité aux inclinaisons, (2) une faiblesse dans l'analyse des structures complexes comme, par exemple, un graphique et (3) la difficulté à trouver des critères pertinents pour l'étiquetage des découpes hiérachiques calculées. Ces limites ont été à l'origine de la seconde approche qui, elle, suit une stratégie d'analyse mixte dans laquelle l'étiquetage des microstructures est basé sur une technique de fusion hiérarchique guidée par les règles de production régissant l'aspect graphique usuel des microstructures.

Comme tout système basé sur des règles ou sur la recherche de régularité dans une structure, notre approche peut être influencée par des incohérences pouvant subvenir dans l'aspect graphique des microstructures. Ces irrégularités peuvent provenir d'origines diverses :

- le bruit provenant de l'acquisition ou de la digitalisation,
- les irrégularités dans le formatage,
- les dégradations provenant des photocopieuses,
- les mises en page fantaisistes favorisées par les traitements de texte qui font de tout auteur un apprenti typographe,

Une partie de ces irrégularités a pu être gommée grâce aux règles de production que nous avons définies pour régir l'aspect graphique usuel des microstructures. Il s'agit en particulier des bruits entaché aux mots.

La critique majeure que l'on peut formuler à l'encontre de notre approche est qu'elle ne permet pas la remise en cause des solutions partielles. En revanche, elle présente l'avantage d'être incrémentale, puisque la construction de chaque niveau, dans la hiérarchie d'une microstructure, est prise en charge par des primitives spécialisées : par exemple, la segmentation en lignes ou en mots, l'extraction des composantes connexes ou encore la classification des composantes connexes. Cet avantage favorise la modélisation des primitives en processus spécialisés, pour leur intégration dans une plateforme de reconnaissance multi-agents ou parallèle. Une collaboration entre les deux approches de reconnaissance (découpe et fusion) a été nécessaire pour l'extraction des structures matricielles qui caractérisent la structure graphique des tableaux. Elle consiste à déterminer les cellules par la méthode de fusion et les structures matricielles par la méthode de découpe lors de la segmentation en blocs des régions qui a lieu durant la reconnaissance des macrostructures. Nous avons prouvé la fiabilité de cette approche par le biais des résultats présentés.

11.1.3 Reconnaissance des macrostructures

Contrairement aux microstructures, il n'existe aucune règle universelle régissant l'aspect graphique des macrostructures de document. En effet, Il n'est pas possible de trouver deux revues (ou journaux) différentes avec une même maquette de page, quand bien même, à l'intérieur de ces revues, les blocs textuels ont la même structure graphique. En conséquence, pour la reconnaissance des macrostructures, nous avons estimé nécessaire de disposer de connaissances caractérisant la classe de documents à analyser. L'acquisition de cette connaissance pouvait être effectuée de deux manières :

1. soit par apprentissage sur des échantillons de documents appartenant à une même classe,
2. soit par l'intermédiaire d'une description de la classe des documents à traiter.

Notre choix s'est porté sur la deuxième solution qui, à notre avis, est aussi une bonne préparation pour la première solution, car l'expertise acquise dans la description des documents est indispensable pour la modélisation d'un système d'apprentissage de macrostructures génériques.

Pour la description des macrostructures physiques génériques, nous avons défini un nouveau langage qui tient son originalité dans le fait qu'il permet de décrire un document non pas, par rapport à son contenu, mais plutôt par rapport aux espaces inoccupés communément appelés *fond*. Ce langage permet de modéliser le fond des documents appartenant à une même classe en un réseau de séparateurs génériques servant à délimiter les régions génériques de la classe. Ainsi, la reconnaissance de la macrostructure d'un document commence par sa segmentation en régions et finit par une découpe hiérarchique de ces régions en blocs. La segmentation en régions, guidée par une description de la macrostructure générique du document à traiter, revient à déterminer la position exacte des séparateurs. La reconnaissance des macrostructures est poursuivie par la segmentation des régions en blocs et de l'identification des tableaux au moyen des rectangles structurants. L'évaluation de cette approche a donné de très bons résultats sur les documents traités, autant au niveau de la fiabilité que de l'efficacité.

11.1.4 Estimation des seuils métriques

Le deuxième problème posé par les techniques usuelles de segmentation, en l'occurence celui de l'estimation des seuils métriques nécessaires pour leur bon fonctionnement, a trouvé une solution dans l'analyse statistique de la distribution des espaces. Il s'agit de l'estimation des seuils métriques de fusion et de découpe qui a pu être réalisée, par apprentissage, à partir d'une analyse statistique des espaces inoccupés dans les documents traités. A cet effet, nous avons développé une fonction d'estimation générale pour l'estimation de tous les seuils métriques utilisés dans notre système.

Sur l'ensemble des documents traités, les primitives développées dans cette thèse ont donné de bons résultats, ce qui est de bon augure pour les perspectives d'avenir.

11.2 Perspectives

Pour rendre plus conviviale l'utilisation du prototype que nous avons développé, une suite nécessaire aux travaux présentés dans cette thèse consisterait à réaliser un éditeur graphique pour assister la saisie de la description des macrostructures physiques génériques. L'idéal, pour affranchir l'utilisateur de la saisie de telles descriptions, serait de réaliser un système capable d'inférer la description de la macrostructure commune à un ensemble de documents, à partir d'une analyse des rectangles structurants extraits d'échantillons de documents spécifiques représentatifs.

L'architecture des systèmes de reconnaissance classiques est souvent séquentielle. Les différentes étapes de traitement, en général, indépendantes les unes des autres, sont exécutées les uns après les autres. Cette séquentialité des traitements n'est évidemment pas intrinsèque à la nature du problème posé par la reconnaissance des documents. Pour rompre avec de telles architectures, nous préconisons une modélisation des primitives de segmentation présentées dans cette thèse en processus spécialisés et ceci, dans l'optique de favoriser leur intégration dans un système de reconnaissance basé sur une plateforme multi-agents ou parallèle.

Bibliographie

[1] Jacques André, Dominique Decouchant, Vincent Quint, and Hélène Richy, 'Vers un atelier pour les documents structurés', Technical report, IRISA, Publication interne No 715, (1993).

[2] Vincent Quint and Irène Vatton, 'Making strucured documents active', Technical report, INRIA-IMAG, (1993). A paraître dans EP-odd Vol 7, No 1, March 1994.

[3] ISO., 'Text and office systems-standard generalized markup language', *Informtion Processing, ISO/DIS 8879*, (1986).

[4] V. Joloboff, 'Trends and standards in document representation', in *Proceedings of the International Conference on Text Precessing And Document Manipulation*, ed., Cambridge University Press, pp. 107–124, (1986).

[5] A. Bhushan and M. Plass, 'The interpress page and description language', *Computer*, 72–77, (1986).

[6] B. Reid, 'Postscript and interpress: A comparison', *ARPANET Laserlovers Distribution*, **1**, (March 1985).

[7] A. L. OAKLEY and A. C. NORRIS, 'Page description languages: development, implementation and standardization', *Electronic Plublishing*, **1**(2), 79–96, (1988).

[8] Inc. Adobe Systems, *PostScript Language Reference Manual. Reading, Mass.*, Addison - Wesley, 1985. Ce livre de référence donne une description complète du langage de formatage PostScript.

[9] Inc. Adobe Systems, *PostScript Language Tutorial and Cookbook. Reading, Mass.*, Addison -Wesley, 1985. Ce livre complete le livre de référence par des examples bien choisis.

[10] Adobe Systems Incorporated, *PostScript Language*, Addison-Wesley Company, Inc., 1986.

[11] S. L. Herring, 'What to know about page description languages', *Canada Data Systems*, **18**(6), 72–76, (1986).

[12] Tim Bienz, Richard Cohn, and Inc. Adobe Systems, *Portable Document Format, Reference Manual*, Addison-Wesley, 1993. Ce livre de référence donne une description complète de PDF.

[13] G. Kronert, 'International standard for an office document architecture model', *Journal of Information Science*, **10**, 69–78, (1985).

[14] Paul Baim, 'Adobe acrobat report', *baim@harpo.aaec.com*, (1993).

[15] Inc. Apple Computer, 'Opendoc–one architectures fits all', *http://www.cilabs.org/*, (8 May 1995).

[16] the OpenDoc for Windows Team., 'Opendoc technical white paper', *http://www.cilabs.org/*, (17 March 1995).

[17] Inc. Apple Computer, 'Opendoc white paper', *WWW*, (1993).

[18] The OpenDoc Design Team, *OpenDoc Technical Summary*, Apple Computer Inc., 23 March 1994.

[19] Heater Brown, 'Standards for structured documents', *The Computer Journal*, **32**(6), 505–514, (1989).

[20] Dominique Decouchant and Vincent Quint, 'A structured approach to cooperative editing', Technical report, INRIA-IMAG, (1994).

[21] S. Kahan, T. Pavlidis, and H. S. Baird, 'On the recognition of printed character of any font or size', *IEEE Trans. PAMI*, **9**(2), (1987).

[22] Julien Anigbogu, *Reconnaissance de textes imprimés Multifontes à l'aide de Modèles Stochastiques et Métriques*, Master's thesis, Université Nancy 1, 1992.

[23] A. Zramdini and R. Ingold, 'Optical font recognition from projection profiles', *Proceedings of the Third International Conference on Raster Image and Digital Typography*, 249–260, (1994). Darmstadt, Germany.

[24] J. Wezka, 'A survey of threshold selection technics', *Computer Vision, Graphics and Image Processing*, **7**, 259–265, (1979).

[25] J. S. Lee, 'Digital image smoothing and the sigma filter', *Computer Vision, Graphics and Image Processing*, **24**(2), 255–269, (1983).

[26] L. T. Watson, K. Arving, and R. W. Ehrich, 'Extraction of lines and regions from grey tone line drawing images', in *7th International Conference on Pattern Recognition*, pp. 493–507, (1984).

[27] M. Aubert, *Système de binarisation optimale de documents*, Ph.D. dissertation, Institut National Polytechnique de Grenoble, 1991.

[28] Stuart C. Hinds, James L. Fisher, and Donald P. D'Amato, 'A document skew detection method using run-length encoding and hough transform', *IEEE*, 464–468, (1990).

[29] H. S. Baird, 'The skew angle of printed documents', in *Society of Photographic Scientists and Engineers*, volume 40, pp. 21–24, (1987).

[30] W. Postl, 'Detection of linear oblique structures ond skew scan in digitized documents', in *Int'l Conf. on Pattern Recognition*, volume 8, pp. 687–689, Paris, (1986).

[31] J. P. Trincklin, *Conception d'un système d'analyse de documents*, Ph.D. dissertation, Université de Franche-Conté, Besançon, 1984.

[32] Eric Andres, 'Cercles discrètes et rotations disrcètes', in *Premier colloque de géométrie discrète en imagerie : fondements et applications*, Strasbourg, France, (Sept. 1991).

[33] Hr. Arabnia and M. A. Oliver, 'Arbitrary rotation of raster images with simd machine architectures', *Computer Graphics Forum*, **6**(1), 3–11, (Jan. 1987).

[34] J. Francon and JP. Reveilles, 'De l'imagerie à géométrie discrète', Technical report, Strasbourg, Rapport No 90/11, (Sept. 1990).

[35] Toshiyuki Sakai, 'A history and evaluation of document information processing', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 336–340, Tsukuba Science City, Japon, (October 1993).

[36] Yuan Y. Tang and Ching Y. Suen, 'Document structures: A survey', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 336–340, Tsukuba Science City, Japon, (October 1993).

[37] K.Y. Wong, R.G. Casey, and F.H. Whal, 'Document analysis system', *IBM J. Res. Dev.*, **26**(6), 647–656, (1982).

[38] Takashi Saitoh, Michiyoshi Tachikawa, and Toshifumi Yamaai, 'Document image segmentation and text area ordering', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 336–340, Tsukuba Science City, Japon, (October 1993).

[39] Antoine AZOKLY, Abdelwahab ZRAMDINI, and Rolf INGOLD, 'Reconnaissance de structures physiques de documents composites', in *CNED'92, Traitement de l'écriture et des documents*, ed., Abdel BELA ID, pp. 30–39. BIGRE, (Juillet 1992).

[40] T. Akiyama and I. Masuda, 'A method for document-image segmentation based on projection profiles, stroke densities and circumscribed rectangles', *Systems and Computers in Japan*, **18**(4), 101–111, (1987).

[41] Rolf Ingold, *Une nouvelle approche de la lecture optique intégrant la reconnaissance des structures de documents*, Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, Suisse, 1988.

[42] M. Grira, R.-P. Bonvin, and P. Gottraux, 'Segmentation structurelle de documents avec images', Technical report, Laboratoire d'Informatique Théorique, Ecole Polytechnique Fédérale de Lausanne, rapport No 79, (Juin 1991).

[43] G. Nagy and S. Seth, 'Hierarchical representation of optical scanned documents', in *7th ICPR*, pp. 347–349, Montreal, Canada, (1984).

[44] G. Nagy, S. Seth, and S. D. Stoddard, 'Document analysis with an expert system', in *Pattern Recognition in Practice II*, pp. 147–159, Amsterdam, (1986).

[45] O. T. Akindele A. and Belaïd, 'Page segmentation by segment tracing', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 341–344, Tsukuba Science City, Japon, (October 1993).

[46] Henry S. BAIRD, S. E. Jones, and S. J. Fortune, 'Image segmentation by shape-directed covers', in *Proc., 10th ICPR*. Atlantic City, NJ, (June 1990).

[47] Henry S. BAIRD, 'Background structure in document images', in *Advances in Structural and Syntactic Pattern Recognition*, Machine Perception and Artificial Intelligence - Vol. 5, pp. 253–269. University of Bern, Switzerland, Word Scientific, (August 1992).

[48] Masayuki Akamoto and Makoto Takahashi, 'A hybrid page segmentation method', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 743–748, Tsukuba Science City, Japon, (October 1993).

[49] Mukkai Krishnamoorthy and Gearge Nagy, 'Syntactic segmentation and labeling of digitized pages from technical journals', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(7), 737–747, (July 1993).

[50] F. Esposito, D. Malerba, G. Semeraro, E. Annese, and G. Scafuro, 'An experimental page layout recognition system for office document automatic classification: An integeted approach for inductive generalization', in *Proc., 10th ICPR*. Atlantic City, NJ, (1990).

[51] James L. Fisher, Stuart C. Hinds, and Donald P. D'Amato, 'A rule-based system for document image segmentation', *IEEE*, 567–572, (1990).

[52] Hiromichi Fujisawa and Yasuaki Nakano, 'A top-down approach to the analysis of document images', in *Structured Document Image Analysis*, pp. 99–114. Springer-Verlag, (1992).

[53] Mahesh Viswanathan, 'Analysis of scanned documents - a syntactic approach', in *Structured Document Image Analysis*, pp. 115–136. Springer-Verlag, (1992).

[54] Xialong Hao, Jason T. L. Wang, and Peter A. Ng, 'Nested segmentation: An approach for layout analysis in document classification', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 319–322, Tsukuba Science City, Japon, (October 1993).

[55] Alan Conway, 'Page grammars and page parsing. a syntactic approach to documnent layout recogntion', in *ICDAR 93, First International Conference On Document Analysis and Recognition*, pp. 761–764, Tsukuba Science City, Japon, (October 1993).

[56] Antoine AZOKLY and Rolf INGOLD, 'A language for document generic layout description and its use for segmentation into regions', Technical Report 94-24, Institute of Informatics, University of Fribourg, (1994).

[57] Lawrence O'Gorman, 'The document spectrum for bottom-up layout analysis', *IEEE Trans. on Patter Analysis and Machine Intelligence*, **15**(10), 1162–1173, (1993).

[58] Hase and Hoshino, 'Segmentation method of document images by two-dimensional fourier transform', *Trans. IECE Japan*, **J67-D**(9), 1044–1051, (Sept. 1984).

[59] Ittner D. J. and H. S. Baird, 'Language-free layout analysis', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 336–340, Tsukuba Science City, Japon, (October 1993).

[60] D. Wang and S. N. Srihari, 'Classification of newspaper images blocks using texture analysis', *Computer Vision, Graphics, and Image Processing*, **47**(3), 327–352, (Sept. 1989).

[61] A. Belaïd and O. T. Akindele, 'A labeling approach for mixed document blocks', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 749–752, Tsukuba Science City, Japon, (October 1993).

[62] Kuo-Chin Fan, Chi-Hwa Liu, and Yuan-Kai Wang, 'Segmentation and classification of mixed text/graphics/image documents', *Pattern Recognition Letters*, **15**, 1202–1209, (1994).

[63] Masayuki Okamoto and Bin Miao, 'Recognition of mathematical expressions by using the layout structures of symbols', in *ICDAR 91, First International Conference On Document Analysis and Recognition*, pp. 242–250, Saint-Malo, France, (August 1991).

[64] Masayuki Okamoto and Akira Miyazawa, 'An experimental implementation of a document recognition system for papers containing mathematical expressions', in *Structured Document Image Analysis*, pp. 36–51. Springer-Verlag, (1992).

[65] Hashim M. Twaakyondo and Masayuki Okamoto, 'Structure analysis and recognition of mathematical expressions', in *ICDAR 95, Third International Conference On Document Analysis and Recognition*, pp. 430–437, Montréal, Canada, (August 1995).

[66] H. Kojima and T. Akiyama, 'Table recognition for automated document entry system', *SPIE, High Speed Inspection Architectures, Barcoding, and Character Recognition*, **1384**, 285–292, (Sept. 1990).

[67] Surekha Chandran and Rangachar Kasturi, 'Structural recognition of tabulated data', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 743–748, Tsukuba Science City, Japon, (October 1993).

[68] Toyohide Watanabe, Quin Luo, and Noboru Sugie, 'Toward a practical document understanding of table-form documents: Its framework and knowledge representation', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 743–748, Tsukuba Science City, Japon, (October 1993).

[69] X. Lin, S. Shimotsuji, M. Minoh, and T. Sakai, 'Efficient diagram understanding with characteristic pattern detection', *Computer Vision, Graphics, and Image Processing*, **30**, 84–106, (1985).

[70] T. Pavlidis, 'A vectorizer and feature extractor for document recognition', *Computer Vision, Graphics, and Image Processing*, **35**(1), 111–127, (1986).

[71] R. Kasturi, S. T. Bow, W. El-Masri, J. Shah, J.R. Gattiker, and U.B. Mokate, 'A system for interpretation of line drawings', *IEEE Transaction on PAMI*, **12**(10), 978–992, (1990).

[72] A. Belaıd and K. Tombre, 'Analyse de documents : de l'image à la sémantique', in *CNED'92, Traitement de l'écriture et des documents*, ed., Abdel BELA ID, pp. 3–29. BIGRE, (Juillet 1992).

[73] Rolf INGOLD, *Structures de documents et lecture optique: une nouvelle approche*, collection Méta, Presses Polytechniques Féderales, 1989.

[74] Tao Hu, *New Methods for Robust and Efficient Recognition of the Logical Structures in Documents*, Ph.D. dissertation, Institut d'Informatique, Universite de Fribourg Suisse, 1994.

[75] Yannick Chenevoy, *Reconnaissance structurelle de documents imprimés : études et rélisations*, Ph.D. dissertation, Institut National Polytechnique de Lorraine, 1992.

[76] Afcet, IRISA-INRIA, and TELECOM, eds. *Proc. First International Conference On Document Analysis And Recognition, Vol. 1, 2*, Saint-Malo, France, September 1991. This is a full PROCEEDINGS entry.

[77] Institute of Electrical and Inc. Electronics Engineers, eds. *Proc. Second International Conference On Document Analysis And Recognition*, Tsukuba Science City, Japan, October 1993. This is a full PROCEEDINGS entry.

[78] IEEE Computer Society Press, ed. *Proc. Third International Conference On Document Analysis And Recognition, Vol. 1, 2*, Montréal, Canada, August 1995. This is a full PROCEEDINGS entry.

[79] Jacques André, 'Font metrics', in *Visual and Technical Aspects of Type*, Roger D. Hersch, (1993).

[80] Peter Karow, *Digital Formats for Typefaces*, URW Verlag, 1987.

[81] Vincent QUINT and Irène VATTON, 'Grif : an interactive system for structured document manipulation', in *Text Processing and Document Manipulation*, ed., J.C. van Vliet, 200–213, Cambridge Unversity Press, (1986).

[82] David Collier, *Collier's Rules of Desktop Design And Typography*, Addison-Wesley, 1991.

[83] Richard J. BEACH, 'Tabular typography conputer science laboratory', in *Text Processing and Document Manipulation*, pp. 18–33. University of Nottigham, Cambridge University Press, (April 1986).

[84] C. Vanoirbeek, *Une Modélisation de Documents Pour le Formatage*, Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 1988.

[85] Hulrburt, *The Grid: A Modular System for the Design in Production of Newspapers, Magazines, and Books.*, Van Nostrand Reinhold, 1978.

[86] Roger C. PARKE, *Looking good in print*, Ventana Press, Chapell Hill NC, 1988.

[87] Vincent QUINT, *Une approche de l'édition structurées des documents*, Master's thesis, Université scientifique, technologique et médicale de Grenoble, 1987.

[88] Richard RUBINSTEIN, *Digital Typography, An Introduction to Type and Composition for Computer System Design*, Addison-Wesley Company, Inc., 1988.

[89] Christian Mengelt, 'Visual aspects of type', in *Visual and Technical Aspects of Type*, Roger D. Hersch, (1993).

[90] James Hartley, 'The layout of computer-based text', in *Computers and Typography*, Rosemary Sassoon, (1993).

[91] J. F. Allen, 'Maintaining knowledge about temporal intervals', *Commun, ACM*, **26**, 832–843, (1983).

[92] Katsuhiko Itonori, 'Table structure recognition based on textblock arrangement and ruled line position', in *ICDAR 93, International Conference On Document Analysis and Recognition*, pp. 743–748, Tsukuba Science City, Japon, (October 1993).

[93] Antoine AZOKLY and Rolf INGOLD, 'A language for document generic layout description and its use for segmentation into regions', in *ICDAR 95, International Conference On Document Analysis and Recognition*, pp. 1123–1126, Montréal, Canada, (August 1995).

[94] Scoot A. Boorman and Donald C. Olivier, 'Metrics on spaces of finite trees', *Journal of Mathematical Psychology*, **10**, 26–59, (1973).

Annexe A

Langage de description des macrostructures physiques

A.1 Grammaire : description des classes de documents

GenericVolumeDef ::= VOLUME Identifier IS
DefaultUnitDef
PageWidthDef
PageHeightDef
[GenericLanguageDef]
GenericPageDef { GenericPageDef }
END

DefaultUnitDef ::= UNIT "=" UnitDef

UnitDef ::= MM | CM | INCH

PageWidthDef ::= WIDTH "=" UnsignedExprDef

PageHeightDef ::= HEIGHT "=" UnsignedExprDef

GenericLanguageDef ::= LANGUAGE "=" LanguageDef

LanguageDef ::= FRENCH | ENGLISH | GERMAN

GenericPageDef ::= PAGE Identifier IS
{ HSeparatorDef | VSeparatorDef }
LayerDef { LayerDef }
END

LayerDef ::= LAYER Identifier IS
{ HSeparatorDef | VSeparatorDef }
RegionDef { RegionDef }
END

HSeparatorDef ::= HSEP Identifier IS "("
PositionDef ","
SizeDef ","
LeftVSeparatorIdentifier ","
RightVSeparatorIdentifier ","
[SeparatorKindDef]
")"
[SUBSTITUTE HSeparatorIdentifier]

VSeparatorDef ::= VSEP Identifier IS "("
PositionDef ","
SizeDef ","
TopHSeparatorIdentifier ","
BottomHSeparatorIdentifier ","
[SeparatorKindDef]
")"
[SUBSTITUTE VSeparatorIdentifier]

PositionDef ::= StaticPositionDef | FloatingPositionDef

StaticPositionDef ::= ExprDef

FloatingPositionDef ::= ["+" | "-"] "[" UnsignedExprDef "," UnsignedExprDef "]"

SizeDef ::= UnsignedExprDef

SeparatorKindDef ::= SPACE |
(THREAD | DOUBLE | DASH)
[LARGE | SMALL
[CENTER | LEFT | RIGHT]]

ExprDef ::= ["+" | "-"] UnsignedExprDef

UnsignedExprDef ::= *natural integer* [UnitDef]

RegionDef ::= REGION Identifier IS "("
LeftVSeparatorIdentifier ","
RightVSeparatorIdentifier ","
TopHSeparatorIdentifier ","
BottomHSeparatorIdentifier
["," CompositeBlocksDef] ["," DominantLetterSizeDef] ["," NColumnDef]
")"

LeftVSeparatorIdentifier ::= Identifier | LEFT

RightVSeparatorIdentifier ::= Identifier | RIGHT

TopHSeparatorIdentifier ::= Identifier | TOP

BottomHSeparatorIdentifier ::= Identifier | BOTTOM

CompositeBlocksDef ::= ANY | SpecificBlockDef | "{" SpecificBlockDef { "," SpecificBlockDef } "}"

SpecificBlockDef ::= TEXT | TABLE | IMAGE | GRAPHIC | FORMULA

DominantLetterSizeDef ::= NORMAL | VLARGE | LARGE | VSMALL | SMALL

NColumnDef ::= *positive integer* | MOSAIC

A.2 Grammaire : description des documents spécifiques

SpecificVolumeDef ::= VOLUME Identifier IS
ImageDirectoryDef
InstanceOfDef
[SpecificLanguageDef]
(PartDef { PartDef } | PageSetDef { PageSetDef})
END

ImageDirectoryDef ::= DIRECTORY "=" ImageDirectoryName

InstanceOfDef ::= INSTANCE OF "=" GenericVolumeIdentifier

SpecificLanguageDef ::= LANGUAGE "=" LanguageDef

PartDef ::= PART Identifier IS
PageSetDef { PageSetDef }
END

PageSetDef ::= SET Identifier IS
SpecificPageDef { SpecificPageDef }
END

SpecificPageDef ::= "(" ImageFilename "," GenericPageIdentifier
{ "," LAYER LayerIdentifier "=" NULL }
")"

A.3 Exemple : description de la classe LRD_articles

```
VOLUME LDR_articles IS
   UNIT = MM
   WIDTH = 160
   HEIGHT = 240
```

```
   LANGUAGE = ENGLISH
   PAGE LDR_garde IS
      LAYER layer IS
         HSEP hsep1 IS (10, 8, LEFT, RIGHT, SPACE)
         HSEP hsep2 IS ([20, 30], 3, LEFT, RIGHT, SPACE)
         HSEP hsep3 IS ([30, 40], 4, LEFT, RIGHT, SPACE)
         HSEP hsep4 IS ([40, 50], 3, LEFT, RIGHT, SPACE)
         HSEP hsep5 IS ([70, 120], 7, LEFT, RIGHT, line)
         HSEP hsep6 IS ([220, 230], 2, LEFT, RIGHT, SPACE)
         HSEP hsep7 IS (236, 3, LEFT, RIGHT, SPACE)
         VSEP vsep1 IS (100, 110, TOP, hsep1, SPACE)
         VSEP vsep2 IS (107, 4, hsep5, hsep6, SPACE)
         REGION Review IS (LEFT, vsep1, TOP, hsep1, TEXT, VSMALL)
         REGION No IS (vsep1, RIGHT, TOP, hsep1, TEXT, VSMALL)
         REGION Title IS (LEFT, RIGHT, hsep1, hsep2, TEXT, VLARGE)
         REGION Author IS (LEFT, RIGHT, hsep2, hsep3, TEXT, SMALL)
         REGION Affiliation IS (LEFT, RIGHT, hsep3, hsep4, TEXT, SMALL)
         REGION Introduction IS (LEFT, RIGHT, hsep5, hsep, TEXT, SMALL)
         REGION Main IS (LEFT, vsep2, hsep5, hsep6, ANY, NORMAL)
         REGION Margin IS (vsep2 , RIGHT, hsep5, hsep6, ANY, SMALL)
         REGION Correspondence IS (LEFT, RIGHT, hsep6, hsep7, TEXT, SMALL)
         REGION Reference IS (LEFT, RIGHT, hsep7, BOTTOM, TEXT, SMALL)
      END
   END
   PAGE LDR_paire IS
      HSEP hsep1 IS (4, 3, LEFT, RIGHT, SPACE)
      LAYER Principale IS
         VSEP vsep1 IS (40, 65, TOP, hsep1, SPACE)
         VSEP vsep2 IS (53, 4, hsep1, BOTTOM, SPACE)
         REGION Main IS (vsep2, RIGHT, hsep1, BOTTOM)
         REGION Margin IS (LEFT, vsep2, hsep1, BOTTOM, ANY, SMALL, 1)
         REGION No IS (LEFT, vsep1, TOP, hsep1, TEXT, VSMALL)
         REGION Author IS (vsep1, RIGHT, TOP, hsep1, TEXT, VSMALL)
      END
      LAYER Secondaire IS
         HSEP hsep2 IS (-[10, 220], 2, LEFT, RIGHT, SPACE) SUBSTITUTE hsep1
         HSEP hsep3 IS ([20, 240], 2, LEFT, RIGHT, SPACE) SUBSTITUTE BOTTOM
         REGION Figure IS (LEFT, RIGHT, hsep2, hsep3, {TABLE, GRAPHIC}, SMALL)
      END
   END
   PAGE LDR_impaire IS
      HSEP hsep1 IS (4, 3, LEFT, RIGHT, SPACE)
      LAYER Principale IS
         VSEP vsep1 IS (135, 46, TOP, hsep1, SPACE)
         VSEP vsep2 IS (107, 4, hsep1, BOTTOM, SPACE)
         REGION Main IS (LEFT, vsep2, hsep1, BOTTOM)
         REGION Margin IS (vsep2, RIGHT, hsep1, BOTTOM, ANY, SMALL)
         REGION No IS (vsep1, RIGHT, TOP, hsep1, TEXT, VSMALL)
         REGION Title IS (LEFT, vsep1, TOP, hsep1, TEXT, VSMALL)
      END
      LAYER Secondaire IS
         HSEP hsep2 IS (-[10, 220], 2, LEFT, RIGHT, SPACE) SUBSTITUTE hsep1
         HSEP hsep3 IS ([20, 240], 2, LEFT, RIGHT, SPACE) SUBSTITUTE BOTTOM
         REGION Figure IS (LEFT, RIGHT, hsep2, hsep3, {TABLE, GRAPHIC}, SMALL)
      END
   END
END
```

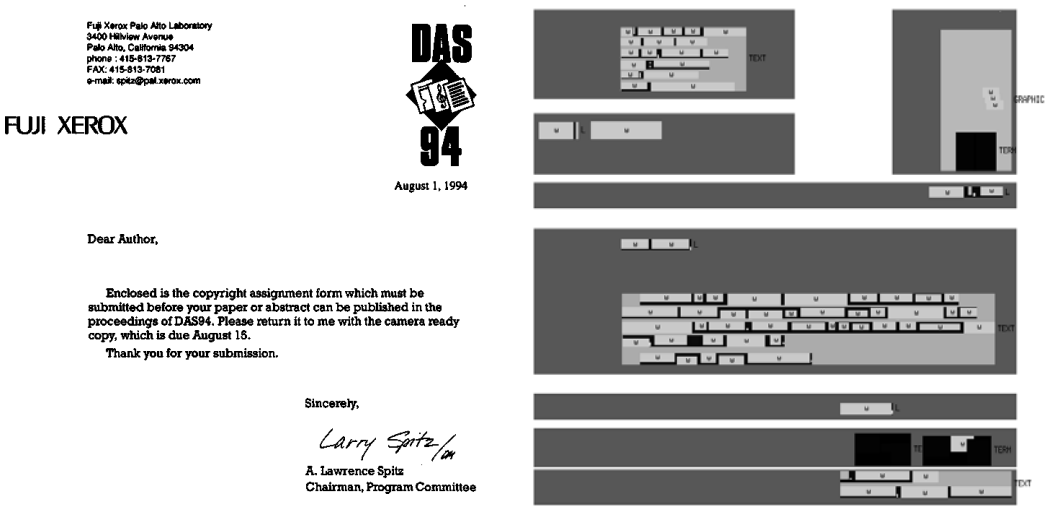
A.4 Exemple : un document spécifique de LRD_articles

```
VOLUME MICP_article IS
   DIRCTORY = Catalogue/MICP_article
   INSTANCE OF = LRD_articles
   SET article IS
      (page_1.400 , first)
      (page_2.400 , even, LAYER Superposée = NULL)
      (page_3.400 , odd )
      (page_4.400 , even)
      (page_5.400 , odd )
      (page_6.400 , even)
      (page_7.400 , odd )
      (page_8.400 , even)
      (page_9.400 , odd )
      (page_10.400, even)
      (page_11.400, odd )
```

```
      (page_12.400, even)
      (page_13.400, odd )
      (page_14.400, even)
      (page_15.400, odd )
      (page_16.400, even)
   END
END
```

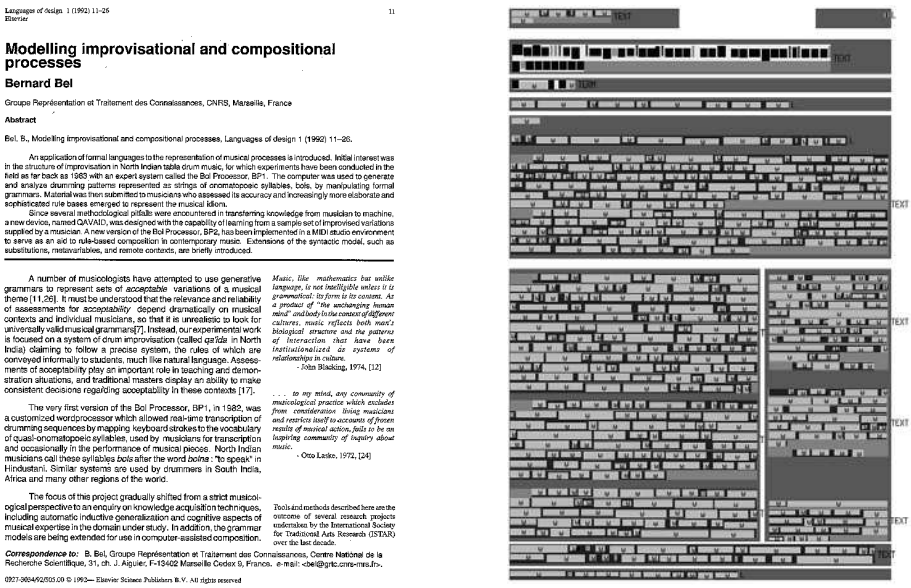
Annexe B

Autres résultats de reconnaissance



$d_y^d = 6$ $d_x^c = 8$ $d_x^m = 24$ $d_y^b = 49$

Figure B.1: Autre résultat de reconnaissance (Doc. 2).



Languages of design 1 (1992) 11-26 11
Elsevier

Modelling improvisational and compositional processes

Bernard Bel

Groupe Représentation et Traitement des Connaissances, CNRS, Marseille, France

Abstract

Bel, B., Modelling improvisational and compositional processes, Languages of design 1 (1992) 11–26.

An application of formal languages to the representation of musical processes is introduced. Initial interest was in the structure of improvisation in North Indian tabla drum music, for which experiments have been conducted in the field as far back as 1983 with an expert system called the Bol Processor, BP1. The computer was used to generate and analyze drumming patterns represented as strings of onomatopoeic syllables, bols, by manipulating formal grammars. Material was then submitted to musicians who assessed its accuracy and increasingly more elaborate and sophisticated rule bases emerged to represent the musical idioms.

Since several methodological pitfalls were encountered in transferring knowledge from musician to machine, a new device, named QAVAID, was designed with the capability of learning from a sample set of improvised variations supplied by a musician. A new version of the Bol Processor, BP2, has been implemented in a MIDI studio environment to serve as an aid to rule-based composition in contemporary music. Extensions of the syntactic model, such as substitutions, metavariables, and remote contexts, are briefly introduced.

A number of musicologists have attempted to use generative grammars to represent sets of *acceptable* variations of a musical theme [11,26]. It must be understood that the relevance and reliability of assessments for *acceptability* depend dramatically on musical contexts and individual musicians, so that it is unrealistic to look for universally valid musical grammars[7]. Instead, our experimental work is focused on a system of drum improvisation (called *qa'ida* in North India) claiming to follow a precise system, the rules of which are conveyed informally to students, much like natural language. Assessments of acceptability play an important role in teaching and demonstration situations, and traditional masters display an ability to make consistent decisions regarding acceptability in these contexts [17].

The very first version of the Bol Processor, BP1, in 1982, was a customized wordprocessor which allowed real-time transcription of drumming sequences by mapping keyboard strokes to the vocabulary of quasi-onomatopoeic syllables, used by musicians for transcription and occasionally in the performance of musical pieces. North Indian musicians call these syllables *bols* after the word *bolna* : "to speak" in Hindustani. Similar systems are used by drummers in South India, Africa and many other regions of the world.

The focus of this project gradually shifted from a strict musicological perspective to an enquiry on knowledge acquisition techniques, including automatic inductive generalization and cognitive aspects of musical expertise in the domain under study. In addition, the grammar models are being extended for use in computer-assisted composition.

Music, like mathematics but unlike language, is not intelligible unless it is grammatical: its form is its content. As a product of "the unchanging human mind" endlessly in the context of different cultures, music reflects both man's biological structure and the patterns of interaction that have been institutionalized as systems of relationships in culture.

- John Blacking, 1974, [12]

. . . to my mind, any community of musicological practice which excludes from consideration living musicians and restricts itself to accounts of frozen results of musical action, fails to be an inspiring community of inquiry about music.

- Otto Laske, 1972, [24]

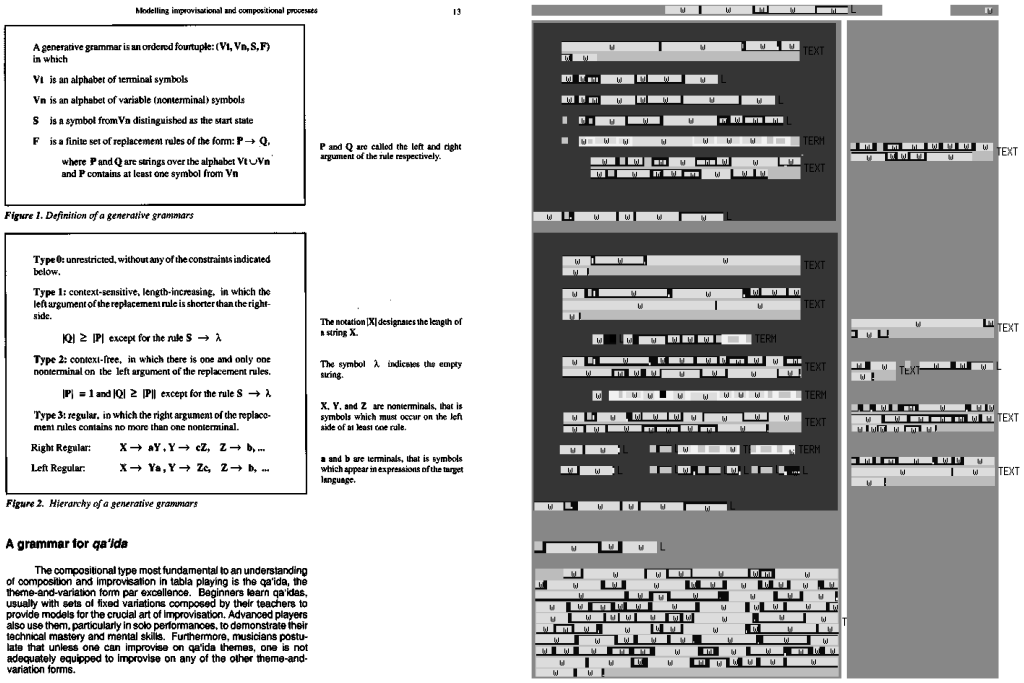
Tools and methods described here are the outcome of several research projects undertaken by the International Society for Traditional Arts Research (ISTAR) over the last decade.

Correspondence to: B. Bel, Groupe Représentation et Traitement des Connaissances, Centre National de la Recherche Scientifique, 31, ch. J. Aiguier, F-13402 Marseille Cedex 9, France. e-mail: <bel@lgrc.cnrs-mrs.fr>.

0167-9604/92/$05.00 © 1992— Elsevier Science Publishers B.V. All rights reserved

$d_y^d = 6$ $d_x^c = 9$ $d_x^m = 58$ $d_y^b = 79$

Figure B.2: Reconnaissance de documents textuels (Doc. 1).



Modelling improvisational and compositional processes 13

A generative grammar is an ordered fourtuple: (Vt, Vn, S, F) in which

Vt is an alphabet of terminal symbols

Vn is an alphabet of variable (nonterminal) symbols

S is a symbol from Vn distinguished as the start state

F is a finite set of replacement rules of the form: P → Q,

where P and Q are strings over the alphabet Vt ∪ Vn and P contains at least one symbol from Vn

P and Q are called the left and right argument of the rule respectively.

Figure 1. Definition of a generative grammars

Type 0: unrestricted, without any of the constraints indicated below.

Type 1: context-sensitive, length-increasing, in which the left argument of the replacement rule is shorter than the right-side.

$|Q| \geq |P|$ except for the rule $S \rightarrow \lambda$

Type 2: context-free, in which there is one and only one nonterminal on the left argument of the replacement rules.

$|P| = 1$ and $|Q| \geq |P|$ except for the rule $S \rightarrow \lambda$

Type 3: regular, in which the right argument of the replacement rules contains no more than one nonterminal.

Right Regular: $X \rightarrow aY, Y \rightarrow cZ, Z \rightarrow b, \ldots$

Left Regular: $X \rightarrow Ya, Y \rightarrow Zc, Z \rightarrow b, \ldots$

The notation |X| designates the length of a string X.

The symbol λ indicates the empty string.

X, Y, and Z are nonterminals, that is symbols which must occur on the left side of at least one rule.

a and b are terminals, that is symbols which appear in expressions of the target language.

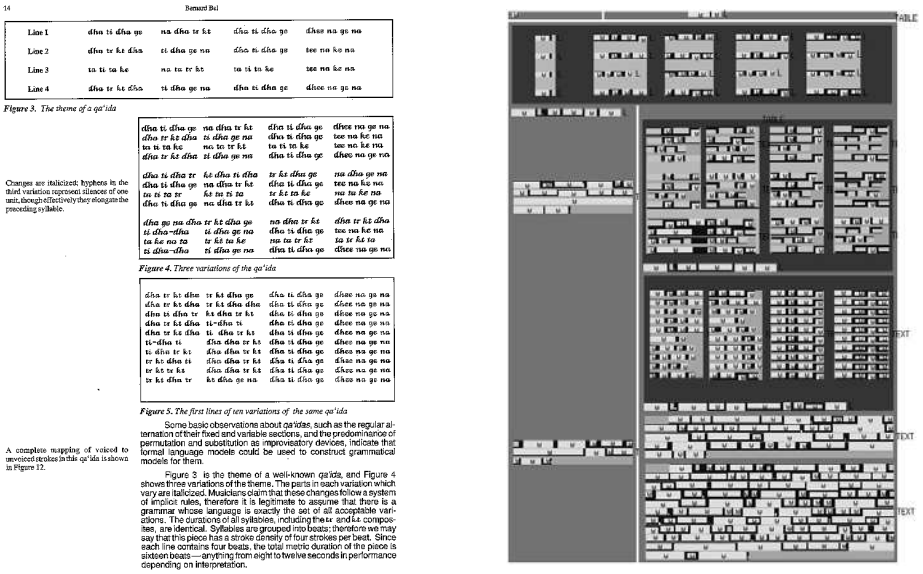
Figure 2. Hierarchy of a generative grammars

A grammar for *qa'ida*

The compositional type most fundamental to an understanding of composition and improvisation in tabla playing is the qa'ida, the theme-and-variation form par excellence. Beginners learn qa'idas, usually with sets of fixed variations composed by their teachers to provide models for the crucial art of improvisation. Advanced players also use them, particularly in solo performances, to demonstrate their technical mastery and mental skills. Furthermore, musicians postulate that unless one can improvise on qa'ida themes, one is not adequately equipped to improvise on any of the other theme-and-variation forms.

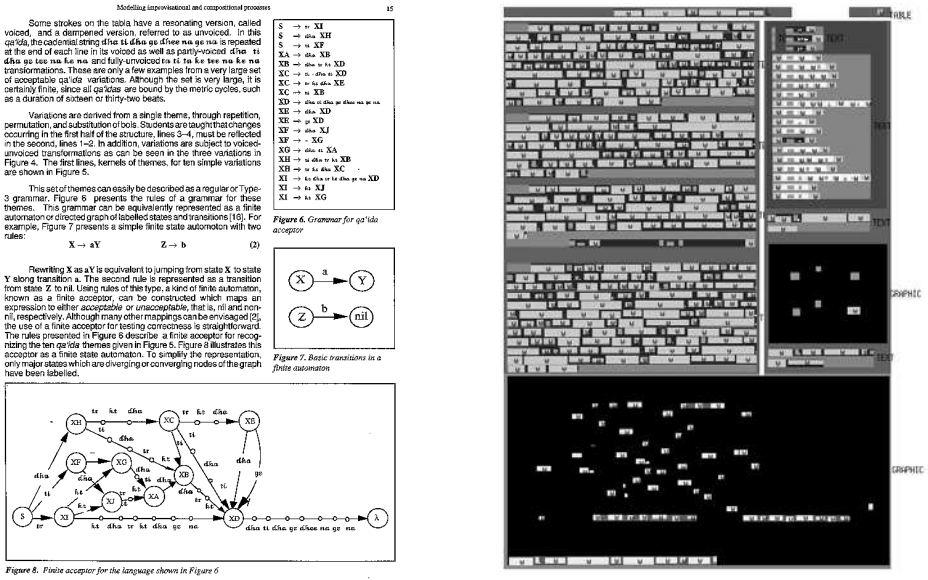
$d_y^d = 7$ $d_x^c = 5$ $d_x^m = 20$ $d_y^b = 71$

Figure B.3: Autre résultat de reconnaissance (Doc. 10).



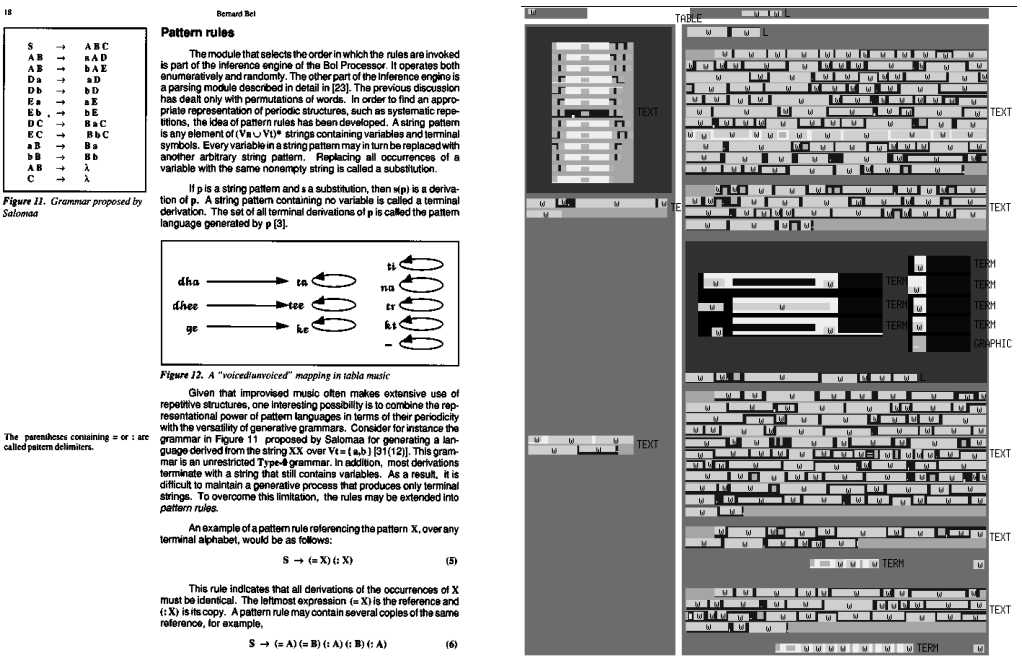
$d_y^d = 7$ $d_x^c = 10$ $d_x^m = 37$ $d_y^b = 68$

Figure B.4: Reconnaissance de tableaux (Doc. 6).



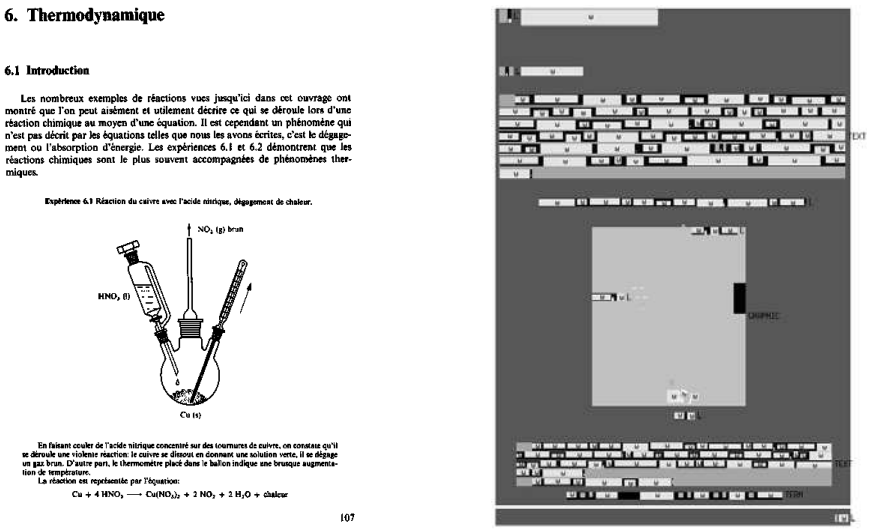
$d_y^d = 7$ $d_x^c = 9$ $d_x^m = 61$ $d_y^b = 73$

Figure B.5: Reconnaissance de blocs graphiques (Doc. 8).



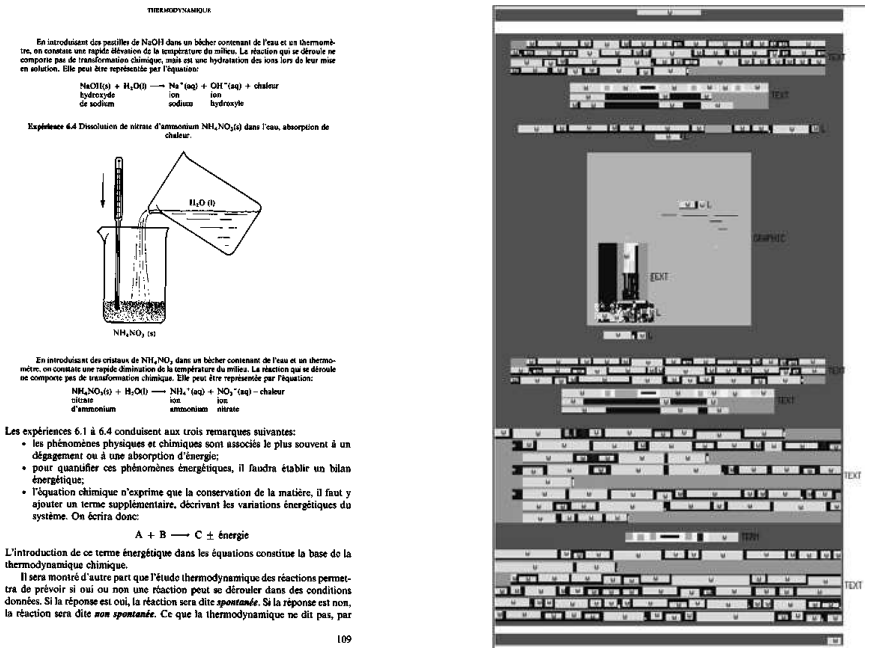
$d_y^d = 6$ $d_x^c = 9$ $d_x^m = 44$ $d_y^b = 69$

Figure B.6: Autre résultat de reconnaissance (Doc. 11).



$d_y^d = 8$ $d_x^c = 6$ $d_x^m = 35$ $d_y^b = 52$

Figure B.7: Autre résultat de reconnaissance (Doc. 12).



THERMODYNAMIQUE

En introduisant des pastilles de NaOH dans un bécher contenant de l'eau et un thermomètre, on constate une rapide élévation de la température du milieu. La réaction qui se déroule ne comporte pas de transformation chimique, mais est une hydratation des ions lors de leur mise en solution. Elle peut être représentée par l'équation:

$$NaOH(s) + H_2O(l) \longrightarrow Na^+(aq) + OH^-(aq) + \text{chaleur}$$

hydroxyde de sodium — ion sodium — ion hydroxyle

Expérience 6.4 Dissolution de nitrate d'ammonium $NH_4NO_3(s)$ dans l'eau, absorption de chaleur.

En introduisant des cristaux de NH_4NO_3 dans un bécher contenant de l'eau et un thermomètre, on constate une rapide diminution de la température du milieu. La réaction qui se déroule ne comporte pas de transformation chimique. Elle peut être représentée par l'équation:

$$NH_4NO_3(s) + H_2O(l) \longrightarrow NH_4^+(aq) + NO_3^-(aq) - \text{chaleur}$$

nitrate d'ammonium — ion ammonium — ion nitrate

Les expériences 6.1 à 6.4 conduisent aux trois remarques suivantes:

- les phénomènes physiques et chimiques sont associés le plus souvent à un dégagement ou à une absorption d'énergie;
- pour quantifier ces phénomènes énergétiques, il faudra établir un bilan énergétique;
- l'équation chimique n'exprime que la conservation de la matière, il faut y ajouter un terme supplémentaire, décrivant les variations énergétiques du système. On écrira donc:

$$A + B \longrightarrow C \pm \text{énergie}$$

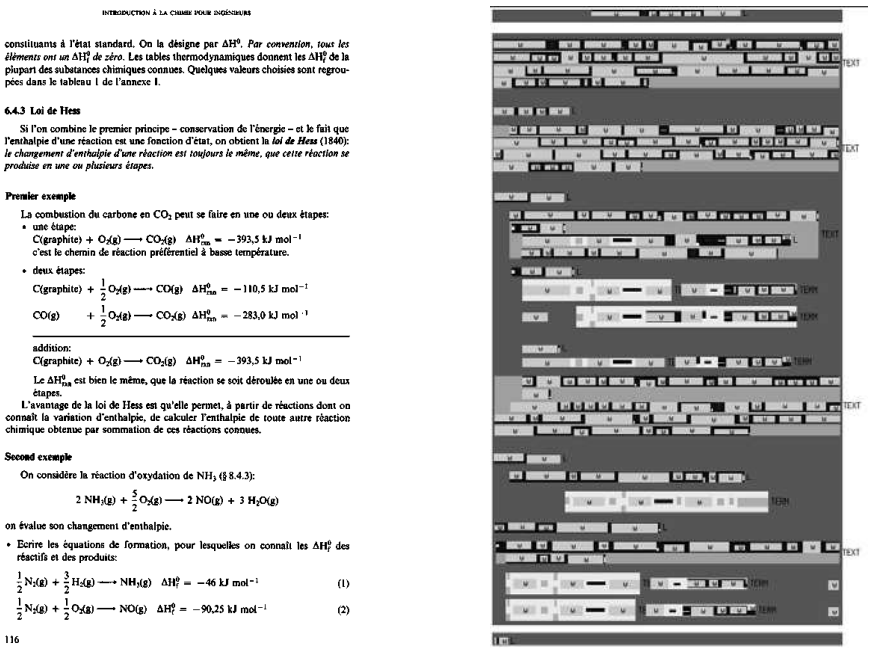
L'introduction de ce terme énergétique dans les équations constitue la base de la thermodynamique chimique.

Il sera montré d'autre part que l'étude thermodynamique des réactions permettra de prévoir si oui ou non une réaction peut se dérouler dans des conditions données. Si la réponse est oui, la réaction sera dite ***spontanée***. Si la réponse est non, la réaction sera dite ***non spontanée***. Ce que la thermodynamique ne dit pas, par

109

$d_y^d = 8$ $d_x^c = 8$ $d_x^m = 42$ $d_y^b = 73$

Figure B.8: Autre résultat de reconnaissance (Doc. 13).



INTRODUCTION À LA CHIMIE POUR INGÉNIEURS

constituants à l'état standard. On la désigne par ΔH^0. *Par convention, tous les éléments ont un* ΔH_f^0 *de zéro.* Les tables thermodynamiques donnent les ΔH_f^0 de la plupart des substances chimiques connues. Quelques valeurs choisies sont regroupées dans le tableau I de l'annexe I.

6.4.3 Loi de Hess

Si l'on combine le premier principe – conservation de l'énergie – et le fait que l'enthalpie d'une réaction est une fonction d'état, on obtient la ***loi de Hess*** (1840): *le changement d'enthalpie d'une réaction est toujours le même, que cette réaction se produise en une ou plusieurs étapes.*

Premier exemple

La combustion du carbone en CO_2 peut se faire en une ou deux étapes:

- une étape:
  $C(\text{graphite}) + O_2(g) \longrightarrow CO_2(g) \quad \Delta H^0_{\text{rxn}} = -393{,}5 \text{ kJ mol}^{-1}$
  c'est le chemin de réaction préférentiel à basse température.
- deux étapes:
  $C(\text{graphite}) + \frac{1}{2}O_2(g) \longrightarrow CO(g) \quad \Delta H^0_{\text{rxn}} = -110{,}5 \text{ kJ mol}^{-1}$
  $CO(g) + \frac{1}{2}O_2(g) \longrightarrow CO_2(g) \quad \Delta H^0_{\text{rxn}} = -283{,}0 \text{ kJ mol}^{-1}$

  addition:
  $C(\text{graphite}) + O_2(g) \longrightarrow CO_2(g) \quad \Delta H^0_{\text{rxn}} = -393{,}5 \text{ kJ mol}^{-1}$

  Le ΔH^0_{rxn} est bien le même, que la réaction se soit déroulée en une ou deux étapes.

L'avantage de la loi de Hess est qu'elle permet, à partir de réactions dont on connaît la variation d'enthalpie, de calculer l'enthalpie de toute autre réaction chimique obtenue par sommation de ces réactions connues.

Second exemple

On considère la réaction d'oxydation de NH_3 (§ 8.4.3):

$$2\,NH_3(g) + \frac{5}{2}O_2(g) \longrightarrow 2\,NO(g) + 3\,H_2O(g)$$

on évalue son changement d'enthalpie.

- Ecrire les équations de formation, pour lesquelles on connaît les ΔH_f^0 des réactifs et des produits:

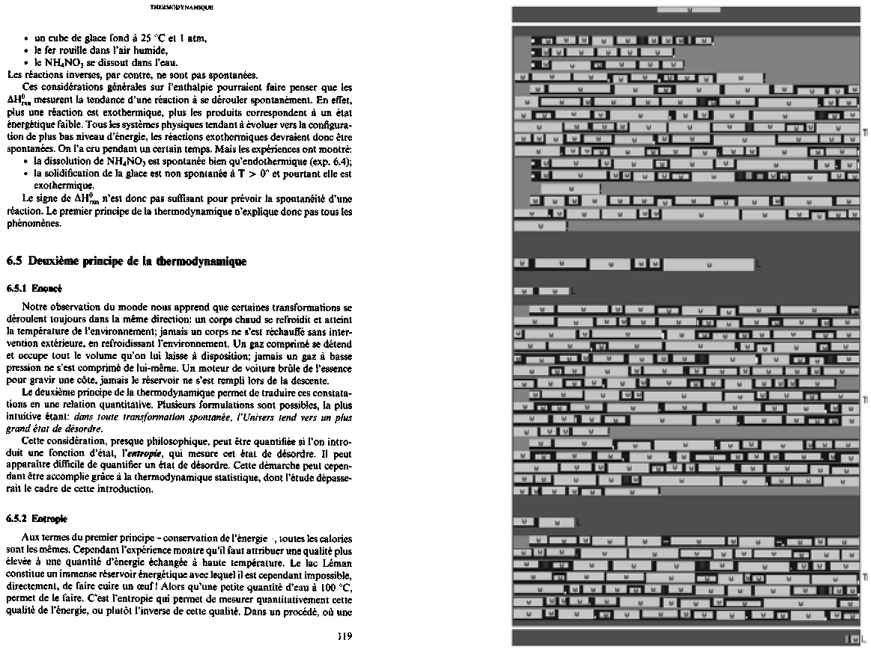
$$\frac{1}{2}N_2(g) + \frac{3}{2}H_2(g) \longrightarrow NH_3(g) \quad \Delta H_f^0 = -46 \text{ kJ mol}^{-1} \qquad (1)$$

$$\frac{1}{2}N_2(g) + \frac{1}{2}O_2(g) \longrightarrow NO(g) \quad \Delta H_f^0 = -90{,}25 \text{ kJ mol}^{-1} \qquad (2)$$

116

$d_y^d = 8$ $d_x^c = 9$ $d_x^m = 43$ $d_y^b = 73$

Figure B.9: Autre résultat de reconnaissance (Doc. 14).



THERMODYNAMIQUE

- un cube de glace fond à 25 °C et 1 atm,
- le fer rouille dans l'air humide,
- le NH_4NO_3 se dissout dans l'eau.

Les réactions inverses, par contre, ne sont pas spontanées.

Ces considérations générales sur l'enthalpie pourraient faire penser que les ΔH^0_{rxn} mesurent la tendance d'une réaction à se dérouler spontanément. En effet, plus une réaction est exothermique, plus les produits correspondent à un état énergétique faible. Tous les systèmes physiques tendant à évoluer vers la configuration de plus bas niveau d'énergie, les réactions exothermiques devraient donc être spontanées. On l'a cru pendant un certain temps. Mais les expériences ont montré:

- la dissolution de NH_4NO_3 est spontanée bien qu'endothermique (exp. 6.4);
- la solidification de la glace est non spontanée à T > 0° et pourtant elle est exothermique.

Le signe de ΔH^0_{rxn} n'est donc pas suffisant pour prévoir la spontanéité d'une réaction. Le premier principe de la thermodynamique n'explique donc pas tous les phénomènes.

6.5 Deuxième principe de la thermodynamique

6.5.1 Enoncé

Notre observation du monde nous apprend que certaines transformations se déroulent toujours dans la même direction: un corps chaud se refroidit et atteint la température de l'environnement; jamais un corps ne s'est réchauffé sans intervention extérieure, en refroidissant l'environnement. Un gaz comprimé se détend et occupe tout le volume qu'on lui laisse à disposition; jamais un gaz à basse pression ne s'est comprimé de lui-même. Un moteur de voiture brûle de l'essence pour gravir une côte, jamais le réservoir ne s'est rempli lors de la descente.

Le deuxième principe de la thermodynamique permet de traduire ces constatations en une relation quantitative. Plusieurs formulations sont possibles, la plus intuitive étant: *dans toute transformation spontanée, l'Univers tend vers un plus grand état de désordre.*

Cette considération, presque philosophique, peut être quantifiée si l'on introduit une fonction d'état, l'***entropie***, qui mesure cet état de désordre. Il peut apparaître difficile de quantifier un état de désordre. Cette démarche peut cependant être accomplie grâce à la thermodynamique statistique, dont l'étude dépasserait le cadre de cette introduction.

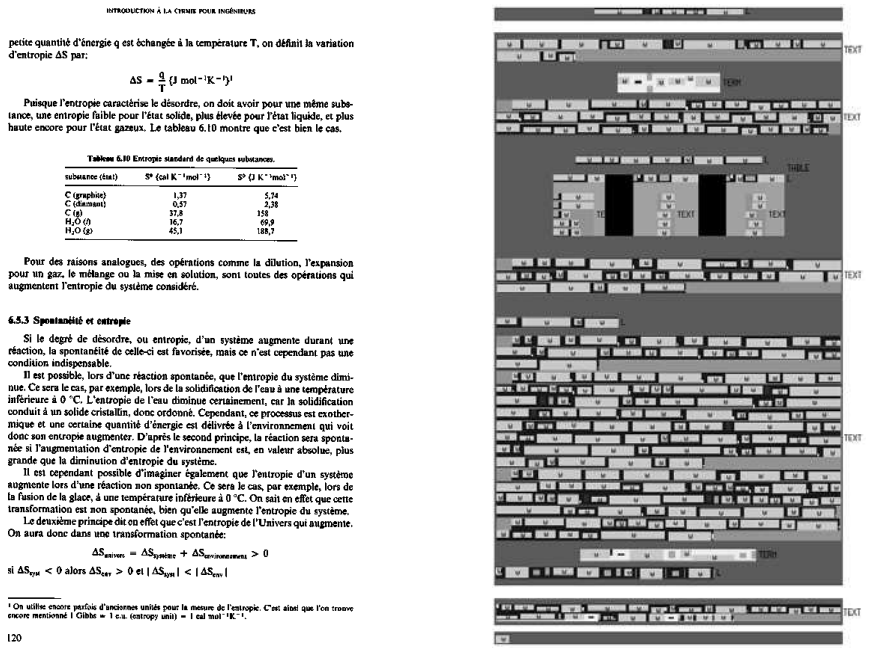
6.5.2 Entropie

Aux termes du premier principe - conservation de l'énergie -, toutes les calories sont les mêmes. Cependant l'expérience montre qu'il faut attribuer une qualité plus élevée à une quantité d'énergie échangée à haute température. Le lac Léman constitue un immense réservoir énergétique avec lequel il est cependant impossible, directement, de faire cuire un œuf ! Alors qu'une petite quantité d'eau à 100 °C, permet de le faire. C'est l'entropie qui permet de mesurer quantitativement cette qualité de l'énergie, ou plutôt l'inverse de cette qualité. Dans un procédé, où une

119

$d_y^d = 9$ $d_x^c = 7$ $d_x^m = 35$ $d_y^b = 75$

Figure B.10: Reconnaissance de documents textuels (Doc. 9).



INTRODUCTION À LA CHIMIE POUR INGÉNIEURS

petite quantité d'énergie q est échangée à la température T, on définit la variation d'entropie ΔS par:

$$\Delta S = \frac{q}{T} \text{ (J mol}^{-1}\text{K}^{-1})^1$$

Puisque l'entropie caractérise le désordre, on doit avoir pour une même substance, une entropie faible pour l'état solide, plus élevée pour l'état liquide, et plus haute encore pour l'état gazeux. Le tableau 6.10 montre que c'est bien le cas.

Tableau 6.10 Entropie standard de quelques substances.

substance (état)	S^0 (cal K^{-1} mol^{-1})	S^0 (J K^{-1} mol^{-1})
C (graphite)	1,37	5,74
C (diamant)	0,57	2,38
C (g)	37,8	158
H_2O (l)	16,7	69,9
H_2O (g)	45,1	188,7

Pour des raisons analogues, des opérations comme la dilution, l'expansion pour un gaz, le mélange ou la mise en solution, sont toutes des opérations qui augmentent l'entropie du système considéré.

6.5.3 Spontanéité et entropie

Si le degré de désordre, ou entropie, d'un système augmente durant une réaction, la spontanéité de celle-ci est favorisée, mais ce n'est cependant pas une condition indispensable.

Il est possible, lors d'une réaction spontanée, que l'entropie du système diminue. Ce sera le cas, par exemple, lors de la solidification de l'eau à une température inférieure à 0 °C. L'entropie de l'eau diminue certainement, car la solidification conduit à un solide cristallin, donc ordonné. Cependant, ce processus est exothermique et une certaine quantité d'énergie est délivrée à l'environnement qui voit donc son entropie augmenter. D'après le second principe, la réaction sera spontanée si l'augmentation d'entropie de l'environnement est, en valeur absolue, plus grande que la diminution d'entropie du système.

Il est cependant possible d'imaginer également que l'entropie d'un système augmente lors d'une réaction non spontanée. Ce sera le cas, par exemple, lors de la fusion de la glace, à une température inférieure à 0 °C. On sait en effet que cette transformation est non spontanée, bien qu'elle augmente l'entropie du système.

Le deuxième principe dit en effet que c'est l'entropie de l'Univers qui augmente. On aura donc dans une transformation spontanée:

$$\Delta S_{univers} = \Delta S_{système} + \Delta S_{environnement} > 0$$

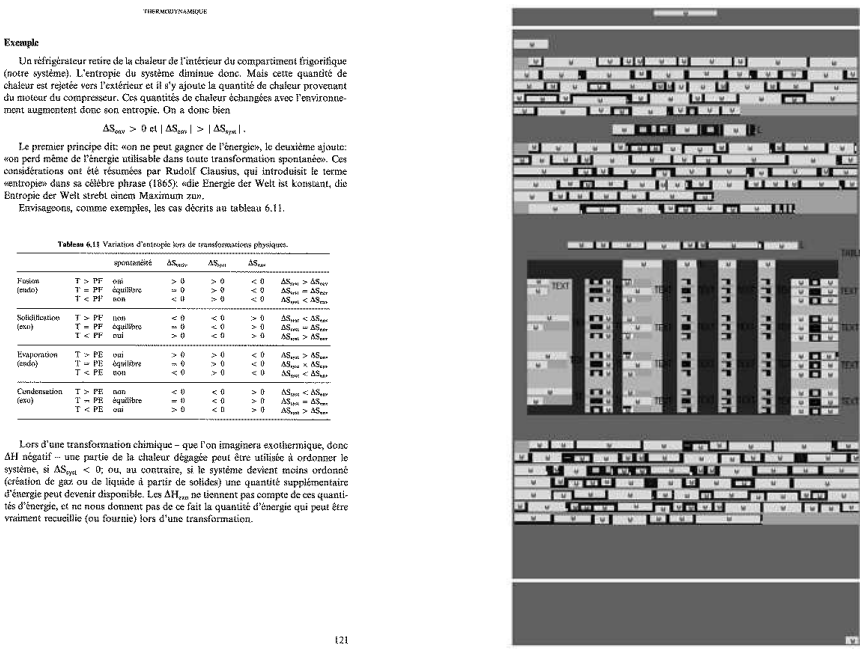
si $\Delta S_{sys} < 0$ alors $\Delta S_{env} > 0$ et $|\Delta S_{sys}| < |\Delta S_{env}|$

[1] On utilise encore parfois d'anciennes unités pour la mesure de l'entropie. C'est ainsi que l'on trouve encore mentionné 1 Clibit = 1 e.u. (entropy unit) = 1 cal mol $^{-1}$K^{-1}.

120

$d_y^d = 8$ $d_x^c = 10$ $d_x^m = 36$ $d_y^b = 74$

Figure B.11: Autre résultat de reconnaissance (Doc. 15).



THERMODYNAMIQUE

Exemple

Un réfrigérateur retire de la chaleur de l'intérieur du compartiment frigorifique (notre système). L'entropie du système diminue donc. Mais cette quantité de chaleur est rejetée vers l'extérieur et il s'y ajoute la quantité de chaleur provenant du moteur du compresseur. Ces quantités de chaleur échangées avec l'environnement augmentent donc son entropie. On a donc bien

$$\Delta S_{env} > 0 \text{ et } | \Delta S_{env} | > | \Delta S_{syst} | .$$

Le premier principe dit: «on ne peut gagner de l'énergie», le deuxième ajoute: «on perd même de l'énergie utilisable dans toute transformation spontanée». Ces considérations ont été résumées par Rudolf Clausius, qui introduisit le terme «entropie» dans sa célèbre phrase (1865): «die Energie der Welt ist konstant, die Entropie der Welt strebt einem Maximum zu».

Envisageons, comme exemples, les cas décrits au tableau 6.11.

Tableau 6.11 Variation d'entropie lors de transformations physiques.

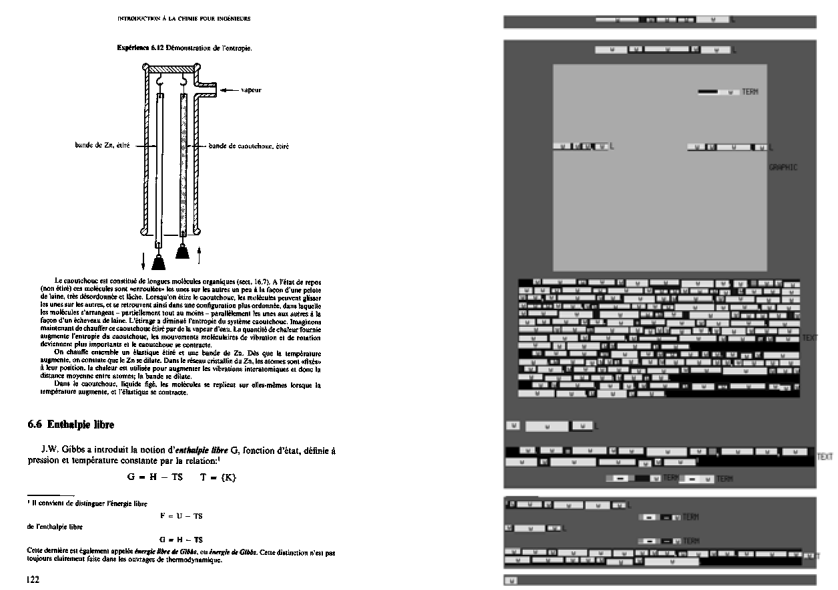
		spontanéité	$\Delta S_{env.}$	ΔS_{syst}	ΔS_{uni}	
Fusion	$T > PF$	oui	> 0	> 0	< 0	$\Delta S_{env} > \Delta S_{syst}$
(endo)	$T = PF$	équilibre	≈ 0	> 0	< 0	$\Delta S_{env} = \Delta S_{syst}$
	$T < PF$	non	< 0	> 0	< 0	$\Delta S_{env} < \Delta S_{syst}$
Solidification	$T > PF$	non	< 0	< 0	> 0	$\Delta S_{env} < \Delta S_{syst}$
(exo)	$T = PF$	équilibre	≈ 0	< 0	> 0	$\Delta S_{env} = \Delta S_{syst}$
	$T < PF$	oui	> 0	< 0	> 0	$\Delta S_{env} > \Delta S_{syst}$
Evaporation	$T > PE$	oui	> 0	> 0	< 0	$\Delta S_{env} > \Delta S_{syst}$
(endo)	$T = PE$	équilibre	≈ 0	> 0	< 0	$\Delta S_{env} \approx \Delta S_{syst}$
	$T < PE$	non	< 0	> 0	< 0	$\Delta S_{env} < \Delta S_{syst}$
Condensation	$T > PE$	non	< 0	< 0	> 0	$\Delta S_{env} < \Delta S_{syst}$
(exo)	$T = PE$	équilibre	≈ 0	< 0	> 0	$\Delta S_{env} = \Delta S_{syst}$
	$T < PE$	oui	> 0	< 0	> 0	$\Delta S_{env} > \Delta S_{syst}$

Lors d'une transformation chimique – que l'on imaginera exothermique, donc ΔH négatif – une partie de la chaleur dégagée peut être utilisée à ordonner le système, si $\Delta S_{syst} < 0$; ou, au contraire, si le système devient moins ordonné (création de gaz ou de liquide à partir de solides) une quantité supplémentaire d'énergie peut devenir disponible. Les ΔH_{syst} ne tiennent pas compte de ces quantités d'énergie, et ne nous donnent pas de ce fait la quantité d'énergie qui peut être vraiment recueillie (ou fournie) lors d'une transformation.

121

$d_y^d = 8$ $d_x^c = 8$ $d_x^m = 39$ $d_y^b = 55$

Figure B.12: Reconnaissance de tableaux (Doc. 5).



INTRODUCTION À LA CHIMIE POUR INGÉNIEURS

Expérience 6.12 Démonstration de l'entropie.

Le caoutchouc est constitué de longues molécules organiques (sect. 14.7). A l'état de repos (non étiré) ces molécules sont «emmêlées» les unes sur les autres un peu à la façon d'une pelote de laine, très désordonnée et lâche. Lorsqu'on étire le caoutchouc, les molécules peuvent glisser les unes sur les autres, et se retrouvent ainsi dans une configuration plus ordonnée, dans laquelle les molécules s'arrangent – partiellement tout au moins – parallèlement les unes aux autres à la façon d'un écheveau de laine. L'étirage a diminué l'entropie du système caoutchouc. Imaginons maintenant de chauffer ce caoutchouc étiré par de la vapeur d'eau. La quantité de chaleur fournie augmente l'entropie du caoutchouc, les mouvements moléculaires de vibration et de rotation deviennent plus importants et le caoutchouc se contracte.

On chauffe ensemble un élastique étiré et une bande de Zn. Dès que la température augmente, on constate que le Zn se dilate. Dans le réseau cristallin du Zn, les atomes sont «fixés» à leur position, la chaleur est utilisée pour augmenter les vibrations interatomiques et donc la distance moyenne entre atomes; la bande se dilate.

Dans le caoutchouc, liquide figé, les molécules se replient sur elles-mêmes lorsque la température augmente, et l'élastique se contracte.

6.6 Enthalpie libre

J.W. Gibbs a introduit la notion d'***enthalpie libre*** G, fonction d'état, définie à pression et température constante par la relation:[1]

$$G = H - TS \qquad T = (K)$$

[1] Il convient de distinguer l'énergie libre

$$F = U - TS$$

de l'enthalpie libre

$$G = H - TS$$

Cette dernière est également appelée ***énergie libre de Gibbs***, ou ***énergie de Gibbs***. Cette distinction n'est pas toujours clairement faite dans les ouvrages de thermodynamique.

122

$d_y^d = 8$ $d_x^c = 7$ $d_x^m = 32$ $d_y^b = 55$

Figure B.13: Reconnaissance de blocs graphiques (Doc. 7).

Remerciements

Je tiens à remercier toutes les personnes qui, par leur participation et leurs encouragements, m'ont permis de mener à bonne fin cette thèse.

En premier lieu, le Professeur Rolf INGOLD, mon directeur de thèse, qui m'a offert un cadre de travail des plus favorables et m'a fait bénéficier de ses conseils avisés qui ont été déterminants pour l'aboutissement de ce travail.

Le Professeur Georges STAMON, directeur du Laboratoire des Systèmes Intelligents de Perception (SIP) à UFR de Mathématiques et Informatique de l'Université de Paris V, pour l'intérêt qu'il manifeste envers mon travail en acceptant d'en être rapporteur et, également, pour la sympathie qu'il m'a témoignée lors de notre première rencontre.

Le Docteur Karl TOMBRE, chargé de recherche à l'INRIA Lorraine à Nancy, pour l'intérêt qu'il manifeste envers mon travail en acceptant d'en être rapporteur ainsi que pour ses observations pertinentes qui ont contribué à améliorer la clarté de ce mémoire.

Le Professeur Béat HIRSBRUNNER, directeur de l'Institut d'Informatique de l'Université de Fribourg (IIUF), pour la confiance qu'il m'a témoignée en m'engageant à l'IIUF en qualité d'assistant à la recherche et à l'enseignement et, également, pour la sympathie qu'il m'a toujours témoignée durant mes cinq années de service à l'IIUF.

Le Docteur Abdelwahab ZRAMDINI, compagnon de recherche que je me permettrai ici d'appeler Abdou, pour sa disponibilité et pour sa collaboration en recherche qui est des plus appréciables.

A tous mes collègues à l'IIUF, merci pour l'ambiance de travail, leur soutien et l'intérêt quotidiens qu'ils ont toujours manisfestés envers mon travail. Je pense en particulier à Monsieur Frédéric BAPST dont les observations ont également contribué à la clarté de ce mémoire.

D'autre part, j'associe à la réussite de cette thèse le Professeur Olivier BESSON, de l'Institut de Mathématiques de l'Université de Neuchâtel, pour la sympathie qu'il m'a toujours manisfestée et pour m'avoir fait bénéficier de ses conseils qui ont été déterminants dans mon choix de poursuivre ma recherche de thèse à l'Institut d'Informatique de l'Université de Fribourg.

Merci aux amis dont le soutien moral m'a permis de tenir bon durant les moments de doutes; je pense tout particulièrement au Docteur Akimou OSSE.

Ma plus profonde reconnaissance à mes parents qui m'ont donné la possibilité de faire des études et qui m'ont enseigné l'humilité et l'amour du prochain.

Ma profonde gratitude au Docteur Edmonde HELD pour son soutien inestimable et pour m'avoir ouvert les portes de sa maison.

Enfin à Dominique HELD, un merci tout particulier pour sa patience et son soutien qui m'ont donné la force nécessaire à l'aboutissement de cette thèse.

Curriculum vitae

Etat civil

Nom et prénoms	:	AZOKLY Antoine Sourou
Fils de	:	AZOKLY Pierre et HOUSSOU Céline
Date de naissance	:	14 juin 1964
Lieu de naissance	:	Cotonou, République du Bénin
Nationalité	:	Béninoise
Etat civil	:	Célibataire

Formations

Ecole primaire

Date	Ecole	Lieu	Diplôme
1971-1977	Ecole régionale d'Akpakpa	Bénin	Certificat d'Etudes Primaires

Ecole Secondaire

Date	Ecole	Lieu	Diplôme
1977-1981	CEMG Akpakpa-centre	Bénin	Brevet d'Etudes Secondaires
1981-1984	CEMG Akpakpa-centre	Bénin	BAC C avec Mention

Etudes supérieures

Date	Ecole	Lieu	Diplôme
1985-1990	Institut de Mathématiques et d'Informatique, Université de Neuchâtel	Suisse	Licence ès sciences (Informatique) avec Mention
1990-1995	Institut d'Informatique, Université de Fribourg	Suisse	préparation de cette thèse

Activités et expériences professionnelles

Date	Employeur	Lieu	Actvité
09-1991/08-1994	Institut d'Informatique, Université de Fribourg	Suisse	Assistant diplômé
09-1990/08-1991	Institut d'Informatique, Université de Fribourg	Suisse	Collaborateur scientifique
04-1990/09-1993	Alusuisse-Lonza Services SA	Chippis Suisse	Consultant
10-1989/01-1990	Institut de Mathématiques et d'Informatique, Université de Neuchâtel	Suisse	Assistant suppléant
07-1989/10-1989	Alusuisse-Lonza Services SA	Chippis Suisse	Stage (Maths appliquées)
07-1988/09-1988	Ascom Autelca	Gümligen Suisse	Stage (Informatique)
09-1984/07-1985	Ministère de l'Enseignement Supérieur	Cotonou Bénin	Prof. de Maths au Collège
09-1984/07-1985	Ministère de l'Intérieur	Cotonou Bénin	Service militaire