

Institut National des Sciences Appliquées de Lyon  
Université Lyon I

Mémoire de DEA Ingénierie Informatique

LECTURE AUTOMATIQUE DE  
PARTITIONS MUSICALES

**Jean-Yves RAMEL**  
Année 1992-1993

**LISPI - Equipe de Reconnaissance  
des Formes et Diagnostics**

## **REMERCIEMENTS**

*Je tiens, tout d'abord, à adresser mes sincères remerciements à :*

*- Monsieur le professeur H. Emptoz, directeur du Laboratoire de Reconnaissance des Formes et Diagnostics, qui malgré ses lourdes occupations, a toujours été à mon écoute afin de régler les différents problèmes rencontrés durant cette étude.*

*- Mademoiselle N. Vincent, professeur agrégé et maître de conférence à l'INSA, pour m'avoir proposé ce sujet puis encadré tout au long de mes recherches. Je la remercie surtout de m'avoir continuellement encouragé dans mes initiatives personnelles tout en m'apportant de nombreux conseils.*

*- Je remercie aussi tout les membres du Laboratoire de Reconnaissance des Formes et Diagnostics; et plus particulièrement les chercheurs qui m'ont dispensé leur aide et leurs encouragements tout au long de cette année.*

*- Je remercie enfin les étudiants de DEA avec qui j'ai passé mes cinq mois de stage.*

# Table des matières

<b><u>1. Introduction</u></b> .....	p 5
1.1. Présentation du domaine de recherche.....	p 5
1.2. Objectifs.....	p 5
<b><u>2. Comparaison avec la reconnaissance de l'écriture</u></b> .....	p 7
2.1. La structure des documents.....	p 7
- Structure physique.....	p 8
- Structure logique.....	p 9
2.2. Symboles et typologie .....	p 13
2.3. Schéma de reconnaissance .....	p 15
- Techniques de prétraitement .....	p 15
- Techniques de segmentation .....	p 17
- Techniques d'analyse.....	p 19
- Techniques de reconnaissance .....	p 20
2.4. Conclusion .....	p 22
<b><u>3. Reconnaissance de partitions : Etat de l'art</u></b> .....	p 23
3.1. Acquisition.....	p 23
3.2. Les méthodes utilisant des histogrammes.....	p 24
3.3. Les méthodes de suivi .....	p 26
3.4. Les méthodes récursives.....	p 29
3.5. Les méthodes par analyse de régions.....	p 31
3.6. Les méthodes topologiques.....	p 34
3.7. Post-traitements .....	p 36
3.8. Résultats obtenus .....	p 37

<b><u>4. Expérimentations</u></b> .....	p 38
<b>4.1. Présentation générale</b> .....	p 38
<b>4.2. Détection par calcul d'histogrammes</b> .....	p 41
- Détection des portées .....	p 41
- Détection des lignes de portées.....	p 43
- Détection des têtes de notes .....	p 44
<b>4.3. Détection par suivi de contour</b> .....	p 51
- Elimination des lignes de portées.....	p 51
- Détection des queues de notes .....	p 53
- Détection des barres de croches.....	p 55
<b>4.4. Traitement des symboles</b> .....	p 58
- Tamisage de la portée .....	p 58
- Détection et analyse.....	p 58
<b>4.5. Post-traitements</b> .....	p 62
<b>4.6. Résultats et perspectives</b> .....	p 64
<b><u>5. Conclusion</u></b> .....	p 65
<b><u>Bibliographie</u></b> .....	p 66

## **1. Introduction :**

### **1.1. Présentation du domaine de recherche :**

Depuis l'avènement de la norme M.I.D.I., permettant de faire communiquer l'ordinateur avec nombre d'instruments électroniques, la musique assistée par ordinateur a subi un essor considérable. L'usage de l'ordinateur est sorti du cadre restreint de la recherche musicale et de la publication, pour entrer dans les moeurs des compositeurs et des particuliers. Ce succès est aussi dû à la baisse des prix du matériel informatique et à l'apparition, sur le marché, de nombreux logiciels de manipulation de partitions, d'aide à la composition, ou d'aide à la synthèse sonore, ...

L'étape préalable à tous ces traitements de l'information sonore est l'acquisition des données musicales par la machine. Elle a connu, jusqu'à présent, peu de développement: le codage d'une partition passe, encore aujourd'hui, par de fastidieuses manipulations de souris ou par l'exécution du morceau désiré sur un instrument MIDI connecté à la machine (technique d'ailleurs réservée aux personnes ayant des connaissances musicales). La quantité d'information ainsi traitée ne peut évidemment pas être importante ou nécessite l'intervention d'experts, il existe donc un véritable enjeu économique.

Au même titre que la reconnaissance de caractères pour la pratique du traitement de texte, la reconnaissance automatique de partition musicale constitue une étape fondamentale susceptible d'apporter une amélioration majeure aux nouveaux systèmes musicaux. Aussi des recherches dans cette direction se poursuivent encore aujourd'hui.

Les premiers travaux ont été menés dans les années 1970, mais se sont heurtés à un manque de technologie entraînant des temps de calcul vraiment trop importants. Les recherches ont alors été interrompues, pour reprendre au Japon en 1982: dans le but de rapprocher, toujours plus, la machine de l'homme, Matsushima et Al. ont développé un système de lecture automatique de partitions musicales, destiné au robot WABOT-2 devant être capable de lire une partition puis de la jouer au piano [Roads86].

Le même module de reconnaissance sera utilisé plus tard pour l'élaboration d'un système de traduction de partitions musicales en braille [Matsushima88].

Aujourd'hui, la qualité croissante des systèmes informatiques, notamment des dispositifs d'acquisition, encourage les nouvelles investigations et devrait permettre d'aboutir à des solutions susceptibles de s'adapter aux nouveaux systèmes musicaux.

## 1.2. Objectifs :

Le plus souvent, on a tendance à comparer la reconnaissance de partitions musicales à la reconnaissance de caractères: les symboles musicaux sont assimilés à une classe particulière de caractères et l'adaptation d'un système de reconnaissance de l'écriture ne semble alors poser aucun véritable problème.

Malheureusement le problème n'est pas aussi simple qu'il paraît être, l'extraction des primitives musicales et leur interprétation posant, à ce jour, encore beaucoup de difficultés. Nous nous proposons donc d'aborder ces différents aspects selon la démarche suivante:

- dans un premier temps, en nous basant sur les connaissances acquises en reconnaissance de l'écriture, on effectuera une comparaison entre ces deux domaines, tant au point de vue des documents à traiter, qu'au point de vue des schémas de reconnaissance.

- ensuite, une étude des principales solutions apportées à la reconnaissance de partitions musicales, permettra de nous familiariser avec les principales techniques d'analyse de documents.

- une fois le bilan dégagé, la réalisation d'un "système expérimental" permettra de vérifier notre bonne compréhension des méthodes et de tester nos choix initiaux.

## **2. Comparaison avec la reconnaissance de l'écriture :**

La reconnaissance de l'écriture est un domaine de recherche largement traité dans la littérature puisque son origine remonte aux années 1900.

De même que pour la reconnaissance de caractères, le but de la reconnaissance de partitions musicales est de transformer un document écrit en une représentation facilement utilisable par une machine.

Il est donc possible que des méthodes utilisées en OCR ( Optical Character Recognition ) soient facilement adaptables à l'OMR (Optical Music Recognition ).

Une comparaison de ces deux techniques apparaît alors comme une étape préalable, indispensable à la mise en place d'un système de lecture automatique de partitions musicales.

Le but de ce chapitre est, en se basant sur les connaissances acquises depuis de longues années sur la structuration de document écrit, ainsi que sur les schémas classiques de reconnaissance de l'écriture, d'étudier si une mise en parallèle de ces deux domaines est possible.

### **2.1 La structure des documents :**

La connaissance de la structure du document à traiter est une nécessité puisqu'on devra l'exploiter durant la phase "Reconnaissance". Elle permet de définir une stratégie de lecture, de segmentation, et d'identification des entités de base.

Nous allons donc dans un premier temps étudier s'il existe des points communs entre une partition musicale et un texte écrit.

Deux niveaux de structuration sont à distinguer dans un document écrit [Boukined91], [Bélaïd92] :

- la *structure physique*, qui résulte de la mise en page.
  
- la *structure logique*, qui précède la mise en page et se rapporte plutôt au contenu.

## - Structure physique :

Tout document est doté d'une présentation personnelle destinée à faire ressortir son contenu. Pour effectuer la comparaison, nous prendrons comme document textuel de référence : la lettre.

La vision globale d'une page de texte, nous permet de déterminer les éléments principaux entrant en jeu dans sa structuration. Le respect de certaines conventions, principalement *l'espacement* et *l'alignement*, nous autorise à caractériser une page par:

- la présence de blocs de texte plus ou moins réguliers.
- des espaces entre les blocs.
- des lignes de base plus ou moins fluctuantes à l'intérieur des blocs.

On parle :

- du *calibrage* d'un document : il s'agit de l'agencement du texte de manière à équilibrer au mieux la page en tenant compte des lignes courtes, des blancs, des titres, des espacements entre les lignes et entre les mots, ...

- de la *mise en page* : elle consiste à choisir la taille de la page, la dimension des marges, le nombre de colonnes, l'interlignage, l'espace entre les mots, la justification et la police de caractères.

Différents problèmes sont liés à la présentation :

- Le premier est la variété *des espacements* et des *type de contact* entre les caractères : les mots sont espacés de manière irrégulière (à cause de la justification) et deux lignes consécutives peuvent se trouver dans une même zone, voire même se toucher (à cause des hampes et des jambages).

- L'autre grand problème de l'écriture est *l'inclinaison* des lignes, du fait des difficultés qu'elle introduit pour réaliser la segmentation.

D'un point de vue global, une partition musicale donne une impression voisine de celle décrite ci-dessus :

On retrouve les notions de calibrage et de mise en page (marges, espacements, interlignes, ...).

Néanmoins, la structure physique d'un *document musical* n'atteint jamais (ou rarement) le degré de complexité d'un texte. Elle se restreint la plupart du temps à une suite de blocs homogènes (correspondant aux portées), blocs séparés par des espaces blancs plutôt réguliers.

Le problème de l'inclinaison des lignes peut se retrouver en lecture automatique de partitions musicales. Mais, dans ce cas, l'inclinaison sera toujours due à la phase d'acquisition de l'image et apparaîtra sur l'intégrité du document de façon constante. Le redressement sera de ce fait (et du fait de la présence des lignes de portées) plus facile à mettre en oeuvre.

La structure physique d'une partition semble donc moins complexe que celle du document écrit (figures 2.1 et 2.2). Sa détermination gagnera en simplicité puisqu'elle prendra en compte moins de données.

### - Structure logique :

La structure logique concerne le contenu des documents. Pour les textes, elle peut parfois être déduite de la structure physique (par analyse du contenu des blocs). Le but de la reconnaissance est d'identifier le contenu et les relations d'enchaînement des entités logiques à partir de leur représentation physique.

Certains documents répondent à une norme telle que ODA ou SGML (normes internationales définies par l'Organisation Internationale de Standardisation en vue de permettre un échange informatique de documents), qui impose de nombreuses contraintes de structuration. Cependant la variété des entités et des styles de présentation ne permet pas de trouver directement et sans ambiguïté la nature et la position des caractères à reconnaître (c'est à dire sa structure logique).

La structure logique décrit le document par une *hiérarchie d'objets* [Bélaïd91]. Il existe deux types d'objets : les objets logiques composites (par exemple les sections) et les objets logiques de base (par exemple les paragraphes).

Prenons un exemple (figure 2.3). On associera à une lettre le contenu logique suivant :

<lettre> = <en-tête> <corps> <fin>  
<en-tête> = <date> <expéditeur> <destinataire> <objet> <titre>  
<corps> = <paragraphe>\*  
<fin> = <formule finale> <signature> <nom>

Les paragraphes du corps ont des tailles et des contenus différents, certaines entités sont optionnelles,... L'exemple montre que dans un texte, même simple, les objets logiques peuvent avoir des contenus différents ou même de types hétérogènes (changement de police de caractères, ...) et se trouver à des emplacements variables.

Etudions maintenant le contenu logique d'une partition musicale (figure 2.4):

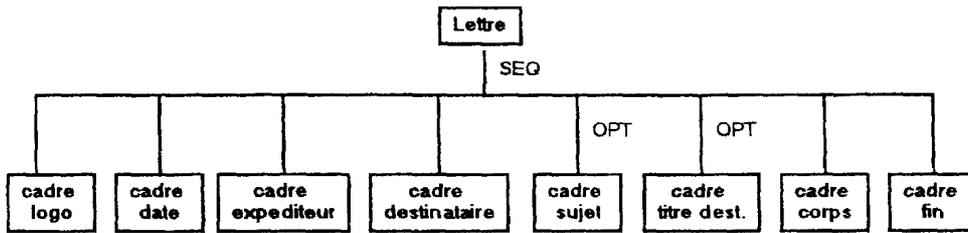
<partition> = ( <commentaire> <portée>)\* <commentaire>

Un système de lecture de partitions ne s'intéresse qu'aux portées : l'entité Portée devient alors objet logique de base unique :

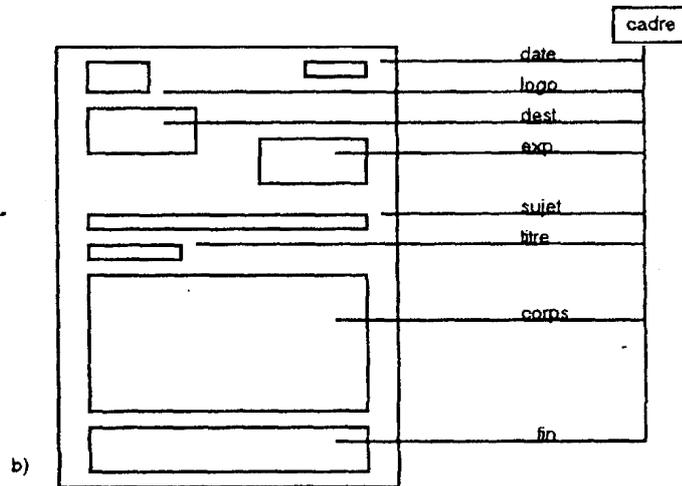
<partition> = <portée>\*

Une portée a, de plus, un aspect constant, connu à l'avance, ainsi qu'une position verticale fixe à l'intérieur de la page.

On peut donc dire que la structure logique d'une partition musicale est connue à l'avance. Ce qui n'est pas le cas pour les textes.

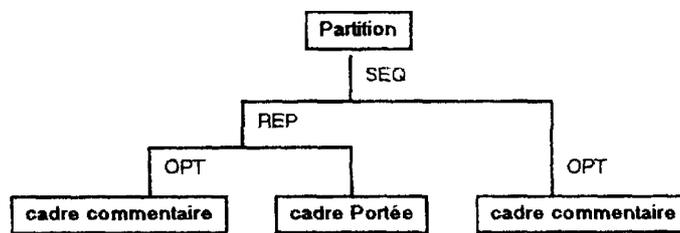


a)

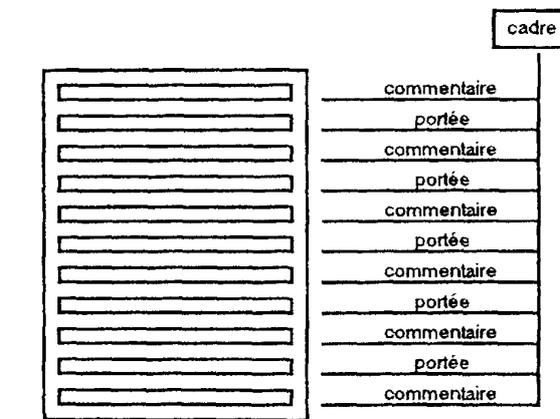


b)

figure 2.1 : a) Structure physique d'une lettre  
b) Exemple de structure



a)



b)

figure 2.2 : a) Structure physique d'une partition  
b) Exemple de structure

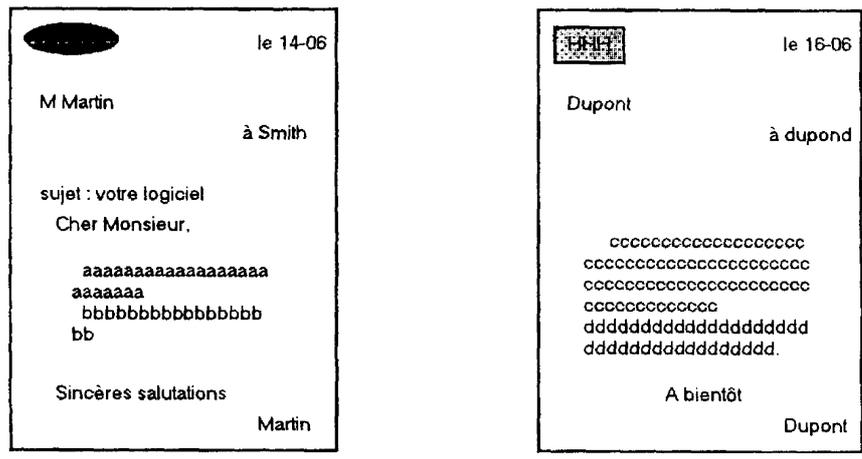
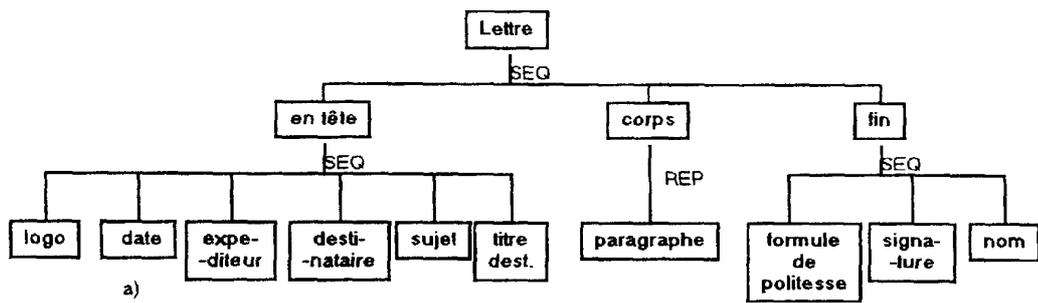


figure 2.3 : a) Structure logique d'une lettre  
b) Exemples de structure

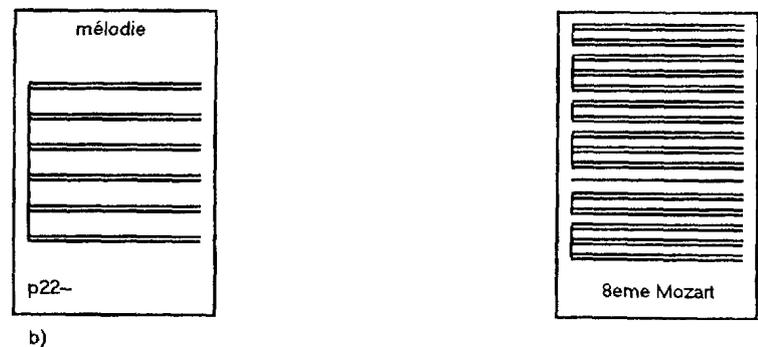
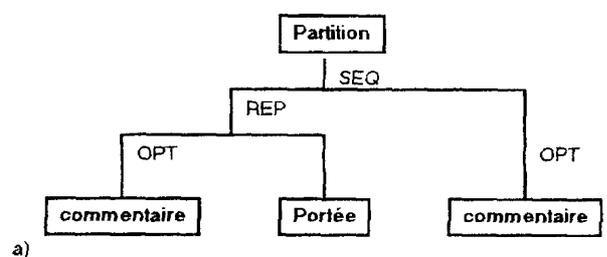


figure 2.4 : a) Structure logique d'une partition  
b) Exemples

\$

## 2.2. Symboles et typologie :

Plaçons nous maintenant au niveau des symboles pour, dans un premier temps, analyser leurs formes, puis étudier les règles syntaxiques, lexicales ou sémantiques qui régissent leur enchaînement.

Des travaux de classification ont été réalisés pour les caractères d'imprimerie, il en résulte certaines définitions que l'on peut essayer d'appliquer aux symboles musicaux [Bélaïd91] :

-les *dimensions* : la *largeur* et la *hauteur* des caractères sont variables d'un signe à l'autre (ex. : i, m, #, .

-la *chasse* comprend en plus de la largeur, l'espace entre les caractères. On ne retrouve pas cette notion pour les signes musicaux.

-le *dessin* : il caractérise la forme et l'épaisseur du symbole (a, **a**, *a*).

-la *normalisation* et la *classification*: elles ont été définies pour restreindre et répertorier les différentes familles de caractères.

Ces particularités ont permis de définir des familles de caractères appelées *fontes*. Un système est dit monofonte s'il ne traite qu'une seule fonte à la fois, il est dit multifontes s'il est capable de reconnaître un mélange de quelques fontes préalablement apprises. Enfin un système sera dit omnifontes s'il reconnaît les symboles de toutes les fontes sans les avoir véritablement appris. On se rapproche alors du manuscrit.

Les caractères manuscrits ont souvent des formes variables dues aux *distorsions* du contour, aux *variations de style* ou à de mauvais *raccordements*, se qui rend leur traitement difficile.

La nature des symboles musicaux ne permet de faire la distinction qu'entre l'imprimé et le manuscrit. Une partition musicale manuscrite est généralement largement aussi difficile à analyser qu'un texte (les ellipses deviennent de petits traits épais, les connexités ne sont plus respectées, ...).

Une étude, plus approfondie, des signes utilisés en notation musicale, montre qu'ils peuvent être divisés en deux catégories [Martin89] : Les *signes iconiques* et les *signes symboliques*.

- Les signes iconiques sont caractérisés par le lien existant entre leur forme et leur signification. Ainsi, une note est interprétée différemment selon sa constitution à partir de primitives élémentaires (ellipses, barres, crochets).

La durée des notes est fonction du nombre de crochets dont elles sont munies (croche, double-croche,...). Leur "hauteur" est déterminée par la position de la primitive "Ellipse" qu'elles contiennent. Ces assemblages pouvant prendre une multitude de formes, la reconnaissance consistera alors à attribuer une classe à chaque description de la forme à partir de ces primitives.

- les autres signes peuvent être qualifiés de symboles, leur forme est purement conventionnelle et l'ensemble des primitives élémentaires qui les composent ne peut subir aucune modification (ex: #, .

Cette distinction de nature n'existe pas pour les signes alphabétiques. Tout caractère a un aspect constant, il peut donc être considéré comme un signe symbolique même s'il est décomposable en primitives plus élémentaires (tiges, arcs, bras, ...).

L'extraction de ces traits caractéristiques n'aura rien à voir avec celle des signes musicaux compte tenu du nombre de constituants et des variations qu'ils peuvent subir: une note manuscrite est encore composée d'ellipses et de barres, tandis que des caractères pris dans deux fontes différentes comportent parfois peu de points communs.

Quant à l'entité *mot*, il est possible de la comparer avec l'entité *note*, c'est à dire porter le mot au niveau des signes symboliques.

Un mot est une juxtaposition linéaire de lettres: le caractère devient composant élémentaire.

Le mot est l'*entité lexicale* minimale nécessaire à la compréhension d'un texte, et en ce sens peut être comparé à une note dont la cohérence (en terme de primitives) peut aussi être vérifiée.

Une suite de mots comme une suite de notes engendre une phrase devant respecter des règles *syntaxiques* et *sémantiques*. Le respect de ces dernières aidera à la Reconnaissance.

Toutefois, il est à noter que lors de la lecture rapide d'une phrase ou d'un mot, l'oubli d'un ou de deux caractères ne constitue pas une gêne importante. Un musicien adoptera une démarche opposée, et veillera, au contraire, à ne négliger aucune note de la partition.

De plus, il ne faut pas perdre de vue la différence de niveau qui existe entre *le mot* (composé de lettres donc de signes symboliques) et *la note*, signe iconique (composé de primitives); remarque qui nous amène à stopper ici notre comparaison.

## 2.3. Schéma de Reconnaissance :

Comme nous l'avons vu, nous disposons, dans le cas de partitions musicales, d'une image très structurée, comportant des symboles a priori connus et composés de primitives simples. Leur ordonnancement obéit de plus à des règles syntaxiques, lexicales et sémantiques aussi précises qu'en OCR.

L'architecture d'un système capable d'analyser de telles images semble se rapprocher du schéma classique utilisé en reconnaissance de l'écriture :

### - Techniques de prétraitement [Bélaïd91] :

Pour les images en niveau de gris, l'étape primordiale est la *binarisation*. Il s'agit en fait d'une première sélection des informations utiles.

Il existe deux types de méthodes :

- Les méthodes qui appliquent un seul seuil de sélection des points sur l'image globale.

- Les méthodes qui analysent localement l'image pour adapter le seuillage aux différentes parties de l'image : l'algorithme de Watson et al [Watson84]. calcule dans un voisinage la moyenne  $m$  des points dont la valeur est supérieure à un seuil bas. Si la valeur du pixel est supérieure à  $m$  ou à un seuil haut, le pixel est considéré comme noir. Les points restants constituent le fond.

Il faut ensuite *extraire le texte* de l'ensemble du document :

- cette localisation peut être faite par étude des transitions zone noire-zone blanche sur chaque ligne de l'image. Cette technique, présentée par Wang et Srihari, est comparable à une analyse de texture appropriée à la détection de caractères.

Il est aussi possible d'utiliser un algorithme de lissage : le R.L.S.A. (Run Length Smoothing Algorithm) qui va noircir les plages blanches de longueur inférieure à un seuil  $S$ , comprises entre deux pixels noirs. On obtient alors des blocs noirs continus à l'emplacement de chaque ligne.

Par exemple, pour la ligne suivante :

0001100011111110011000000011111100001100

le lissage horizontal (avec  $S = 4$ ) donne:

0001111111111111111000000011111111111100

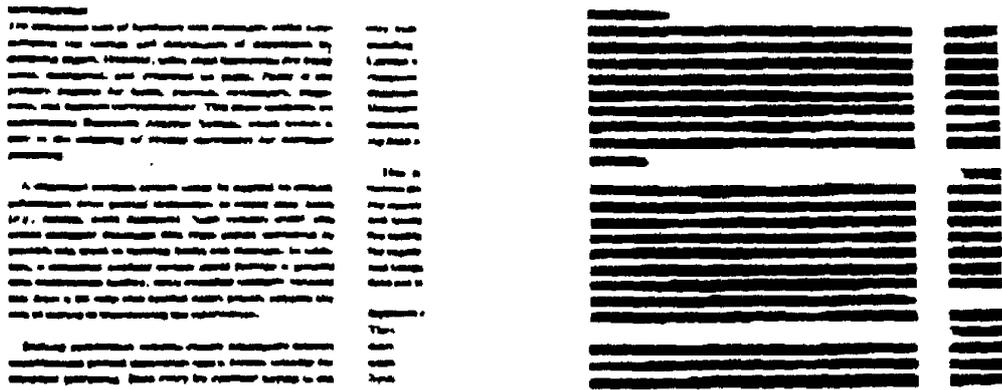


Figure 2.5 : exemple de lissage.

Wong, Casey et Wahl ([Wong82] cité par [Bélaïd91]) appliquent ce lissage horizontalement et verticalement sur l'image, puis font un ET logique entre les deux matrices obtenues pour avoir l'image des composantes connexes (figure 2.5).

Il convient aussi de *redresser le texte* afin de disposer de lignes horizontales. La procédure classique consiste à suivre les lignes de texte pour en déduire une direction. Une fois l'inclinaison estimée, il est possible d'appliquer une rotation à l'image, ou alors de translater seulement les composantes connexes pour redresser uniquement les lignes sans faire subir de déformations aux caractères (méthode de Baird).

L'image peut comporter des bruits apparus lors de la phase d'acquisition (trous, barbules, ...) qu'il est parfois nécessaire d'éliminer par *lissage* : des opérations locales appelées nettoyage et bouchage permettent de résoudre ce problème, par étude du voisinage d'un point. Les masques sont décrits en utilisant les conventions suivantes: 1 pour un point de la forme, 0 pour un point du fond, X pour un point indifférent:

- Pour égaliser les contours et boucher les trous internes de la forme, on lui ajoute des points, centres des configurations suivantes:

Exemples de masques de bouchage :	1 1 1	X 1 1
	1 0 1	0 0 1
	1 1 1	X 1 1

- Pour supprimer les points noirs isolés ou situés à l'extrémité des contours, on enlève les points, centres des masques suivants:

Exemples de masques de nettoyage :	0 0 0	X 0 0
	X 1 0	1 1 0
	0 0 0	X 0 0

En vue de simplifier l'analyse de l'image, on peut opérer une *squelettisation* : réduction de l'image des objets, en tentant de garder leurs propriétés (figure 2.6). On dénombre trois grandes classes d'algorithmes : ceux à critères *topologiques*, ceux à critères *métriques* et ceux dérivés de la *vectorisation* :

- les algorithmes à critères topologiques sont basés sur une suppression itérative des points inessentiels, le long des frontières de la forme, par des opérations booléennes.

- les algorithmes à critères métriques s'efforcent de conserver les points du centre de la forme, c'est à dire à égale distance des frontières. On définit la distance d'un point  $x$  à un ensemble de points  $A$  par :  $d(x,A) = \min_{y \in A} d(x,y)$ .

Avec  $B$  frontière de la forme, le squelette  $S$  est donné par:

$$S = \{ x / \exists y, z \in B, y \neq z : d(x,B) = d(x,y) = d(x,z) \}$$

- les algorithmes dérivés de la vectorisation effectuent un codage des segments noirs et blancs de l'image balayée ligne par ligne, pour ensuite construire le squelette.

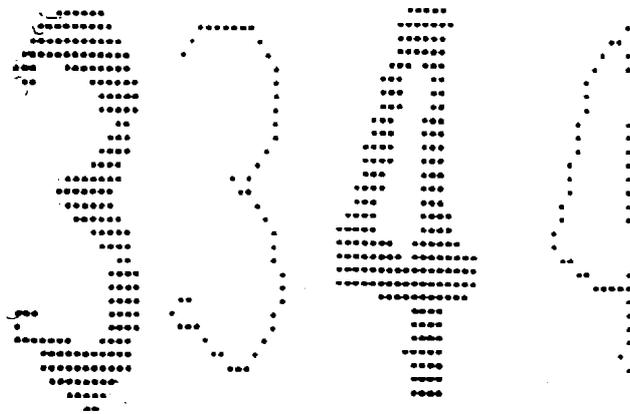


Figure 2.6 : exemples de squelettes, d'après [Marthon79].

### - Techniques de segmentation :

Le problème de la segmentation physique d'un texte en blocs, puis des blocs en lignes, puis des lignes en mots, puis des mots en signes peut être formulé en terme de recherche de composantes connexes [Boukined91]. A chaque étape, il faut, bien sûr, redéfinir la connexité :

- La première méthode d'analyse consiste à définir la connexité en terme de composantes 8-connexes, puis de distance entre signes, puis entre mots, puis entre lignes pour former les blocs. On parle alors d'*analyse ascendante*. Il est possible d'utiliser le R.L.S.A. (vu précédemment) pour regrouper les composantes noires en blocs.

- L'*analyse descendante* correspond au mouvement de l'observateur qui se rapproche progressivement du document.

Une première démarche consiste à rechercher les grandes composantes connexes vides pour en déduire les blocs pleins. Ceux-ci peuvent, à nouveau, être segmentés de manière récursive [Chauvet91].

Une variante de cette méthode peut être effectuée (sur des documents ayant un format spécifique) en se basant sur l'étude des *histogrammes* calculés sur les lignes et les colonnes de l'image.

- L'utilisation d'une *stratégie mixte*, combinant les deux méthodes ci-dessus, permettra de profiter des avantages de l'une et de l'autre [Ingold92].

La *transformée de Hough* permet de détecter facilement les droites en changeant d'espace de représentation, et peut être utilisée pour décomposer un texte comportant des lignes (blanches et noires) et des colonnes (blanches).

Il est aussi possible de procéder par lissage ou par rectangulation pour regrouper les composantes noires en blocs [Boukined91] [Ingold92].

Il s'agissait jusqu'à présent d'une segmentation de bas niveau (macro-segmentation) plutôt destinée à extraire les mots. La segmentation de haut niveau (micro-segmentation) traite de la décomposition du mot en chacun de ces éléments constitutifs. Les techniques courantes sont l'*histogramme vertical* sur chacune des colonnes graphiques, l'extraction de *composantes connexes*, ou l'*analyse de profil* inférieur ou supérieur (figure 2.7). Les techniques d'histogrammes apportent des informations sur les lignes, les mots et les lettres d'un texte imprimé ou manuscrit, l'extraction de composantes connexes permet d'isoler des signes dans les textes imprimés tandis que l'analyse de profil aidera au traitement du manuscrit.



Figure 2.7 : exemples d'histogrammes et de profils inférieurs.

### - Techniques d'analyse :

L'analyse consiste à extraire les traits caractéristiques des symboles. On distingue souvent les *méthodes structurelles* et les *méthodes fonctionnelles* [Miclet84] [Simon84].

Les premières décomposent le symbole en primitives élémentaires plus simples à détecter: "tronc" ( I ), "arc" ( B ), "bras" ( E ), "virage" ( S ) ...

Les secondes décrivent l'objet par un ensemble de mesures mathématiques et géométriques discriminantes: moments, allongements, rectangle d'encadrement, ...

Il est possible de n'étudier que certains endroits stratégiques de la forme (observation de points) pour déterminer la présence de traits caractéristiques. On effectue ensuite un *calcul de corrélation* par rapport au modèle du symbole pour déterminer s'il y a correspondance. Les limites de ce procédé résident dans la rigidité des modèles conçus à partir de points fixes.

Ce principe peut être étendu à la comparaison directe du symbole avec des masques: chaque masque (élément du dictionnaire) est déplacé sur l'image et l'on cherche le maximum de corrélation (les masques sont binaires et peuvent correspondre, par exemple, aux différentes orientations d'une ligne droite).

Une mesure de corrélation peut être définie comme la dissemblance entre le masque M et la forme F:

$$D(M,F) = \sum_{i,j} d(F,M,i,j)$$

$$\text{avec : } d(F,M,i,j) = 1 \text{ si } M(i,j) = F(i,j),$$

$$d(F,M,i,j) = 0 \text{ sinon.}$$

Au lieu de chercher à détecter l'ensemble de la forme, on peut chercher à localiser certaines parties caractéristiques: *particularités locales* comme les

segments, croisements, extrémités,... qui seront utilisées pour la reconnaissance. Cette extraction peut être faite, soit par masque à position fixe (dans l'image du caractère), soit par masque à position variable (déplacé sur l'image du caractère), ceci avant ou après squelettisation.

D'autres méthodes proposent d'utiliser des informations comme la courbure, les concavités, les boucles, ou le nombre d'intersections avec des droites pour caractériser les formes.

### - Techniques de reconnaissance :

Avant de pouvoir prendre une décision, il faut acquérir des connaissances et les organiser en modèles de référence ou *classes*. C'est durant la phase d'*apprentissage* qu'est réalisé ce travail. Lorsque le concepteur (ou professeur) indique le nom de la forme en entrée, l'apprentissage est dit supervisé. Si la construction des classes est automatique, on parle d'apprentissage non supervisé.

- *Dans l'approche statistique*, la forme est représentée par un vecteur de  $n$  composantes correspondant à la mesure de  $n$  caractéristiques observées sur elles. La reconnaissance consiste à trouver la classe à laquelle le symbole a la plus grande probabilité d'appartenir et à évaluer le risque lié à la prise de la décision (ex : approche bayésienne).

- *L'approche stochastique* considère la forme comme un signal continu dans le temps, observable à différents endroits constituant les "états d'observation" (elle n'est donc utilisable que sur des objets comportant une information temporelle). Le modèle décrit ces états à l'aide de probabilités, de transition d'état à état, et d'observation d'états.

Un *modèle de Markov* représentera un caractère (ou un mot) par un double processus: la séquence de primitives (ou de caractères) appelée observations  $\{O_1...O_n\}$ , et la séquence d'états  $\{S_1...S_n\}$ . La reconnaissance d'un objet défini par la séquence de primitives  $\{o_1...o_n\}$  revient à trouver la meilleure suite d'états  $\{s_1...s_n\}$  et ainsi le modèle de ce symbole.

- Les approches vues précédemment ne permettent pas de prendre en compte l'information contextuelle de la forme. Une nouvelle description des objets peut donc être réalisée en utilisant ses composants élémentaires ou *primitives* (droite,

cercle, ...), leurs propriétés et leurs connexions. C'est ce que l'on appelle l'*approche structurelle*. Les modèles descriptifs capables de représenter de tels objets sont les graphes (figure 2.8) et les structures syntaxiques (chaînes ou phrases, inspirées de la théorie des langages) (figure 2.9).

La mesure de ressemblance se fait alors, soit par calcul de *distances entre chaînes*, soit par *appariement de graphes*: mise en correspondance de graphes ou sous-graphes dans le but de reconnaître si deux formes ont la même représentation, soit par *analyse de la validité* d'une phrase dans les grammaires correspondant aux modèles des formes (analyse syntaxique).

Il est aussi possible de rechercher la présence ou l'absence de ces composants élémentaires par tests successifs.

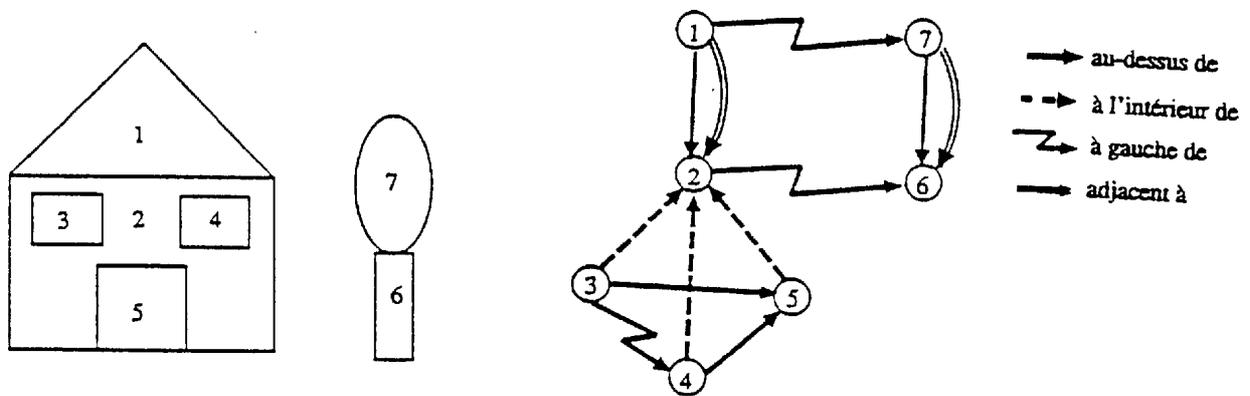


Figure 2.8 : représentation d'objets sous forme d'un graphe.

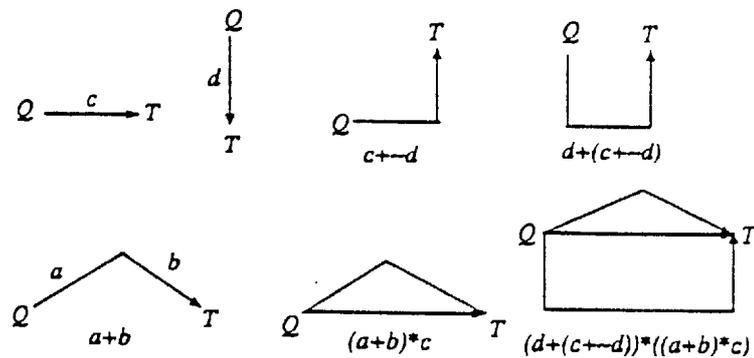


Figure 2.9 : exemple de formes définies par grammaire.

## 2.4. Conclusion :

Cette étude succincte des documents, puis du schéma classique de reconnaissance de l'écriture et des méthodes utilisées dans ce domaine, permet de discuter de leur utilité en lecture automatique de partitions musicales.

L'acquisition d'une partition nécessite les mêmes précautions que celles prises pour un texte. L'image binaire devra restituer correctement les lignes de portées : informations primordiales de la partition [Martin89].

L'image d'une partition musicale comporte rarement des bruits nécessitant la réalisation d'un filtrage avant l'analyse, l'étape de prétraitement sera donc souvent réduite à l'élimination du biais.

La structure particulière des portées, ne permet pas d'utiliser les méthodes classiques de segmentation : contrairement aux documents dactylographiés, où tous les symboles se présentent sous forme de composantes 8-connexes étiquetables, dans une partition tous les symboles sont reliés par les lignes de portées. Leur isolement suivra donc une procédure particulière qui constituera une étape importante du traitement.

La nature des signes iconiques (notes) présents dans les partitions plaide pour l'utilisation de l'approche structurale. Cependant les autres symboles (dièse, clé, ...) sont comparables aux caractères et peuvent être traités de manière différente puisque rien ne permet a priori d'opter pour une méthode plutôt qu'une autre.

Une étape de post-traitement permettra de lever certaines ambiguïtés de classification et de vérifier la cohérence des données en sortie, ceci en effectuant des vérifications d'ordre lexical et en utilisant les règles de la syntaxe musicale.

### **3. Reconnaissance de partitions : Etat de l'art**

La reconnaissance automatique de partitions musicales est un domaine qui, jusqu'à présent, a suscité beaucoup moins d'intérêt que la reconnaissance de textes imprimés ou manuscrits. Des recherches ont cependant déjà été menées, les premières tentatives datent de 1967 avec Pruslin et de 1970 avec Prerau [Carter88]. Depuis, la qualité croissante des scanners semble relancer les recherches dans ce domaine pour aboutir, peut-être, à une solution susceptible de s'intégrer dans les nouveaux systèmes musicaux.

Comme nous l'avons laissé entendre précédemment, l'architecture classique d'un système de reconnaissance de partitions musicales se présente sous la forme suivante :

- Acquisition de l'image,
- Prétraitement (correction du biais et du bruit),
- Segmentation (des portées et des symboles),
- Détection des symboles,
- Reconnaissance des symboles,
- Post-traitement (génération d'une sortie cohérente).

#### **3.1. Acquisition :**

La plupart des systèmes de lecture automatique de partitions travaillent à partir de documents scannés avec une résolution comprise entre 100 et 300 points par pouce. De toute manière, cette dernière doit être suffisante pour restituer correctement les lignes de portées, information indispensable à l'analyse. L'image est ensuite convertie en image binaire de taille plus ou moins importante 120 Ko à 1 Mo pour le format A4. Une équipe de chercheurs de l'université de Waseda (Japon) [Roads86], [Matsushima88] propose l'utilisation d'une caméra CCD (charge-coupled device) d'une résolution de 250 points par pouce pour l'acquisition de l'image, dans le cadre de la création d'un "robot musicien".

Il est parfois difficile de séparer les phases de prétraitement, de segmentation, de détection, et de reconnaissance: certains systèmes effectuent ces traitements simultanément. Ces derniers ont cependant recours à des méthodes appartenant à l'une des catégories suivantes: les *méthodes utilisant des projections*, les *méthodes par suivi*,

les *méthodes récursives*, les *méthodes par analyse de régions*, et les *méthodes topologiques*.

### 3.2. Les méthodes utilisant des histogrammes :

Le *calcul d'histogramme* suivant les deux principales directions, verticale et horizontale, est une méthode fréquemment utilisée en traitement de document. Elle l'est aussi en reconnaissance de partitions.

Les rotations subies par l'image sont toujours faibles (quelques degrés). Une valeur précise de la correction à apporter peut être obtenue en calculant *l'histogramme, sur les lignes, des pixels*. L'angle de la rotation qui maximise, soit la variance [Martin89], soit le maximum [Martin90] [Sicard92] de cet histogramme donne une excellente mesure de la correction à apporter. En pratique, Philippe Martin propose de ne mesurer l'horizontalité que sur une partie significative de l'image, par exemple autour du pic maximum du premier histogramme, avant d'appliquer la correction sur l'image globale (figures 3.1 et 3.2).

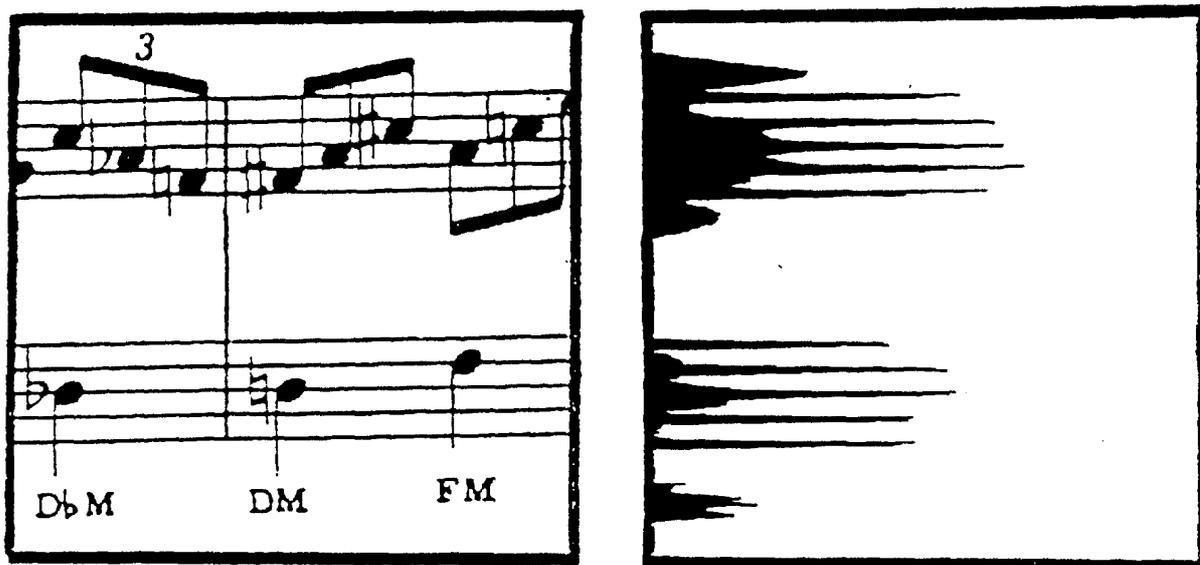


figure 3.1: partition et histogramme avant correction du biais.

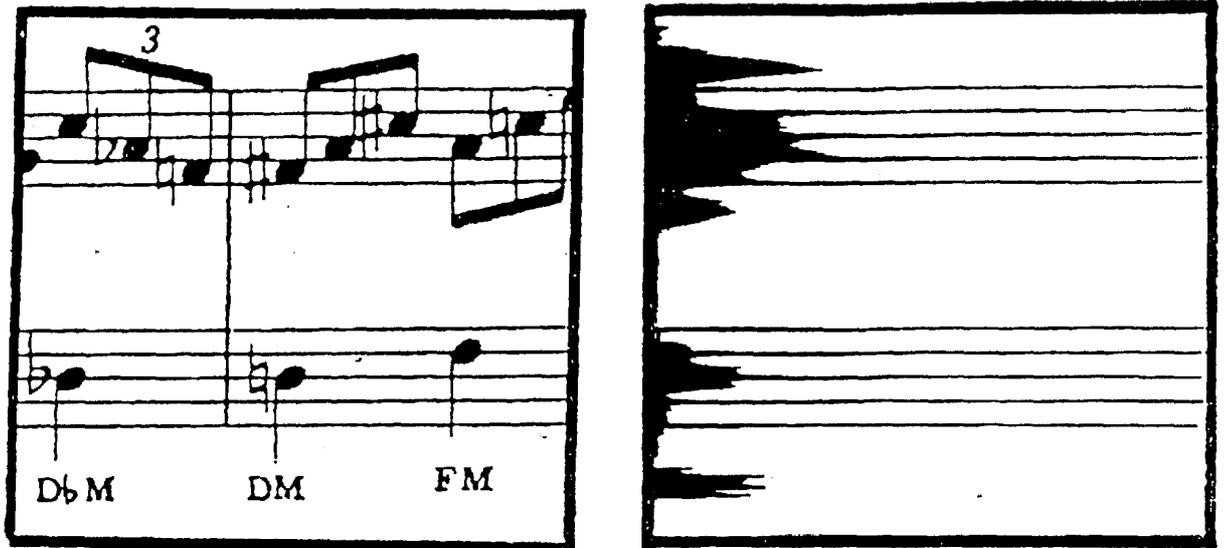


figure 3.2: partition et histogramme après correction du biais.

Les figures 3.1 et 3.2 montrent le lien entre les lignes de portées et les pics de l'histogramme. Les séparations entre portées correspondent aux larges espaces blancs entre groupes de cinq pics. Après seuillage, on ne retiendra que les groupes de cinq pics régulièrement espacés entre eux. La position et la largeur d'un pic fournissent respectivement l'ordonnée et l'épaisseur de la ligne de portée correspondante [Martin90] [Sicard92]. L'espace entre deux lignes de portée donne une bonne approximation du diamètre des têtes de notes.

Andréas Dengel et Stéphan Baumann [Dengel92] proposent d'utiliser cette méthode, seulement pour déterminer la position et la largeur des lignes. Les portées ont préalablement été séparées par une stratégie que nous détaillerons plus loin. La détection des pics est ainsi plus robuste et moins difficile à mettre en oeuvre. L'histogramme peut, de plus, n'être calculé que sur une partie (5 % à partir de la limite droite) de chaque portée, puisque les portées ont été préalablement localisées. Le temps de calcul est ainsi significativement diminué.

Les barres verticales constituent des primitives facilement détectables. Etienne Sicard [Sicard92] procède à leur extraction en calculant l'histogramme sur les colonnes d'une portée (figure 3.3). Les barres de croches (barres épaisses) ont été préalablement effacées. Les pics correspondent alors, soit à des queues de notes, soit à des barres de mesures. Les queues de notes sont toujours attachées à une tête. La hauteur des pics est comparable à celle de la portée, et leur épaisseur dépasse rarement trois ou quatre pixels. Une fois reconnues, les barres sont effacées de l'image.

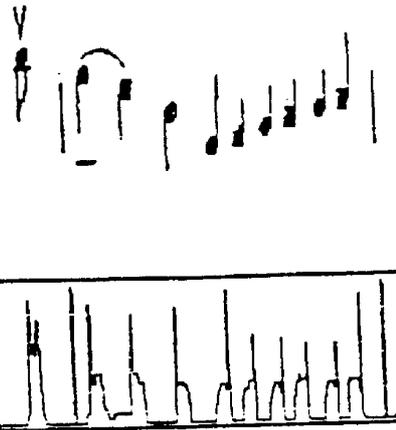


figure 3.3 : Détection des barres verticales

Dengel et Baumann [Dengel92] localisent d'abord les regroupements de notes (composantes connexes) en utilisant des rectangles englobants. La *projection verticale* de l'intérieur du rectangle permet ensuite de déterminer la présence des queues de notes et leur abscisse.

### 3.3. Les méthodes de suivi :

Une fois situées, les lignes de portées doivent être éliminées puisqu'elles nous empêchent de localiser les symboles en tant que composantes connexes. La segmentation symbole / portée consiste donc à identifier les pixels noirs appartenant à une ligne de portée. Lors de cette phase, on devra éviter, l'oubli de morceaux de lignes, et l'effacement d'informations appartenant aux symboles musicaux.

Les pixels adjacents aux lignes de portées devront donc être analysés avant de prendre la décision d'effacement. La méthode de Dengel et Baumann élimine aussi les bruits se trouvant aux alentours des lignes (figure 3.4)



figure 3.4 : Elimination des portées

Philippe Martin et Camille Bellissant [Martin90] utilisent un algorithme de suivi des lignes et déterminent les pixels à supprimer:

- Soit une ligne de portée d'épaisseur  $e$  détectée entre les lignes  $x$  et  $(x+e-1)$ .
- Pour chaque colonne  $c$  de l'image, la section verticale de la ligne de portée en  $c$  est la suite de pixels adjacents:  $(i,c), (i+1,c), \dots, (i+n-1,c)$  telle que:
  1.  $(x-\partial e \leq i \leq x+e-1+\partial e)$  ou  $(i-\partial e \leq x \leq i+n-1+\partial e)$
  2.  $\ln-e \leq \partial e$  ( $\partial e$  est l'incertitude pré-définie sur l'épaisseur de la ligne)
- La recherche de la suite en  $c$  se fait depuis l'abscisse du centre de la section en  $(c-1)$  puis en s'éloignant de part et d'autre de ce point.

La première condition exprime les variations locales d'épaisseur admises pour une ligne de portée (rétrécissement, dilatation). La deuxième réalise un premier filtrage, puisqu'elle n'étiquette pas comme "portée" des pixels qui appartiennent à la ligne mais dont l'épaisseur trop importante, traduit la présence d'un symbole.

Ce premier critère s'avère, néanmoins, insuffisant puisqu'un symbole qui est localement tangent à une ligne de portée (liaison plate, ...) n'entraînera aucune variation locale d'épaisseur et sera altéré localement. Martin et Bellissant préconisent l'examen d'un contexte plus large de l'image pour résoudre ce problème: le choix de l'effacement ou de la conservation d'un pixel sera réalisé (par un réseau de neurones multi-couches) selon la distribution des cordes.

La corde d'orientation  $\alpha$ , d'un pixel  $P$  appartenant à une composante 8-connexe  $C$ , est le segment de droite de pente  $\alpha$  passant par  $P$  et inscrit dans la composante  $C$ . La longueur de la corde est définie par la distance entre  $P_1$  et  $P_2$  (figure 3.5).

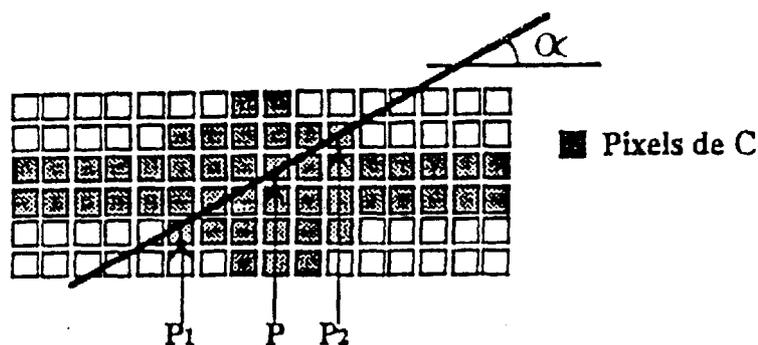


figure 3.5 : Corde d'orientation  $\alpha$  en  $P$ .

Pour détecter les lignes de portée, on considère qu'un pixel se trouvant à l'intersection d'une ligne de portée et d'un symbole est caractérisé par au moins deux longues cordes d'orientations distinctes, l'une horizontale correspondant à la portée,

l'autre selon la direction du trait du symbole intersectant. Un pixel candidat à l'érosion n'aura qu'une corde de direction horizontale (figure 3.6).

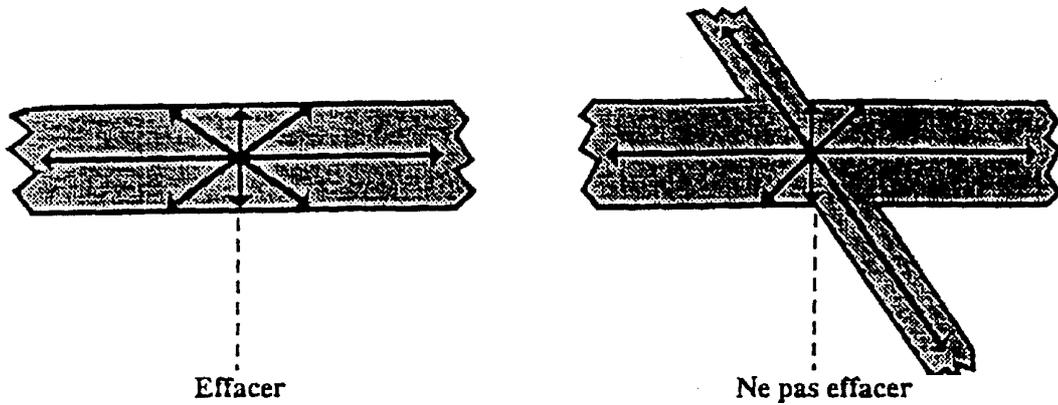


Figure 3.6: érosion des portées selon les cordes.

Ainsi, l'histogramme des longueurs de cordes calculées à différentes orientations sur  $[-\pi/2; \pi/2]$ , traduit par plusieurs pics les intersections et par un seul pic autour de zéro les fragments de portées (figure 3.7).

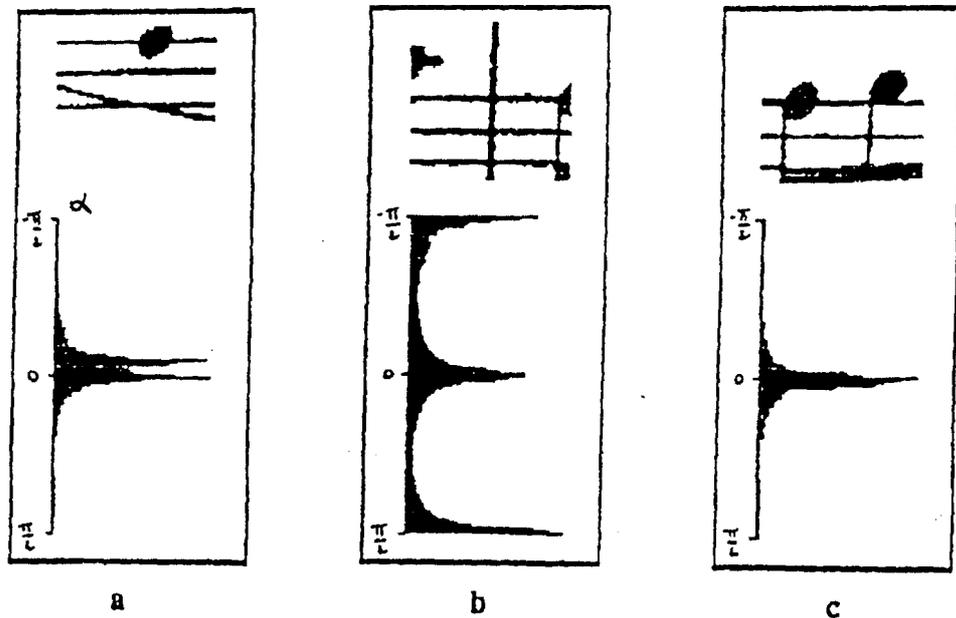


Figure 3.7: histogrammes des cordes.

Une simple détection de pics par seuillage étant insuffisante (figure 3.7.c), Martin et Bellissant soumettent l'histogramme à un réseau de neurones pour tenter d'obtenir de meilleurs résultats.

Sicard [Sicard92] utilise un algorithme de suivi pour détecter les barres de croche qu'il nomme aussi lignes épaisses. Ces lignes ont généralement une inclinaison comprise entre  $-\pi/2$  et  $\pi/2$  et une épaisseur supérieure à celle des lignes de portées. La procédure travaille sur l'image d'une portée et commence par sélectionner chaque abscisse pour laquelle le nombre des pixels noirs est supérieur à l'épaisseur d'une ligne de portée. La progression du suivi (aux colonnes voisines) est effectuée selon des critères d'épaisseur, de régularité, et par comparaison avec les caractéristiques réelles de ces lignes. Si la probabilité est suffisante la ligne est reconnue et les pixels correspondants sont effacés.

### **3.4. Les méthodes récursives :**

La formule "méthodes récursives" ne correspond pas, ici, aux techniques d'analyse employées, mais plus à une méthode d'approche générale du problème.

Jean-Pierre Armand\* base toute sa méthode sur la structure hiérarchique des partitions musicales:

*Page*  $\Rightarrow$  *Systèmes*  $\Rightarrow$  *Portées*  $\Rightarrow$  *Mesures*  $\Rightarrow$  *Blocs de notes*  $\Rightarrow$  *Primitives élémentaires*

Après extraction des objets non musicaux selon des critères topologiques, il utilise une procédure récursive pour effectuer les traitements et la reconnaissance (figure 3.8).

La procédure reçoit en entrée l'image de l'objet à analyser, les filtres nécessaires au traitement de l'image et les critères sur lesquels va se baser le choix du bon filtre. Les filtres et les critères dépendent de l'image à analyser et des résultats escomptés: si l'image en entrée est celle d'un bloc de notes, on attend en sortie, selon le schéma de la structure hiérarchique vu plus haut, des primitives élémentaires (ellipses, barres, ...). La procédure agit par filtrage, érosion, ou/et dilatation et décompose l'objet en composantes connexes plus élémentaires. Ces composantes forment de nouvelles images sur lesquelles on appliquera la même procédure.

---

\* Informations tirées d'un séminaire de Jean-Pierre Armand (RFMQ - TIMC - IMAG)

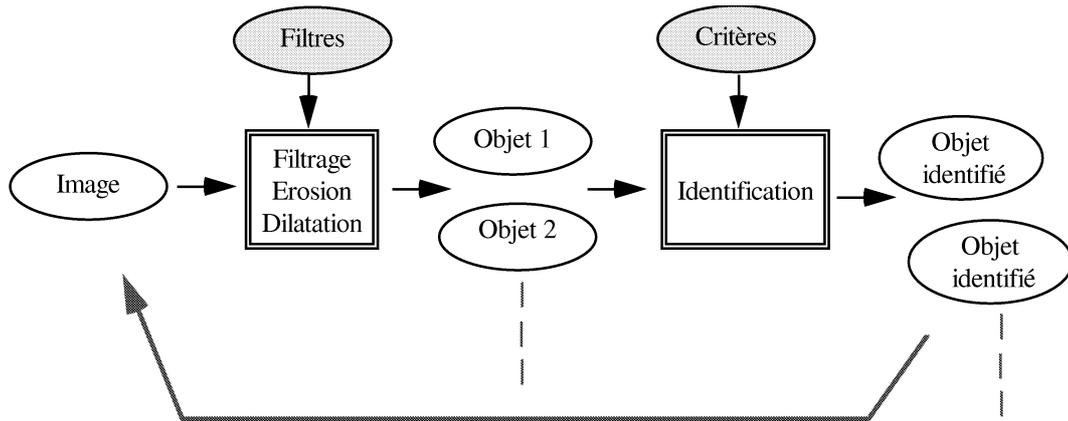


Figure 3.8: Procédure réursive.

Les critères de reconnaissance sont de type géométrique (masse, surface, inertie, compacité) et de type topologique (position de l'objet dans l'image). La décision est prise par recherche du minimum de distance par rapport aux prototypes.

Dengel et Baumann [Dengel92] utilisent une méthode comparable. Après segmentation, ils localisent les images des composantes connexes et leur appliquent une décomposition réursive en blocs. La subdivision a lieu dans les directions horizontale puis verticale, en blocs de hauteur et largeur identiques, par exploration des régions homogènes (ayant des pixels de même couleur). On obtient alors un puzzle ordonné de l'image d'origine pouvant être représenté sous forme d'arbre (figure 3.9). Les rectangles contenant des pixels noirs sont isolés et à partir de leurs dimensions, une pré-classification des symboles est réalisée. Grâce à cet unique critère (en le comparant à l'épaisseur d'un interligne), en s'aidant de règles syntaxiques musicales (clé en début de portée, ...) et de la cohérence rythmique, chaque composante connexe se voit attribuer une classe (parmi 14). La classification utilise un arbre binaire où chaque noeud représente une décision.

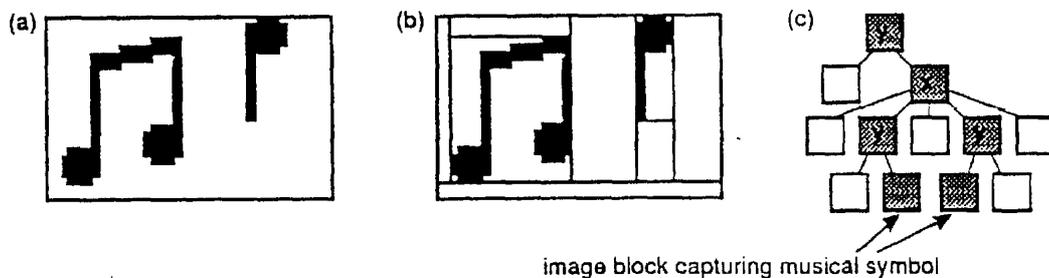


Figure 3.9: Décomposition d'une image (a) en rectangles (b), sa représentation (c).

Les éléments de la classe "bloc de notes" subissent, ensuite une analyse plus poussée: l'histogramme sur les colonnes de l'intérieur d'un rectangle permet de déterminer la présence de queues de notes (figure 3.10). Une fois qu'elles sont localisées, Dengel et Baumann recherchent l'épaisseur du segment liant les queues pour établir la durée de la note. Enfin l'existence et l'emplacement des têtes de notes sont évalués par suivi des colonnes du rectangle, en partant du côté situé à l'opposé de la barre de croche. Le premier pixel noir rencontré précise la "hauteur" à attribuer à la note.

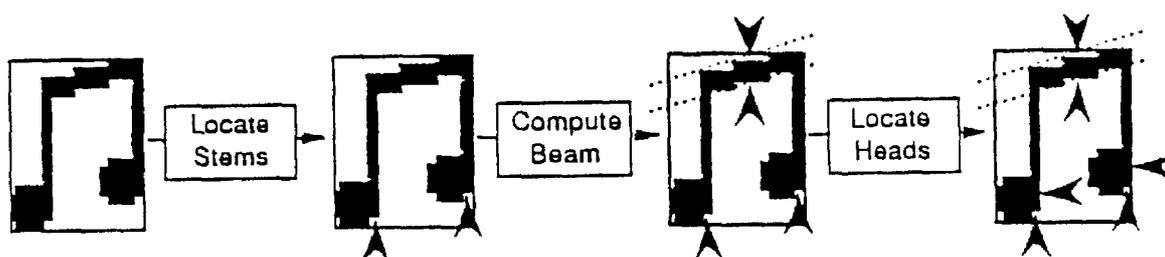


Figure 3.10: Analyse progressive des notes.

### 3.5. Les méthodes par analyse de régions :

La segmentation physique d'un document se fait généralement par analyse de régions, Dengel et Baumann [Dengel92] procèdent par *analyse descendante*. Partant d'une partition musicale, ils commencent par extraire les groupes de portées. Un groupe de portées est caractérisé par une accolade à sa gauche et par la présence de barres de mesures communes à plusieurs portées (donc plus longues). Ces composantes connexes sont détectées par *lissage*, en utilisant une version modifiée du R.L.S.A. (algorithme expliqué en 2.3). Le lissage des lignes donne une matrice A et celui des colonnes une matrice B. Les seuils utilisés, horizontalement et verticalement, sont de 0.5 cm pour des images de résolution 200 dpi. Un "OU" logique entre A et B donnera une matrice C contenant les régions recherchées.

A l'intérieur d'un groupe de portées, les barres de mesure sont ensuite détectées, par projection verticale des pixels, et éliminées. Une nouvelle application du R.L.S.A. permettra l'isolement des mesures, les seuils employés étant alors de 0,05 cm en x et 0,2 cm en y (figure 3.11).

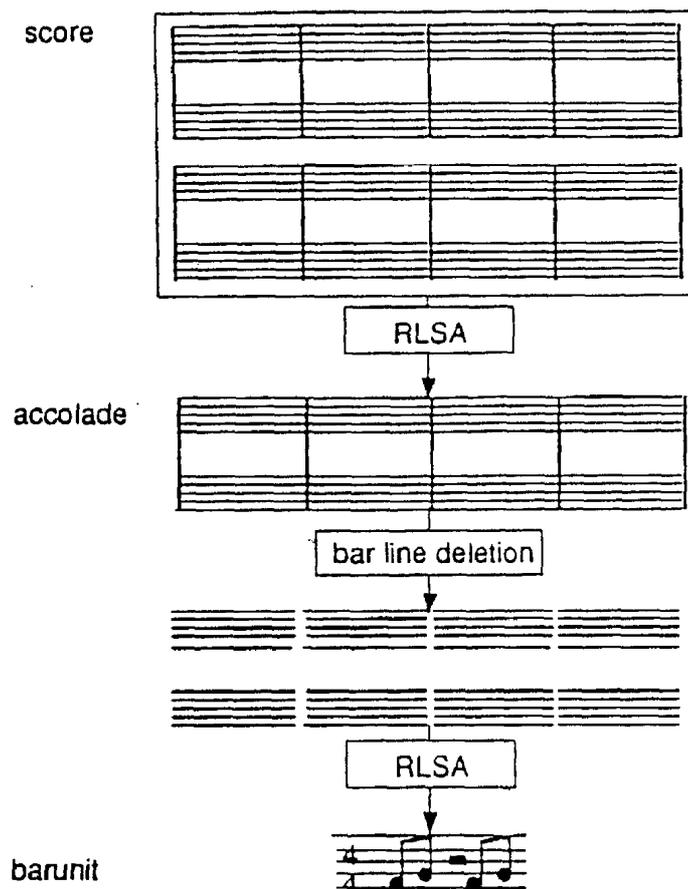


Figure 3.11: Segmentation par lissage.

Pour s'affranchir du problème de l'inclinaison des lignes de portées, Ivan Leplumey et Jean Camillerapp [Leplumey91a] emploient une technique totalement différente des techniques habituelles puisque leur méthode exploite de façon duale, l'information contenue dans les pixels blancs plutôt que dans les pixels noirs. Lorsque le pas d'échantillonnage de l'image est suffisant pour assurer la continuité des lignes de portées, il est alors possible d'effectuer un étiquetage des composantes blanches qui servira de référence pour la segmentation des pixels noirs.

Trois stratégies différentes peuvent être utilisées pour étiqueter les composantes blanches [Leplumey91b]:

Il est possible d'associer à chaque région blanche une direction et une longueur (déduites de sa matrice d'inertie); les composantes peuvent être assimilées à un rectangle ayant chacun une longueur et une hauteur. Le maximum des hauteurs correspondra à la distance entre portées, quelle que soit l'inclinaison. La moyenne des directions donnera une bonne approximation de l'angle d'inclinaison.

Un balayage de l'image selon cette pente permet de chaîner les rectangles appartenant à un même espace (élément du fond, du 1<sup>er</sup> interligne, du 2<sup>ème</sup>, ...)

Cette méthode, dite métrique, ne fait pas intervenir la notion de portée, tandis que les deux suivantes attribuent l'un des labels suivants à chacune des composantes blanches:

- élément de l'interligne FA,
- élément de l'interligne LA,
- élément de l'interligne DO,
- élément de l'interligne MI,
- élément du fond.

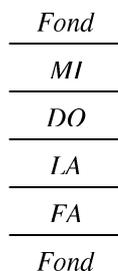
Elles utilisent pour cela un graphe de voisinage entre composantes. Les noeuds représentent les composantes blanches (ayant une hauteur et une largeur); et il y aura un arc du noeud  $i$  vers  $j$  s'il existe des lignes ou des colonnes de pixels noirs entre les 2 objets. Chaque arc possède une longueur (distance minimum entre les 2 composantes) et une hauteur (nombre de lignes ou de colonnes qui induisent l'arc).

La nature du label à attribuer un label à un noeud dépend alors du label de ses voisins. On construit donc des probabilités conditionnelles. Si un noeud a une forte probabilité d'être un MI, alors la probabilité que ses voisins de gauche et de droite soient étiquetés MI va augmenter. On peut alors s'attendre à une convergence des probabilités vers un état stable. Les probabilités initiales sont déterminées en considérant seulement les voisins de dessus et dessous ayant des surfaces de contact maximum.

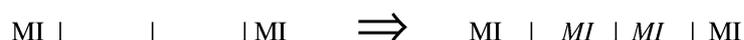
Leplumey et Camillerap ont aussi défini les notions "ressemble à" et "plus gros que" entre noeuds, qu'ils prennent en compte durant le calcul des probabilités.

L'étiquetage des noeuds peut aussi suivre une stratégie "structurelle", par application successive de 4 règles:

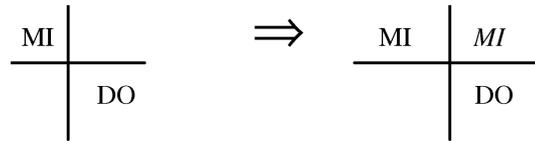
- La première effectue une détection verticale des portées, en regroupant les 5 interlignes:



- La seconde réalise l'étiquetage horizontal:



- La troisième l'étiquetage en diagonale:



- La dernière tente de labeller les noeuds restants selon la surface de contact.

### 3.6. Les méthodes topologiques :

La structure figée des partitions incite à utiliser des méthodes topologiques pour leur reconnaissance. J. W. Roach et J. E. Tatem ont choisi de tenir compte des "connaissances sur le domaine", dès les traitements de bas niveau [Tatem88].

Une fenêtre circulaire de diamètre supérieur à la largeur d'une portée, parcourt l'image. Si le pixel central est noir, on examine son voisinage. Les pixels noirs à l'intérieur de la fenêtre, ayant un voisin blanc plus proche du centre qu'eux, sont effacés. Le point noir le plus éloigné du centre (à l'intérieur de la fenêtre) forme, avec le pixel central, un vecteur  $V(\alpha, d)$  caractéristique (figure 3.12).

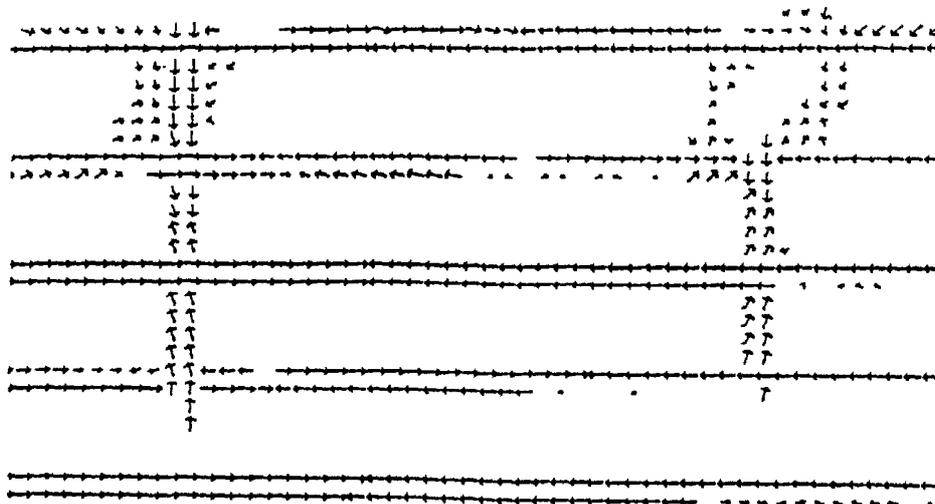


Figure 3.12: Image des vecteurs caractéristiques.

Les vecteurs ayant une direction horizontale ( $\alpha \approx 0$ ) et une norme  $d$  importante, appartiennent normalement aux lignes de portées. L'épaisseur des lignes peut alors être trouvée en parcourant la fenêtre, à partir du point central, jusqu'à détection de pixels blancs, suivant une direction perpendiculaire à celle du vecteur. Les pixels sont ensuite regroupés (à partir de règles de voisinage) selon deux labels: pixels "ligne" et

pixels "ambigus". Une ligne sera constituée d'un groupe de pixels "ligne" de largeur comparable à celle de l'image.

Les pixels blancs de l'image sont aussi étiquetés, de manière à détecter les trous correspondant aux Blanches.

En se basant toujours sur une étude du voisinage, les pixels correspondant aux lignes de portées sont effacés; il ne reste ensuite que les symboles musicaux sur l'image.

Le processus précédent (déplacement d'une fenêtre sur l'image) est répété pour détecter les notes. On recherche, cette fois, les barres verticales (queues de notes), c'est-à-dire les vecteurs de direction verticale ( $\alpha \approx \pi/2$ ). Si la barre possède une extrémité assez large et vérifie le test de circularité d'Haralick, il s'agit d'une note.

Martin a expérimenté deux méthodes de détection d'ellipses [Martin89]: L'érosion par un élément structurant ayant la forme correspondante est rapide mais échoue pour les symboles déformés.

La transformée de Hough, aussi testée dans [Tatem88], permet de rechercher de tels objets. Les rayons étant connus, l'espace des notes correspond à l'espace des centres (à deux dimensions). La résistance au bruit est excellente, mais malgré le faible nombre de dimension, les temps de calculs se sont avérés beaucoup trop longs. Tatem et Roach expliquent, de la même manière, que cette transformée n'est pas adaptée à la détection des lignes de portées, du fait de leur nombre, du bruit, et du temps de calcul.

Sicard [Sicard92] reconnaît les symboles en fonction de leur position sur la portée par comparaison directe, étude de contours et de points particuliers (figure 3.13). Partant de la gauche, il cherche une Clé, puis des dièses, puis des lettres ou chiffres, ...

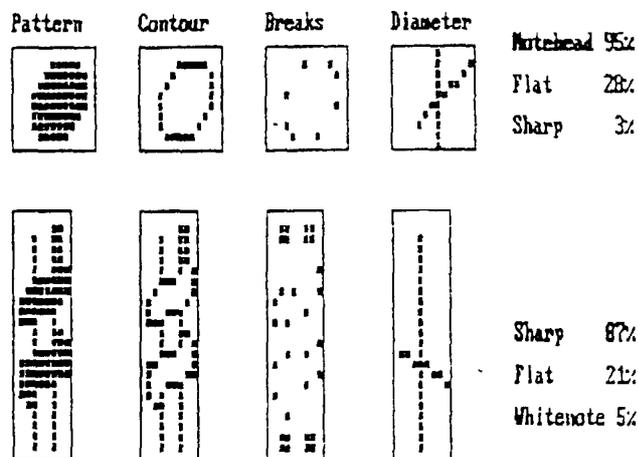


Figure 3.13: Reconnaissance des symboles.

Les symboles musicaux se décomposent naturellement en segments. L'emploi d'*algorithme de squelettisation* est donc fréquent. Tatem et Roach utilisent l'algorithme de Arcelli et Di-Baja pour obtenir un codage en chaînes des symboles qu'ils segmentent ensuite, selon une courbure maximum autorisée, pour obtenir une décomposition en segments des symboles [Tatem88].

Martin utilise une méthode similaire [Martin89]: après squelettisation par l'algorithme de Zhang et Suen, les symboles sont décomposés en suite de points connexes commençant et finissant par une extrémité ou une jonction. Les segments ainsi obtenus ne constituent pas des primitives facilement identifiables, ils sont donc soumis à un ébarbulage. Une approximation linéaire par morceau permet ensuite leur découpage récursif.

Le résultat est un graphe planaire orienté (figure 3.14) dont les sommets représentent les points remarquables du squelette, et dont les arcs codent les segments reliant ces points (sous forme de chaîne de Freeman). La classification est exécutée par réseau de neurones multi-couches.

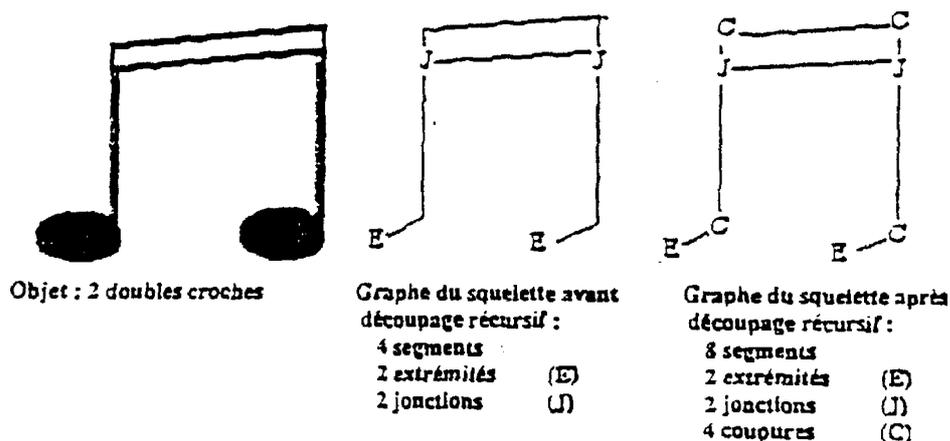


Figure 3.14: Graphe d'un squelette.

### 3.7. Post-traitements :

Cette phase vise à lever les ambiguïtés de classification, par vérifications d'ordre syntaxique ou lexical, et à fournir une représentation cohérente de la partition.

Dengel et Baumann restreignent leur système en fixant une spécification syntaxique que les séquences de symboles devront obligatoirement respecter [Dengel92]. Puis ils effectuent un transcodage des symboles en événements musicaux élémentaires répondant à la norme MIDI, qu'ils stockent dans un fichier.

Il existe de nombreux langages de représentation de la musique: M.R.L. (Music Représentation Language) [Carter88], utilisés par les éditeurs de partitions, que l'on peut aussi employer pour fournir des partitions cohérentes (souvent sous forme de chaînes de codes ASCII).

### **3.8. Résultats obtenus :**

Il est difficile de déterminer le niveau atteint par les systèmes de reconnaissance de partitions musicales. On peut cependant dire qu'ils sont encore très peu nombreux, aussi possède-t-on peu de moyens de comparaison.

Les temps de traitement sont encore très longs: il faut plus de 10 secondes au robot de l'université de Waseda [Roads86], équipé de plusieurs processeurs (dont un spécialement destiné à la détection des têtes de notes) pour lire et analyser des partitions relativement simples.

Les taux de reconnaissance dépendent énormément du type de partitions qu'est capable de traiter le système. Ils varient, d'après leurs concepteurs entre 80 et 95 % pour les partitions imprimées.

La lecture de partitions manuscrites ([Tatem88], Armand\*) reste au même titre que l'écriture, confrontée à de nombreux problèmes (non respect de la connexité, objets fortement déformés, ...) les taux de reconnaissance chutent alors fortement.

Même si certains résultats sont prometteurs, de nombreux problèmes restent en suspens, en particulier au niveau de l'isolation des symboles (symboles tangents aux lignes de portées), la qualité de cette étape devient alors un préalable incontournable à la reconnaissance de partitions complexes.

---

\* Informations tirées d'un séminaire de Jean-Pierre Armand (RFMQ - TIMC - IMAG)

## **4. Expérimentations :**

### **4.1. Présentation générale :**

Pour l'acquisition, nous avons utilisé un scanner fournissant des images binaires avec une résolution pouvant aller jusqu'à 300 points par pouce. Cette résolution est suffisante puisque les lignes de portées ont alors une épaisseur de 3 à 5 pixels. Une résolution plus faible entraîne l'apparition de discontinuités dans certains symboles, lignes verticales, ou même lignes de portées qui viennent s'ajouter aux défauts déjà présents sur le document original.

Un document musical se présente généralement sous la forme d'une feuille papier A4 sur laquelle se trouvent plusieurs portées. Les images obtenues ont donc une taille importante, environ 1 Mo, nécessaire pour stocker les 2500x3500 pixels correspondants.

Pour tester notre prototype nous utiliserons des *partitions imprimées* de complexité moyenne, du type de celles présentées figures 4.1 et 4.2. La première partition (figure 4.1) est composée de portées largement espacées, elles comprennent des symboles musicaux simples et bien séparés. Ces portées sont cependant de largeur assez faible. La deuxième partition comporte des portées aux lignes plus espacées mais sa complexité est supérieure. Les symboles et autres objets sont en plus grand nombre à l'intérieur comme autour des portées.

La structure des partitions musicales composées de nombreuses droites régulièrement espacées et ellipses laisse à penser que l'utilisation de transformées (telle que la transformée de Hough et la transformée de Fourier), adaptées à la détection de tels objets, constituerait un outil efficace. Ayant déjà été testées et abandonnées à l'instar d'autres méthodes [Tatem88] [Martin89], nous avons d'abord tenté de les utiliser à l'aide de logiciels déjà existant. S'il est en effet possible de détecter les lignes de portées et les ellipses avec la transformée de Hough et d'éliminer les lignes de portées avec la transformée de Fourier; dans les deux cas le choix de seuils présente des difficultés. De plus, les micro-ordinateurs ne permettent pas encore l'application de ces méthodes à de grandes images; celles-ci doivent être préalablement découpées ou réduites et les temps de calcul restent énormes.

*Wien* THE ENTERTAINER RAG  
 Scott Joplin – Herwig Psychär

*Moderately Ragtime* Echo:

zähl:  
 compte:  
 count:

1 + 2 + 3 + 4 1 + 2 + 3 + 4

C F C G7 C

F D7 G7 C F

C G7 C G7 F Fm C G7

Copyright 1989 by EDITION MÉLODIE (EMZ) Anton Peterer, EDUCATION DIVISION,  
 Musik-Center Zurich, Postfach 260, 8049 Zurich, Switzerland  
 Alle Rechte vorbehalten / Tous droits réservés / All Rights Reserved

Figure 4.1: Exemple de partition

# HUITIÈME LEÇON

La croche dans les mesures à  $\frac{3}{4}$  et  $\frac{4}{4}$

N° 27

Exercise N° 27 is written in 3/4 time. It consists of two staves. The first staff (labeled '1') contains a melody of eighth notes and quarter notes. The second staff (labeled '2') contains a bass line with eighth notes and quarter notes, including rests.

N° 28

Exercise N° 28 is written in 3/4 time. It consists of two staves. The first staff (labeled '1') contains a melody of eighth notes and quarter notes. The second staff (labeled '2') contains a bass line with eighth notes and quarter notes, including rests.

N° 29

Exercise N° 29 is written in 4/4 time. It consists of two staves. The first staff (labeled '1') contains a melody starting with a half note, followed by eighth notes and quarter notes. The second staff (labeled '2') contains a bass line with eighth notes and quarter notes, including rests.

N° 30

Exercise N° 30 is written in 4/4 time. It consists of two staves. The first staff (labeled '1') contains a melody starting with a quarter rest, followed by eighth notes and quarter notes. The second staff (labeled '2') contains a bass line with eighth notes and quarter notes, including rests.

Figure 4.2: Exemple de partition

La méthode que nous avons adoptée consiste, après avoir mis en évidence la structure globale, à privilégier les formes ayant la plus grande fréquence d'apparition ainsi que celles de détection aisée. En ce sens, on se rapproche de la méthode utilisée par le système visuel humain. Il est alors possible de fixer une stratégie d'analyse indépendante du document à analyser et dans laquelle chaque étape est motivée par les règles de la syntaxe musicale.

Les premières étapes reposent sur des méthodes basées sur le calcul d'histogrammes, nous utilisons ensuite des méthodes proches du suivi de contour, enfin l'analyse des symboles restants sera effectuée par calcul d'intersections avec des droites. Le plan suivi sera alors le suivant:

- Détection et séparation des portées constituant la page.
- Pour chaque portée:
  - Localisation des 5 lignes constituant (généralement) la portée.
  - Détection des têtes de notes situées au voisinage des lignes.
  - Détection des barres de notes associées aux têtes de notes.
  - Détection des barres de mesures.
  - Détection et analyse des symboles restants.
  - Post-traitements.

## **4.2. Détection par calcul d'histogrammes :**

### - Détection des portées :

Les rotations subies par l'image durant la phase d'acquisition ou induites par les photocopies successives sont toujours faibles. Aussi avons nous décidé, dans un premier temps, de ne pas effectuer la correction du biais de l'image mais plutôt d'utiliser, durant toute la suite du traitement, des méthodes capables de surmonter le problème.

Lorsque l'inclinaison reste assez faible, la localisation des portées peut être réalisée par calcul de *l'histogramme en considérant les lignes* de l'image.

Il est inutile de calculer l'histogramme sur toute l'image [Dengel92]. Le début de chaque ligne de portée comporte la clé et d'autres renseignements qui rendent très difficile l'utilisation de cette partie de l'image; le fragment le plus significatif, dans lequel les portées sont le plus facilement localisable, nous a semblé être la partie

droite de l'image. Les tests ont montré qu'un calcul réalisé sur le dernier cinquième des colonnes était suffisant (figure 4.3).

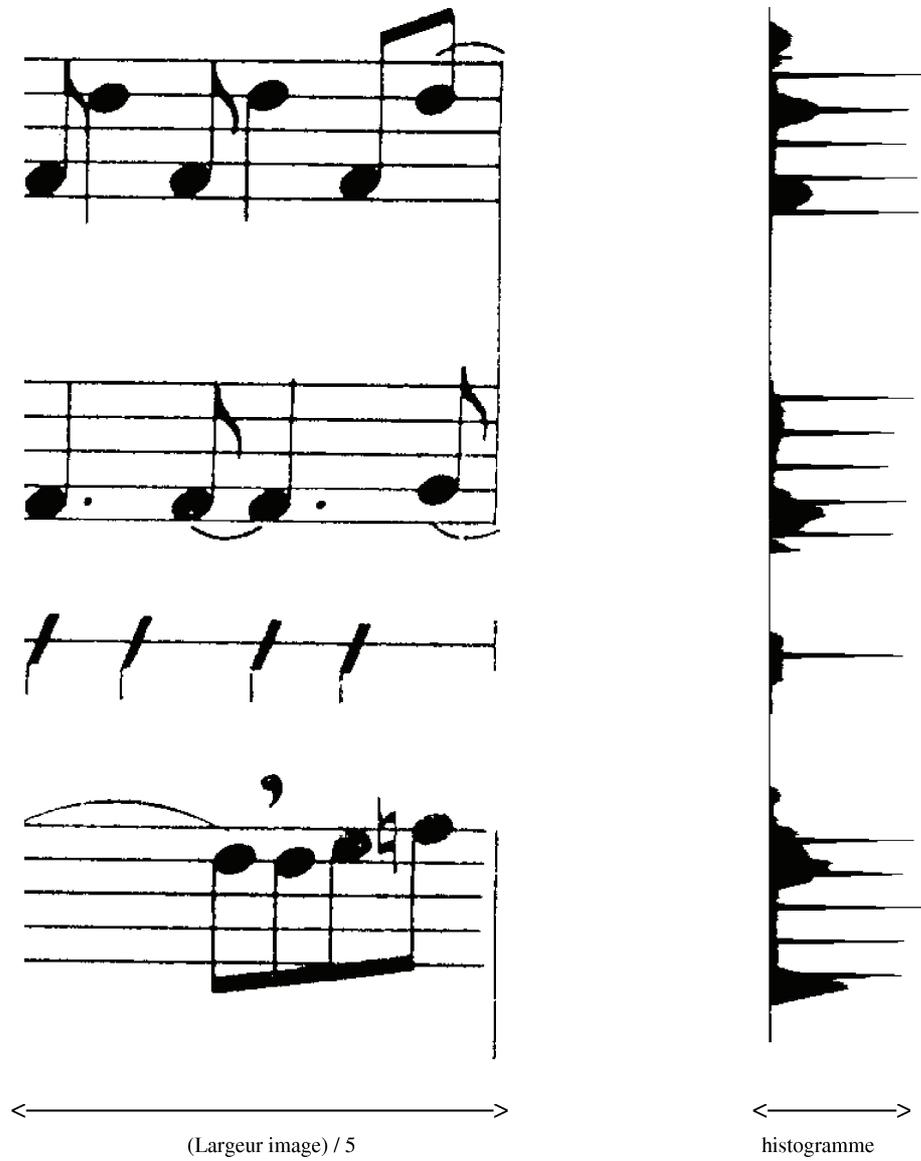


Figure 4.3: Histogramme horizontal du dernier cinquième des colonnes de l'image

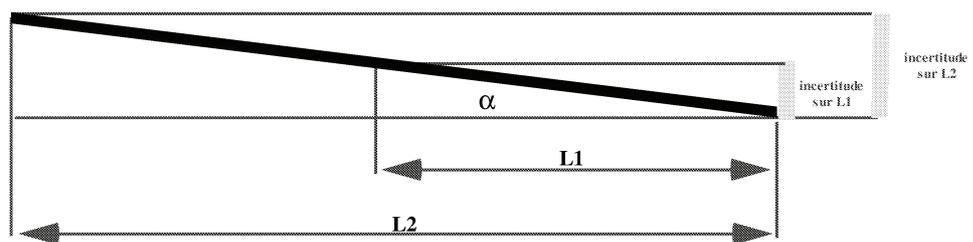


Figure 4.4: Inclinaison et localisation

Outre la diminution du temps de calcul, cette optimisation autorise un angle d'inclinaison initial plus grand. En effet une ligne de portée inclinée de  $\alpha$  degrés produira un décalage entre l'ordonnée droite et l'ordonnée gauche (incertitude) d'autant plus important que la largeur étudiée ( $L1$  et  $L2$ ) est grande (figure 4.4), leur localisation horizontale perdant alors en précision.

Les portées sont isolées en recherchant les grands espaces blancs qui les séparent. Soit  $M$  la valeur maximum de l'histogramme. En parcourant l'histogramme ligne par ligne et en regroupant celles dont la valeur est inférieure au seuil  $S = M/2$ , les régions blanches sont localisées. Les larges zones blanches entre deux groupes de pics correspondent alors aux espaces inter-portées (figure 4.3). A la fin de cette opération chaque portée est délimitée par une ordonnée initiale et une ordonnée finale.

- Détection des lignes de portées :

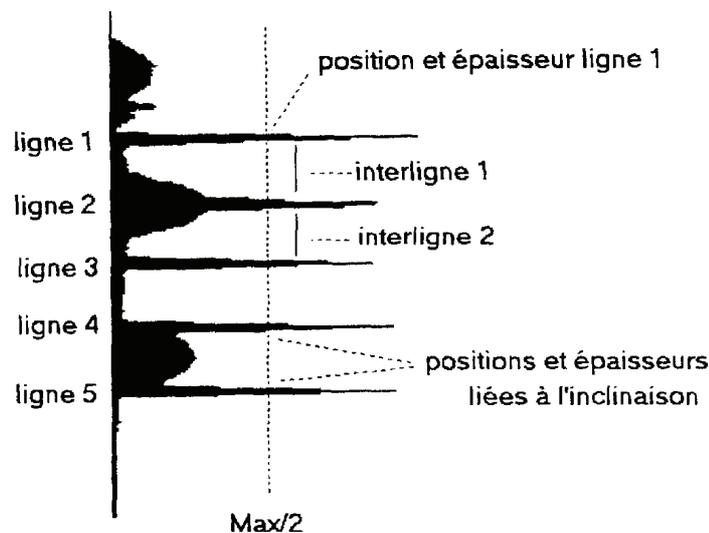


Figure 4.5: Données fournies par l'histogramme horizontal.

Il semble ensuite naturel de travailler portée par portée. Comme le montre la figure 4.3, le même histogramme permet aussi de détecter les 5 lignes de la portée en cours d'étude. Il a été décidé, pour l'instant, de ne traiter que les portées composées de 5 lignes: l'histogramme d'une portion d'image reconnue comme portée devra donc contenir 5 pics. S'il est impossible d'obtenir ces 5 pics, par diminutions successives d'un seuil (en partant du maximum  $M$ ), la portée est rejetée.

Dans le cas contraire et en se plaçant toujours au niveau du seuil  $S = M/2$  (voir figure 4.5), la position et l'épaisseur de chaque pic donnent respectivement l'emplacement et une approximation de l'épaisseur de la ligne qui l'a induit. La distance entre deux pics successifs, hauteur de l'interligne, fournit une donnée importante qu'il faut mémoriser. Les informations conservées sont, en fait, l'ordonnée "droite" de chaque ligne de portée, l'épaisseur moyenne des lignes de la portée ( $e_{moy}$ ), calculée sur les 5 pics de l'histogramme, et l'interligne moyen ( $i_{moy}$ ) calculé d'après la hauteur des 4 espaces entre pics.

Dans la suite, les numéros 1 à 5 désigneront, respectivement, les lignes Ré à Fa de chaque portée (figure 4.5).

### - Détection des têtes de notes :

Muni de ces données, il est possible d'assimiler les têtes de notes à des ellipses légèrement inclinées.

Plutôt que d'appliquer un masque ellipsoïdal, alourdissant fortement les temps de traitement, il semble préférable d'utiliser le **rectangle englobant** correspondant (figure 4.6); celui-ci est suffisant pour la détection des têtes de notes et possède, de plus, une largeur  $l$  et une longueur  $L$  définies comme suit:

$$l = i_{moy} + e_{moy}$$

$$L = i_{moy} + 2 \times e_{moy}$$



Figure 4.6: Masques correspondant aux têtes de notes.

Nous parlerons, dans la suite, du *masque des têtes de notes*, il s'agira alors du rectangle  $R(L,l)$ .

Notre objectif est la détection des têtes de notes; celles-ci sont généralement en très grand nombre et toujours situées à proximité des lignes de portées, il est donc inutile de parcourir toute l'image. Partant du côté droit de l'image (côté que l'on vient d'étudier), connaissant l'ordonnée de chaque ligne de la portée, un algorithme de suivi du trait permet, au fur et à mesure de sa progression (vers la gauche), d'examiner le voisinage de chaque ligne afin de déceler la présence de notes. Le traitement se voit ainsi accéléré et le risque d'erreur réduit.

Durant le suivi d'une ligne, seules les notes ayant l'une des 2 positions, "hauteurs", décrites par la figure (4.7.a) sont détectées. Une fois ces configurations fixées, chaque note est détectée une seule fois lors du suivi de la ligne adéquate, on évite ainsi les duplications. Un traitement spécial a été mis en place pour la première et la dernière ligne puisque les notes peuvent en être plus éloignées. Pour la ligne 5, il faut étudier les parties situées au-dessus et au-dessous de la ligne.

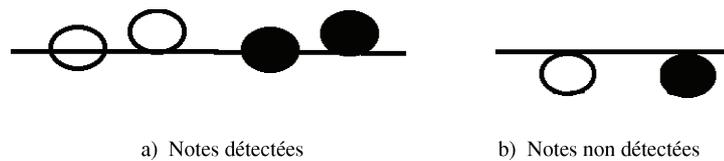


Figure 4.7: Suivi de ligne et détection (lignes 2 à 4).

Un (ou plusieurs) "pointeur(s)" placé(s) à une distance  $d$  de la ligne suivie indique(nt) la présence d'objets (figure 4.8). Lors du suivi de la ligne 1 et de la ligne 5, trois pointeurs sont nécessaires à l'exploration des parties à étudier. Pour les autres lignes (2 à 4), un seul pointeur permet de localiser les têtes de notes appartenant à l'une des 2 configurations décrites précédemment.

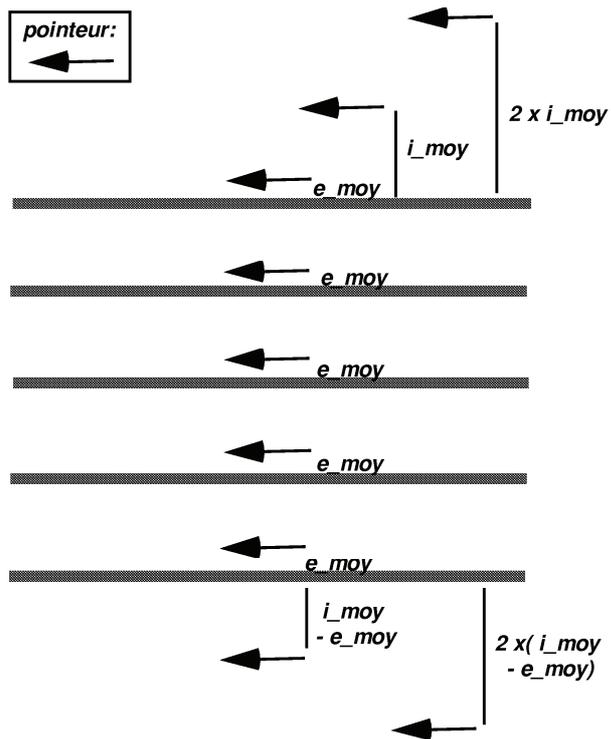
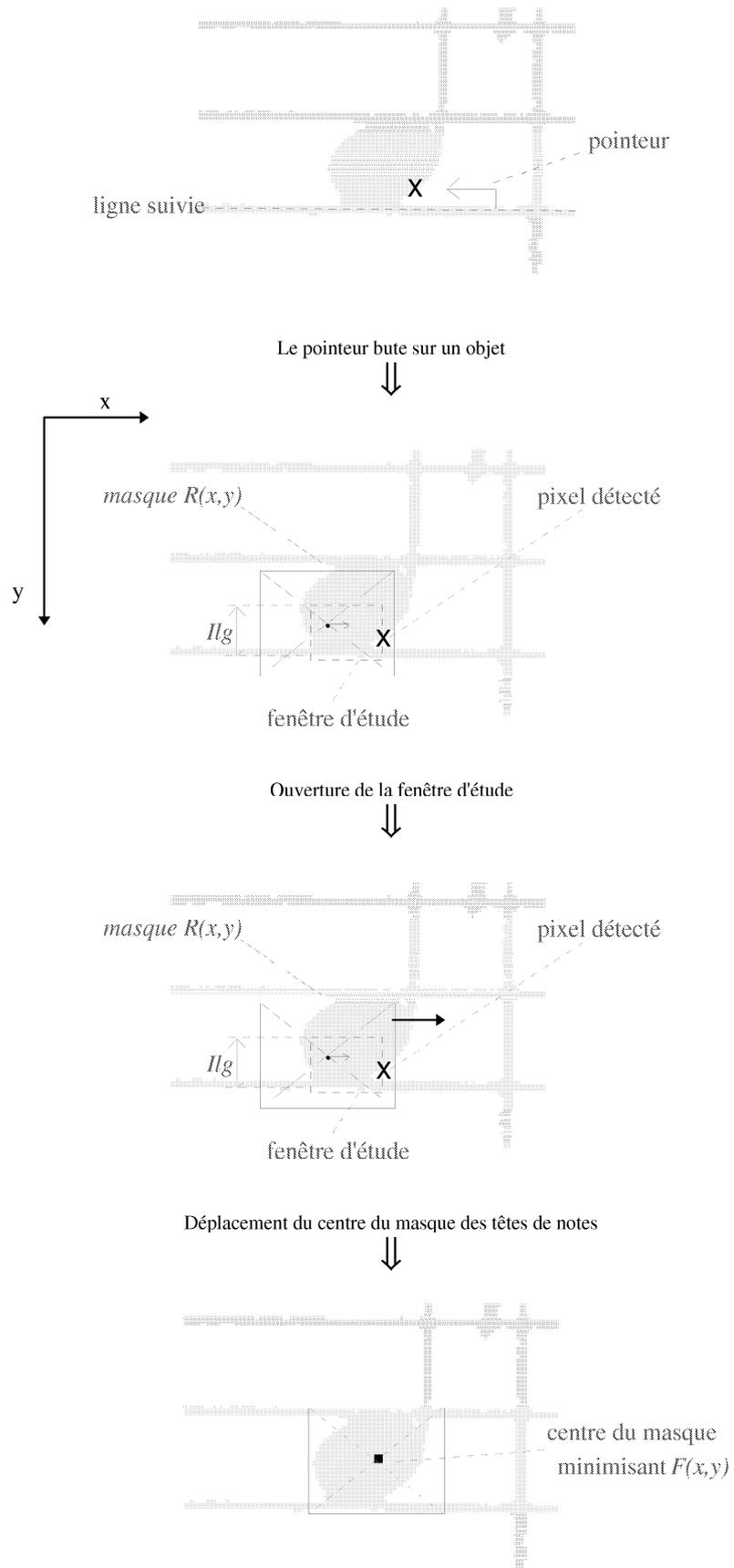


Figure 4.8: Emplacements des pointeurs lors des suivis de lignes.

Si un pointeur "bute" sur un pixel noir (détection d'un objet), une *fenêtre d'étude* est ouverte autour de ce point. Le centre du masque des têtes de notes est déplacé sur toute la fenêtre pour ensuite conserver la position qui maximise le nombre de pixels noirs en son intérieur. Ce processus est résumé figure 4.9.



Conservation de l'emplacement maximisant le nombre de pixels noirs

**Figure 4.9:** Processus de détection des têtes de notes.

Une fois le masque positionné, son intérieur doit être analysé afin de lui attribuer l'un des labels suivants: *Noire*, *Blanche*, *Symbole à ignorer*. Au lieu d'étudier des configurations particulières, c'est à l'aide de *deux histogrammes* calculés sur l'intérieur du masque, l'un suivant les colonnes, l'autre suivant la pente  $p$  d'une diagonale du masque, que la décision va être prise. Le nombre des pixels d'un segment de pente  $p$  est calculé à l'aide d'un algorithme dérivé de celui de Bresenham [Schweizer87].

La figure 4.10 illustre le lien entre histogrammes et têtes de notes correspondantes. Les symboles, dont les histogrammes ne s'apparentent pas à ceux d'une tête de note, sont ignorés dans cette phase de l'étude. Pour les autres, la position du centre du rectangle englobant est stockée puisqu'elle représente celle du centre de la note.

Par comparaison avec l'ordonnée (actuelle) de la ligne suivie, une hauteur (Do, Ré, ..., Si) est attribuée à la tête de note.

L'existence d'un point (composante connexe de diamètre avoisinant  $e_{moy}$ ) à la droite de la tête est aussi testée durant cette phase; les notes pointées sont ainsi localisées.

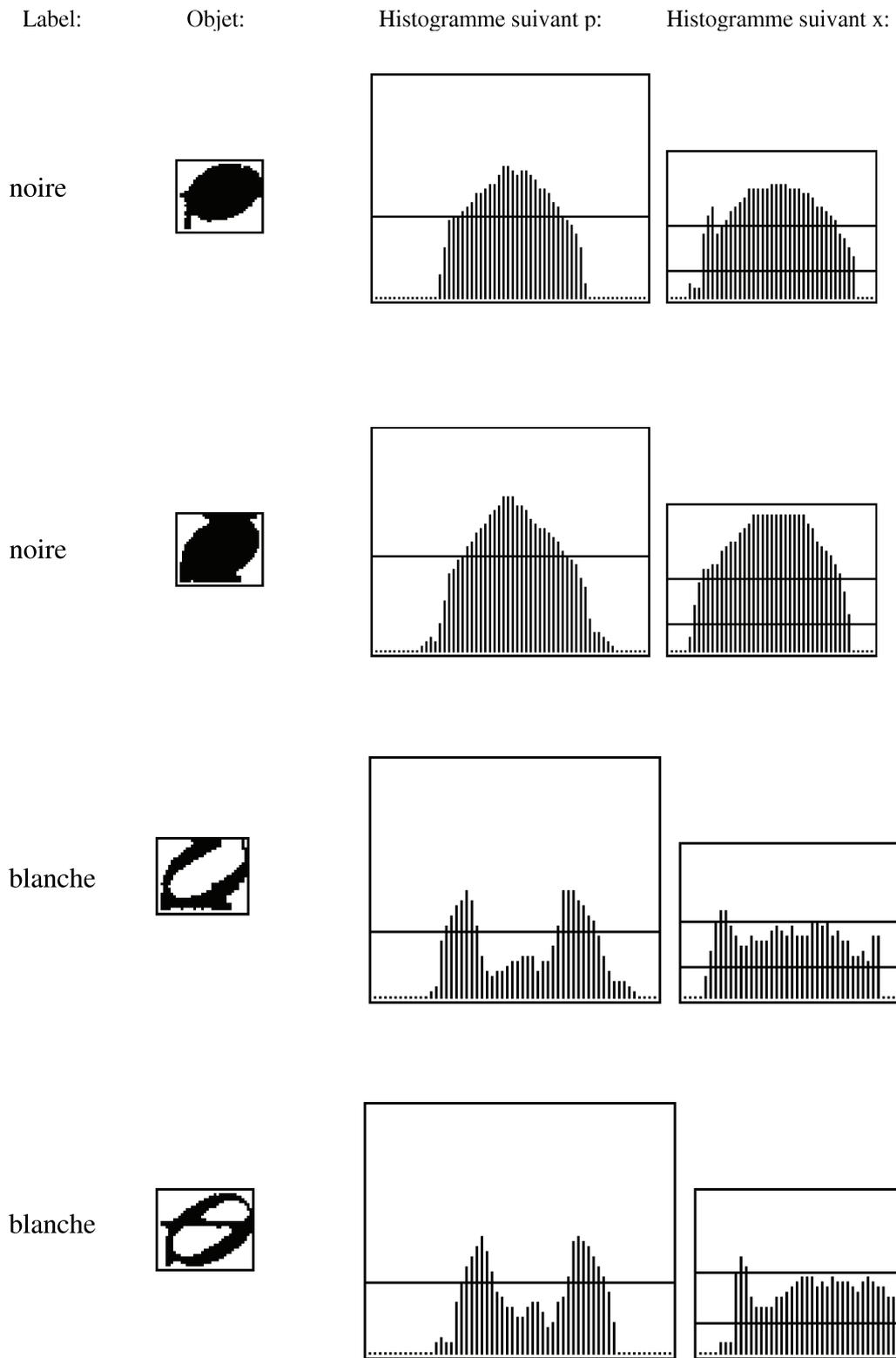


Figure 4.10: Histogrammes d'objets détectés.

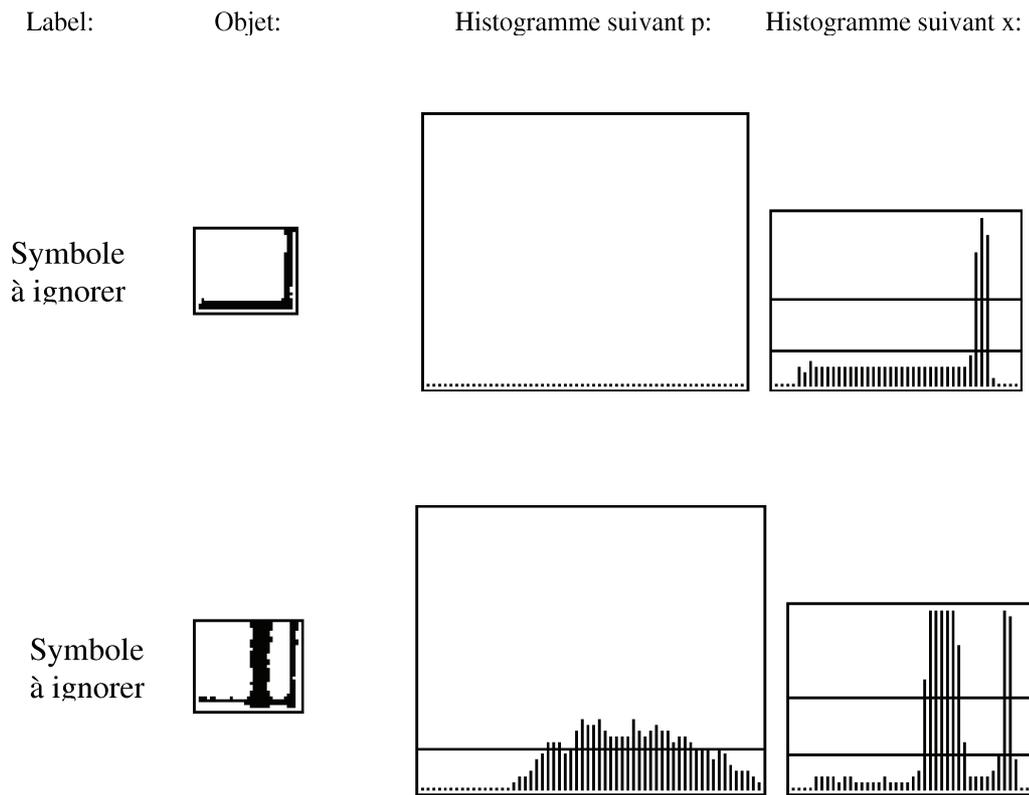


Figure 4.10: Histogrammes d'objets détectés.

### 4.3. Détection par suivi de contour

La forte connexité des partitions musicales, induite par les lignes de portées, interdit la mise en place de méthodes de suivi sans prendre des *précautions préalables*. En effet, la plupart des symboles intersectent des lignes de portées; il est alors difficile de savoir si l'on suit l'objet désiré ou une ligne.

D'autres problèmes, du même type (figure 4.11), nous ont poussé à *effacer les objets déjà reconnus*, on élimine ainsi la plupart de ces difficultés.

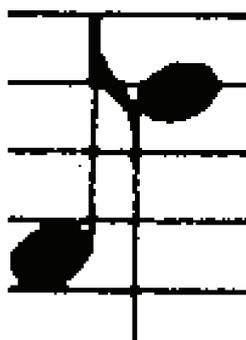


Figure 4.11: Problème de connexité.

#### - Elimination des lignes de portées :

Il est bien sûr nécessaire de supprimer les lignes de portées. Les méthodes d'érosion par un élément structurant ont été proscrites, d'une part, parce qu'elles ne tiennent compte que d'un contexte réduit de l'image, et d'autre part, du fait de la difficulté d'élaboration d'un élément structurant s'adaptant à toutes partitions.

Cette opération peut être réalisée lors de la recherche destinée à la détection des têtes de notes (vue précédemment). Afin de n'enlever que les fragments de lignes non recouverts par les symboles musicaux, la décision d'effacement (de la suite de pixels noirs) est prise, pour chaque colonne, après comparaison avec l'épaisseur moyenne des lignes (*e\_moy*) et analyse de son voisinage supérieur et inférieur.

Ces critères n'empêchent pas l'effacement de certaines parties d'objets tangentes ou confondues avec les lignes. Les résultats étant, néanmoins, satisfaisants (figure 4.12) et le traitement de ce problème entraînant l'emploi de méthodes beaucoup plus lourdes, nous avons décidé de laisser cette difficulté à la charge des procédures de plus haut niveau.



(a)



(b)

Figure 4.12: Portée initiale (a) et après effacement des lignes (b).

### - Détection des queues de notes :

Toute queue de note se trouve obligatoirement à proximité d'une tête de note. Un algorithme de suivi permet, maintenant, à partir de l'emplacement connu de chaque ellipse, de déterminer si la note possède ou non une barre verticale.

L'ellipse est divisée verticalement en sa moitié. Partant de sa partie gauche, et privilégiant la direction verticale tout d'abord vers le bas, on recherche le plus long segment vertical présent. On fait ensuite de même pour la partie droite, en se dirigeant cette fois vers le haut (figure 4.13). L'algorithme accepte les coupures du segment tant que celles-ci n'excèdent pas l'épaisseur d'une ligne de portée  $e_{moy}$ . S'il existe un segment de longueur suffisante, les coordonnées de l'extrémité (opposée à la tête de note) sont mémorisées.

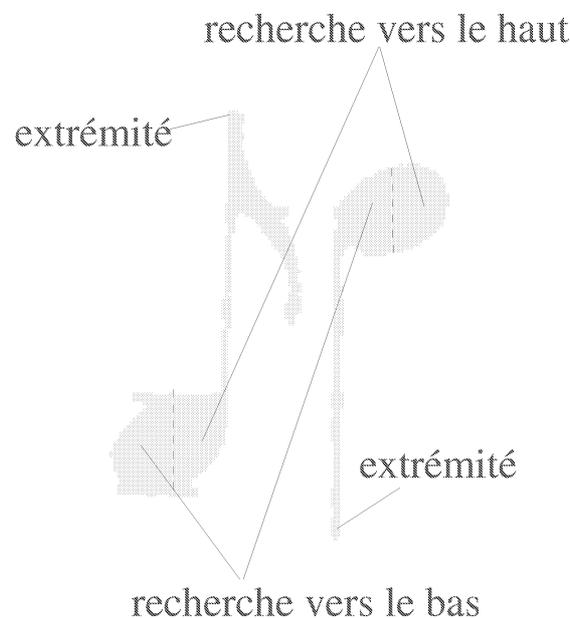


Figure 4.13: Détection des queues de notes.

Une fois cette étape terminée, pour rester fidèle à notre stratégie, la tête de note et sa queue (si elle existe) sont effacées de l'image. Les queues ne sont, en fait, pas effacées totalement, mais leurs extrémités sont conservées puisqu'elles peuvent faire partie de barres de croches (voir figure 4.14).

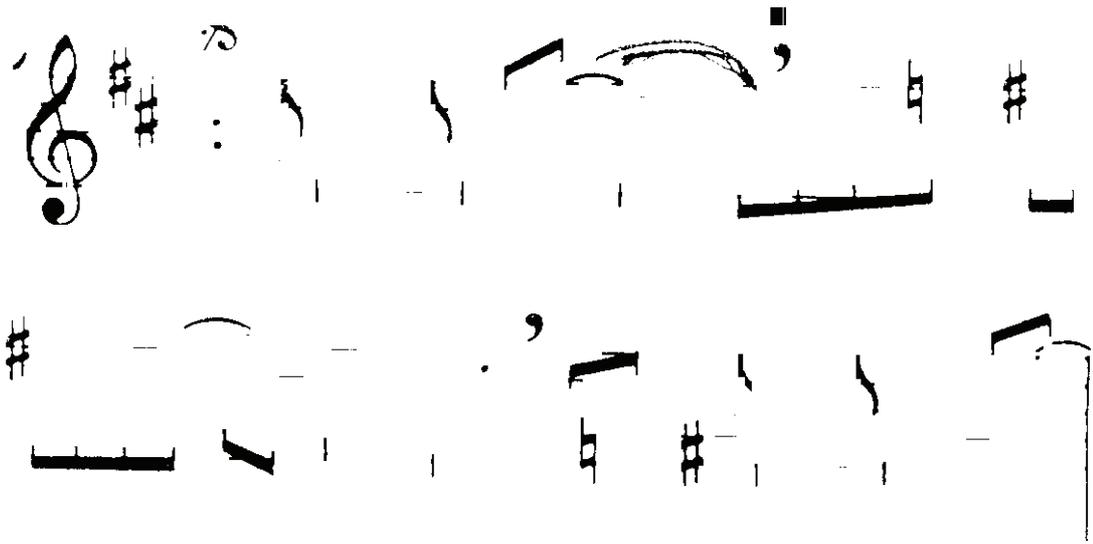


Figure 4.14: Portées après effacement des têtes et queues de notes.

### - Détection des barres de croches :

On rencontre deux types distincts de *barres de croches*: les barres horizontales ou obliques liant plusieurs queues de notes et les crochets attachés à une seule queue.

- Intéressons nous d'abord aux barres liant plusieurs queues de notes. Connaissant les extrémités de deux barres successives, il est possible par utilisation d'un algorithme proche de l'algorithme de Bresenham [Schweizer87] de *vérifier l'existence de segments* entre chaque couple de queues successives. L'algorithme tente de relier les points extrémités, puis les autres points des 2 queues, en se dirigeant vers les têtes de notes, par diminution ou augmentation des ordonnées (de ces points). L'analyse des segments trouvés permet de déceler tous les types de liens entre queues de notes.

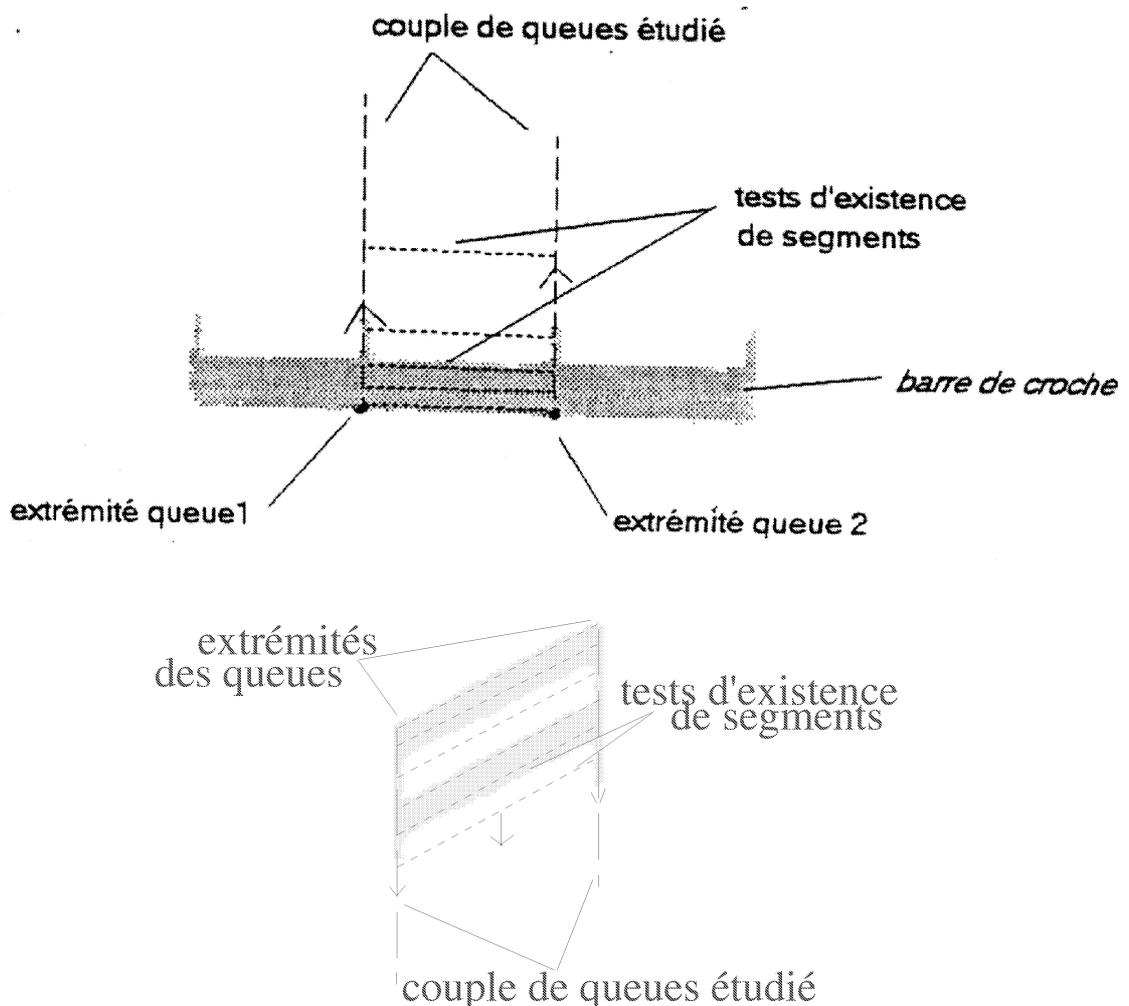
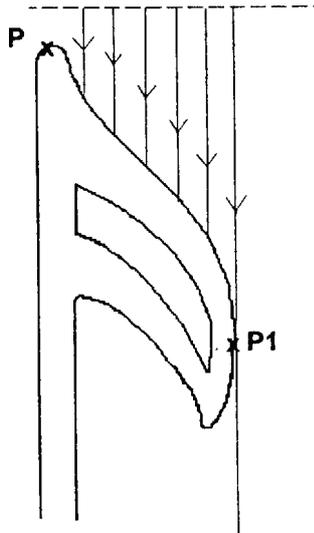


Figure 4.15: Détection des barres de croches.

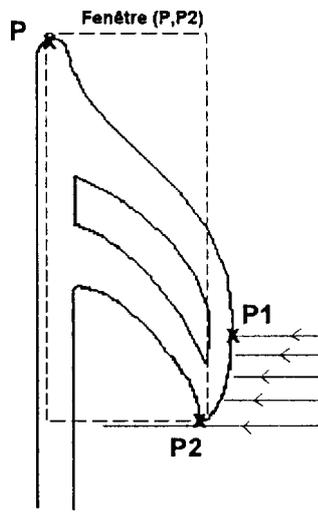
- S'il n'a été découvert aucun lien entre deux queues, il faut alors regarder si la première queue du couple ne se termine pas par un crochet.

Afin de pouvoir analyser le type du crochet, l'algorithme commence par rechercher l'extrémité du crochet (pixel noir le plus "éloigné" de l'extrémité de la queue examinée). La quête de ce point est réalisée en deux étapes:

- Recherche de la fin du crochet selon les colonnes  $\Rightarrow$  Point intermédiaire P1.



- A partir de P1, recherche selon les lignes  $\Rightarrow$  Point final P2.



Une fois le rectangle (P,P2) défini, l'analyse de sa taille permet de statuer sur l'absence éventuelle de crochet.

Si, au contraire, ses dimensions sont suffisantes, le nombre de transitions Noir-Blanc rencontrées lors du parcours de la colonne centrale détermine la durée de la note. La figure 4.16 montre des exemples de fenêtres obtenues avec cette méthode.



Figure 4.16: Exemples de crochets détectés.

Après l'examen de chaque queue, les barres de croches détectées ou morceaux de queues restants sont effacés.

#### 4.4. Traitement des symboles :

##### - Tamissage de la portée :

Il ne reste, normalement, à ce niveau de l'analyse, que les barres de mesures et les signes musicaux symboliques (clés, dièses, bémols, ...) dans l'image de la portée. Toutefois, certains symboles ayant été mal effacés (voir figure 4.17), l'image comporte en plus un certain nombre de bruits (morceaux de ligne de portée non effacés car situés à proximité d'un symbole, points, ...) qu'il est préférable d'éliminer par *tamissage*.

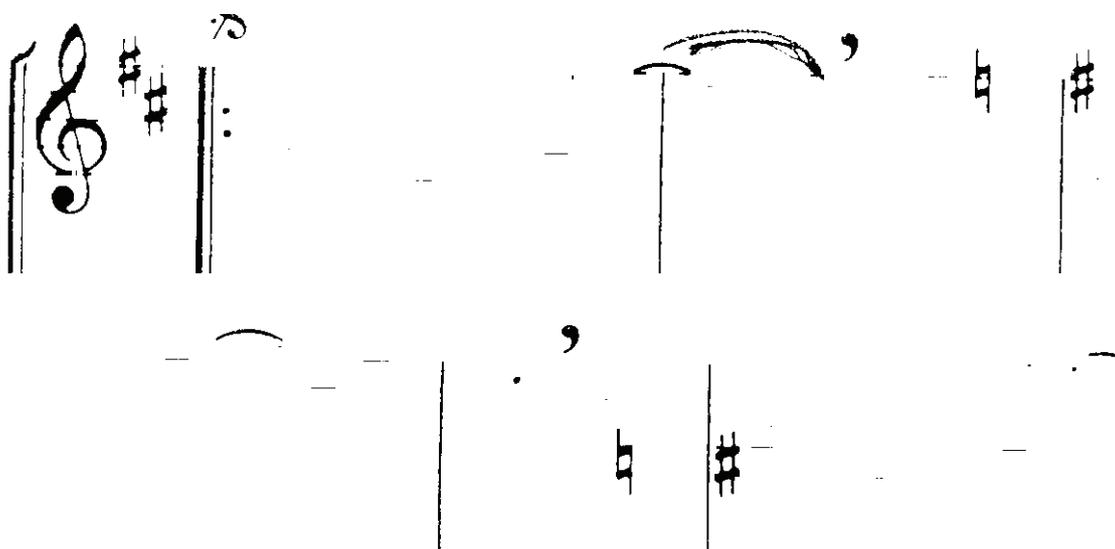


Figure 4.17: Portée avant tamissage.

Cette opération revient à ne garder que les composantes connexes vérifiant un critère de taille. Dans notre cas, les objets à éliminer sont toujours minces et parfois horizontalement allongés; ceux à conserver sont de taille non négligeable, le risque de confusion semble donc très faible.

Le filtrage est réalisé par déplacement d'un rectangle de largeur  $l$  et de longueur  $L$  sur l'image, si le bord du rectangle n'intersecte aucun pixel noir, les pixels à l'intérieur sont mis à blanc ( avec  $l = 2 \times e_{moy}$  et  $L = (3 \times i_{moy}) / 2$  ).

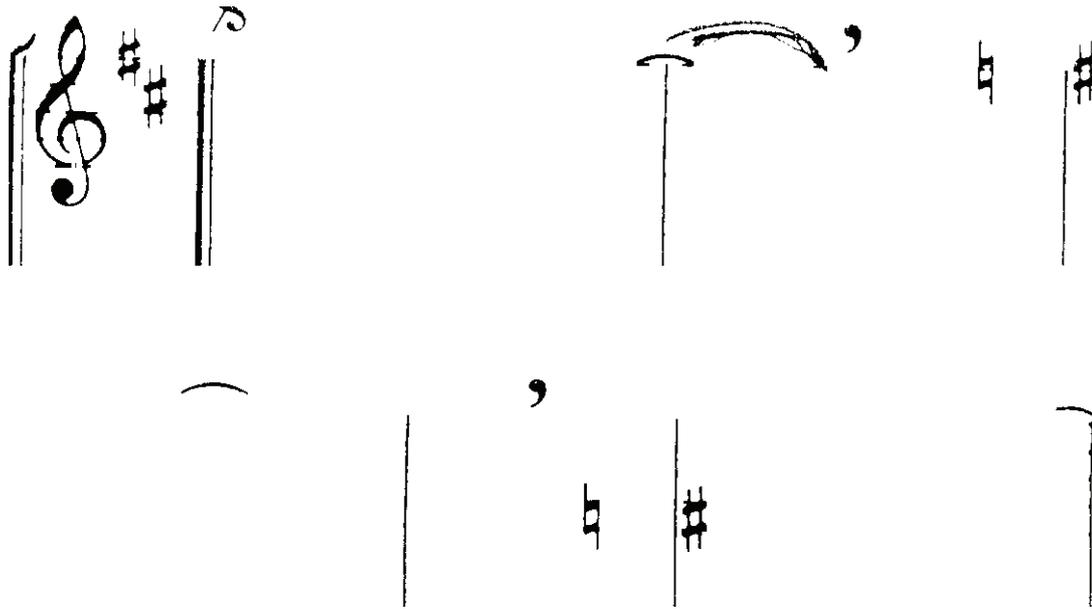


Figure 4.18: Portée après tamisage.

#### - Détection et analyse :

Avant d'aborder la phase de traitement des symboles, il nous reste encore à détecter les barres verticales (barres de mesure, de renvoi, ...). Pour cela nous pouvons de nouveau avoir recours à l'utilisation d'un *histogramme suivant les colonnes* ou suivant la direction orthogonale à celle des lignes de portée. Seul des liaisons ou des éléments de commentaires peuvent interférer. Après leur détection, les barres verticales sont effacées de l'image.

Chaque symbole est, alors, une composante connexe (clés, dièses, bémols, liaisons, soupirs, ...) que l'on peut localiser, isoler, puis analyser. Certains symboles (points d'orgue, ...) sont composés de plusieurs composantes connexes; la détection de l'un des composants devra, alors, activer un processus de recherche des autres primitives et de reconstruction du symbole.

Pour localiser et isoler les symboles, la méthode la plus simple, nous a semblé être de déterminer le rectangle englobant chaque composante connexe.

La taille des rectangles (*Largeur* et *Longueur*) fournit une information importante, mais cependant insuffisante pour classifier tous les symboles. C'est pourquoi nous avons choisi d'effectuer, en plus, une classification selon le nombre d'intersections entre le symbole et les lignes horizontales contenues dans son rectangle englobant [Vo-Van77]. Les valeurs ainsi obtenues, regroupées puis analysées, devraient permettre d'obtenir une Suite significative, caractéristique d'une classe de symboles.

Chaque symbole serait alors représenté par un vecteur à "trois" composantes. Il sera préférable d'attribuer une taille fixe à  $S$  (choix d'un majorant et complément de la suite par des 0):

$$\vec{S} \begin{pmatrix} l \\ L \\ S \end{pmatrix}$$

Dans un espace euclidien à  $n$  dimensions, la distance entre deux points repérés par leur vecteur  $s_1$  et  $s_2$  s'écrit [Besançon88]:

$$D(s_1, s_2) = \|s_1 - s_2\| \Leftrightarrow$$

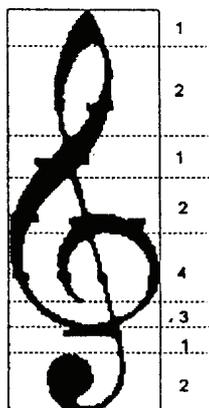
$$D(s_1, s_2) = \sqrt{\sum_{k=1}^n (s_{1k} - s_{2k})^2}$$

Cette *distance* devra être calculée entre l'objet en cours d'étude et chaque prototype de symbole musical, stocké dans une base de connaissances (valeurs fixes obtenues à partir d'échantillons). On attribuera au symbole, le label du prototype qui lui est le plus proche (c'est-à-dire celui dont la distance est minimum). Si la valeur de cette distance est trop élevée, le symbole est alors considéré comme inconnu. Afin de réduire encore le risque d'erreur, il est possible de tenir compte de l'emplacement des symboles lors de la classification.

Faute de temps, cette partie du traitement n'a pas été totalement mise en place, sa réalisation et sa fiabilité n'ont donc pas pu être entièrement vérifiées. La figure 4.19 montre, cependant, les rectangles englobants et les suites représentatives du nombre d'intersections obtenues pour les symboles musicaux les plus courants

L'analyse des suites obtenues révèle, d'ores et déjà, certains risques de confusion (par exemple entre bécarres et dièses): l'ajout d'informations, telle la longueur associée à chaque partie (définie par le nombre d'intersections) du symbole, devra permettre de résoudre ce type de problème.

Clé de sol



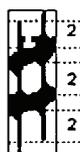
Respiration



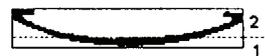
Bécarre



Dièse



Liaison



Soupir



Demi-soupir



Bémol

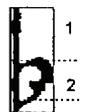


Figure 4.19: Symboles et intersections avec les droites horizontales.

Une fois le symbole reconnu, l'intérieur du rectangle englobant est mis à blanc dans l'image de la portée. Le parcours de l'image, à la recherche d'une nouvelle composante connexe, peut alors se poursuivre. Il ne reste, normalement, après cette étape, aucun pixel noir dans l'image.

#### **4.5. Post-traitements :**

Durant l'analyse de chaque portée, ce ne sont pas les symboles musicaux et leur emplacement qui ont été stockés mais chacune des primitives qui les composent.

De manière à obtenir une liste cohérente et suivant l'ordre imposé par la méthode employée, le stockage des têtes de notes est d'abord réalisé, suivant leur abscisse respective dans l'image (selon l'ordre croissant). Un code sur 8 bits est attribué à chaque tête de note, représentatif, à la fois de son type (Noire, Blanche, Noire pointée, ou Blanche pointée) et de sa hauteur sur la portée (Do, Ré, Mi, ...).

La présence d'une queue de note est indiquée par insertion, dans la même liste, d'un nouvel octet juste après le code représentant l'ellipse à laquelle elle correspond. Il vient alors la règle suivante: tout octet représentant une queue de note est précédé, dans la liste, du code décrivant son ellipse associée. La réciproque est fautive puisque les Rondes ne possèdent pas de queues. Il est, de plus, possible que l'on n'ait pas détecté la queue d'une Noire.

Durant la phase d'analyse de l'extrémité de chaque queue (détection du type de barre de croche), le code marquant la présence d'une queue est modifié afin d'indiquer, en plus, la nature de la barre (noire simple, croche, double-croche, ...).

Un codage du même type peut être réalisé pour les barres de mesures et les autres symboles. Tous seront ensuite insérés dans la liste suivant leur abscisse dans l'image.

Dans un premier temps, la phase de post-traitements devra vérifier la cohérence musicale de la chaîne obtenue, pour tenter de supprimer des éventuelles erreurs (présence d'une note à la même abscisse qu'une clé, clé en fin de portée, ...), ou tout au moins avertir l'utilisateur qu'une incohérence a été rencontrée (problème dans la rythmique, Noire non suivie d'une queue de note, ...) afin qu'elle puisse être corrigée.

Cette phase peut nécessiter, au préalable, la définition d'une "grammaire musicale" spécifiant les règles syntaxiques que devra vérifier la chaîne des codes, et donc la partition.

Une fois la chaîne finale obtenue, il est possible d'effectuer un transcodage, de façon à fournir à l'application hôte une entrée compréhensible.

Il s'agira donc dans la plupart des cas, de générer un fichier:

- soit répondant à la norme MIDI,
- soit utilisable par un éditeur de partitions, c'est à dire suivant un MRL (Music Representational Language),
- soit alimentant un autre processus (traduction en braille, ...)

## 4.6. Résultats et perspectives:

Ce système expérimental, nous a permis, par combinaison et mise en place de plusieurs méthodes, d'appréhender les différentes phases du processus complet de reconnaissance de partitions musicales et les problèmes qui en relèvent. L'implémentation a été réalisée sur un ordinateur PC 386 (équipé d'un coprocesseur mathématique) en langage C. Les tests ont été réalisés, pour l'instant, sur des partitions imprimés de difficulté moyenne. Les résultats, pour l'instant tous qualitatifs, se sont avérés, sur ce type de partitions, très satisfaisants et les temps de calcul tolérables.

Cependant, il reste certains points que nous n'avons pas eu le temps d'approfondir et qui restent à améliorer notamment (figure 4.20):

- la détection des notes très éloignées des lignes de portées,
- l'étude de l'intérieur du masque des têtes de notes,
- la restauration des symboles (altérés lors de l'élimination des lignes de portées).



Figure 4.20: Des problèmes.

Ces problèmes risquent de se multiplier et d'exiger des traitements nouveaux lorsque nous aborderons, après les partitions imprimées, celles écrites à la main. Certains problèmes seront aisés à résoudre, lors de futurs travaux, par simples modifications des algorithmes existants (notes éloignées, étude du masque). Pour ce qui concerne la restitution des symboles, il faudra choisir entre effectuer une meilleure segmentation ou utiliser des techniques de reconstruction de symboles fragmentés.

Les bons résultats obtenus lors des détections de lignes de portées, têtes, tiges et barres de notes, laissent alors à penser, qu'une fois ce problème réglé, une adaptation à la reconnaissance de partitions plus complexes sera envisageable.

## **5. Conclusion :**

Cette étude, tout d'abord basée sur les points communs liant Reconnaissance de l'écriture et Reconnaissance de partitions musicales, a en fait permis de souligner les nombreuses différences qui existent entre ces deux domaines (structure des documents, symboles, traitements, ...).

La reconnaissance automatique de partitions se heurte à des problèmes nouveaux, pour la plupart dus à la présence des lignes de portées, et que l'on ne retrouve pas en OCR. On ne peut donc pas prétendre que la reconnaissance optique de l'écriture englobe la reconnaissance des partitions musicales. Cette dernière constitue à elle seule un vaste sujet ayant, jusqu'à présent, été très peu exploré.

Le tour d'horizon des systèmes de lecture automatique de documents musicaux que nous avons effectué, a confirmé qu'il était possible d'utiliser durant certaines phases précises du traitement, des méthodes robustes, testées de nombreuses fois en reconnaissance de caractères, mais que la majeure partie du traitement nécessitait des procédures particulières faisant appel à des méthodes originales.

Il est à noter, bien-sûr, que le système expérimental, que nous avons partiellement mis en place, se limite à exposer l'intérêt et les limites des méthodes choisies. En effet nous n'avons malheureusement pas eu le temps d'approfondir chacune des phases du traitement. La fiabilité des méthodes devra, en outre, être évaluée à partir de nombreuses autres partitions musicales, une fois le système global terminé.

Au terme de cette étude, on peut penser, devant la progression constante des matériels informatiques et les motivations supplémentaires apportées par l'attrait soudain du public pour la musique assistée par ordinateur, que les résultats actuels sont prometteurs pour l'avenir. Mais tout en étant optimiste, nous savons bien, qu' *"Il ne faut jamais vendre la peau de l'ours avant que l'on ne l'ait mis par terre"*.

# Bibliographie

- [Bélaïd91] : **A. et Y. Bélaïd.** *Reconnaissance des Formes, Méthodes et applications.* InterEdition 429 pages. 1991.
- [Bélaïd92] : **A. Bélaïd et K. Tombre.** *Analyse de document: de l'analyse à la sémantique.* Actes de CNED'92, Traitement de l'écriture et des documents - BIGRE 80. p 3-29. Nancy 1992.
- [Besançon88] : **J.E. Besançon.** *Vision par ordinateur en deux et trois dimensions.* Eyrolles. Classifieur par distance minimale. p 124-125. 1988.
- [Boukined91] : **L. Boukined, B. Taconet, A. Zahour, et A. Faure.** *Recherche de la structure physique d'un document par rectangulation.* Actes 8ème congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle. Vol 3. p 1027-1032. Lyon-Villeurbanne 1991.
- [Carter88] : **N.P. Carter, R.A. Bacon, et T. Messenger.** *The Acquisition, Representation and Reconstruction of Printed Music by Computer: A Review.* Computer and the humanities. Vol 22. p 116-136. 1988.
- [Chauvet91] : **P. Chauvet.** *Systèmes d'analyse, reconnaissance et description de documents complexes.* Actes 8ème congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle. Vol 3. p 1033-1044. Lyon-Villeurbanne 1991.
- [Dengel92] : **A. Dengel et S. Baumann.** *Transforming printed piano music into MIDI.* Preprint of an article in Proceedings of the IAPR Workshop on SSPR'92. Bern 1992.

- [Ingold92] : **A. Azokly, A. Zramdini et R. Ingold.** *Reconnaissance de la structure physique de documents composites.* Actes de CNED'92, Traitement de l'écriture et des documents. BIGRE 80. p 30-39. Nancy 1992.
- [Leplumey91a] : **I. Leplumey et J. Camillerapp.** *Coopération entre la segmentation des régions blanches et des régions noires pour l'analyse de partitions musicales.* Actes 8ème congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle. Vol 3. p 1045-1052. Lyon-Villeurbanne 1991.
- [Leplumey91b] : **I. Leplumey et J. Camillerapp.** *Comparaison des méthodes d'étiquetage de régions d'une partition musicale.* Proceedings of the First International Conference on Document Analysis. Vol 2. p 674-682. St Malo 1991.
- [Marthon79] : **P. Marthon, A. Bruel et G. Biguet.** *Squelettisation par calcul d'une fonction discriminante sur un voisinage de 8 points.* Actes du 2ème congrès AFCET : Reconnaissance des Formes et Intelligence Artificielle. p 107-114. Toulouse 1979.
- [Martin89] : **P. Martin.** *Reconnaissance de partitions musicales et réseaux de neurones: une étude.* Actes 7ème congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle. p 217-226. Paris 1989.
- [Martin90] : **P. Martin et C. Bellissant.** *Segmentation et classification par réseau pour la reconnaissance de partitions.* Colloque de Reconnaissance Automatique de l'Écrit - BIGRE 68. p 102-111. 1990.
- [Matsushima88] : **T. Matsushima.** *Automatic Printed-Music-to-Braille Translation system.* Journal of information processing. Vol 11 n°4. p 249-257. 1988.
- [Miclet84] : **L. Miclet.** *Méthodes structurelles pour la reconnaissance de formes.* Eyrolles. 184 pages. Paris 1984.

- [Roads86] : **C. Roads.** *The Tsukuba Musical Robot.* Computer music journal. Vol 10 n° 2. p 39-43 1986.
- [Schweizer87] : **P. Schweizer.** *Infographie 1: Introduction, Bibliothèques et périphériques graphiques, Trassage, remplissage.* Presses Polytechniques Romandes. Algorithme incrémental optimisé de Bresenham. p 255-265. 1987.
- [Sicard92] : **E. Sicard.** *An efficient method for recognition of printed music.* IEEE proceedings of the 11th International Conference on Pattern Recognition. p 573-576. Den Haag 1992.
- [Simon84] : **J. C. Simon.** *La reconnaissance de formes par algorithmes.* Masson. 251 pages. Paris 1984.
- [Tatem88] : **J.W. Roach et J.E. Tatem.** *Using domain knowledge in low-level visual processing to interpret handwritten music: An experiment.* Pattern Recognition. Vol 21 n°1. p 33-44. 1988.
- [Vo-Van77] : **T. Vo-Van.** *Contribution à la conception d'un système de reconnaissance de caractères non stylisés.* Thèse de docteur en mathématiques. 122 pages. Besançon 1977.
- [Watson84] : **L.T. Watson, K. Arvind et R.W. Ehrich.** *Extraction of lines and regions from Grey Tone Line Drawing Images.* Pattern Recognition. vol 17 n° 5. p 493-507. 1984.
- [Wong82] : **K.Y. Wong, R.G. Casey, et F.M. Wahl.** *Document Analysis System.* IBM journal Research Development . vol 26 n°6. p 678-683. 1982.