Rapport de stage du Master 2 Signal Image Parole Télécoms
Année 2004/2005

Alignement d'un enregistrement Audio avec sa Partition : passage de l'algorithme DTW global à un DTW à court terme

---

Hagen Kaprykowsky*

Master SIPT (Signal Image Parole Télécommunications)
Option : ATIS (Analyse et Traitement des Images et des Signaux)
ENSIEG, Ecole Doctorale EEATS et Universität Karlsruhe (TH)

Réalisé à :
Ircam (Institut de Recherche et de Coordination Acoustique/Musique)
Place Igor Stravinsky
75004 Paris

de Février à Juillet 2005

Responsable Ircam : Xavier Rodet
Responsable ENSIEG : Jocelyn Chanussot
Responsable IIIT Universität Karlsruhe (TH) : Benedikt Merz

*<hagen.kaprykowsky@ircam.fr>

Informations générales :

Ce rapport donne un aperçu du travail de recherche que j'ai effectué entre Février et Juillet 2005, à :

Ircam (Institut de Recherche et de Coordination Acoustique/Musique)
Place Igor Stravinsky
75004 Paris
www.Ircam.fr

Ce travail correspond au stage du Master 2 Signal Image Parole Télécommunications (SIPT) et également au projet de fin d'études pour les enseignements dispensés par l'Ecole Nationale Supérieure d'Electricité de Grenoble (ENSIEG) en Analyse et Traitement des Images et des Signaux (ATIS), et par l'Université Karlsruhe (TH) en Analyse et Traitement des Images et des Signaux. (Double diplôme)

Mon responsable de stage est Xavier Rodet (Responsable de l'équipe analyse synthèse)

Le travail effectué lors de stage consiste en la participation active à l'amélioration de l'algorithme d'alignement de partitions de l'équipe analyse synthèse de l'Ircam, comprenant :

1. Invention de l'algorithme STDTW (Short Time Dynamic Time Warping)
2. Implémentation de l'algorithme et intégration dans le projet existant en Matlab
3. Test de l'algorithme
4. Coopération avec différents projet de l'Ircam

Pour informations complémentaires :

Hagen Kaprykowsky
9, Place de la République
75003 Paris
hagen.kaprykowsky@ircam.fr

Résumé :

L'alignement de partition permet de lier les événements d'une partition musicale avec des instants sur l'axe temporel de l'audio d'une interprétation de cette partition. Parmi l'ensemble de ces événements, on s'intéresse particulièrement au temps de début de chaque note ou accord, fins de notes, hauteur exacte, timbre, etc. L'alignement s'effectue en temps differé, en utilisant tout l'enregistrement audio et sa partition. L'alignement trouve des applications en synthèse (recoder exactement une interprétation d'une partition, avec toutes ses nuances, par exemple dans un fichier MIDI enrichi pour fournir une synthèse de haute qualité), en musicologie (étudier diverses interprétations d'une même oeuvre), en pédagogie (montrer à l'élève une image de son interprétation), en indexation (constitution de bases de données d'apprentissage), etc.

Un programme d'alignement a été écrit dans l'équipe analyse synthèse de l'Ircam Centre Pompidou. Il utilise un algorithme de Dynamic Time Warping (DTW) sur tout l'enregistrement audio. Ceci est optimal mais limite la taille des morceaux possibles et est très coûteux en temps de calcul.

Le problème est donc de remplacer l'optimisation DTW globale, c'est à dire sur l'ensemble du morceau, par une optimisation à court terme, c'est à dire sur une fenêtre de durée bien inférieure, et de faire glisser cette fenêtre du début jusqu'à la fin du morceau afin d'obtenir a peu près le même résultat optimal.

Nous avons inventé un nouvel algorithme de DTW, nommé STDTW (Short Time Dynamic Time Warping). Il permet donc en théorie d'aligner des morceaux aussi long qu'on veut. Mais mieux encore, il est aussi optimal que l'algorithme global. Enfin, s'il était utilisé dans un contexte "temps réel" il fournirait le résultat de l'alignement avec un delai aussi faible que possible, c'est à dire dès que l'on a "entendu" suffisamment de l'audio pour être certain de l'alignement.

mots-clé :

alignement à court terme, alignement musical, programmation dynamique, recalage temporel, Dynamic Time Warping

Abstract :

Music alignment links events in a score and points on the audio performance time axis. Score alignment can use the whole score and the whole audio file if needed to perform the task. An alignment implies a segmentation of the performance according to the events in the score. All the parts of a recording can be thus indexed according to score information by labeling (tagging) with the information from the score to build up unit databases. Along with the note's pitch and length, there can be additional symbolic information attached to the score, such as dynamics, articulation, or lyrics. Therefore, it is a powerfully tool to compare different performances which can be used for musicological research, for instance aiming at the study of the expressive parameters related to timing. It is also possible to index continuous media through segmentation for content-based retrieval. The total alignment cost between pairs of documents can be considered as a distance measure (as in early works on speech recognition), allowing to find the best matching documents from a database.

To perform the alignment automatically, an algorithm has been developed at the Ircam Centre Pompidou. The alignment is carried out using DTW (Dynamic Time Warping), which finds the best alignment between two sequences according to a number of constraints, using the entire performance. This method leads to optimal results, but limits the size of the pieces because of its high memory and calculation costs.

The problem is now to replace the global optimization for the entire audio file by a short time optimization, where a smaller window is slided from the beginning to the end, obtaining approximately the same result.

In this work, we present a new Dynamic Time Warping algorithm which is called STDTW (Short Time Dynamic Time Warping). It permits theoretically to align a piece which is arbitrary long and obtains the same results as the global DTW algorithm. Finally, within a "real time" context, it furnishes the alignment result with a delay as short as possible, once we have "heard" enough of the audio file to be certain.

keywords :

dynamic programming, Dynamic Time Warping, short time alignment, musical alignment

Remerciements :

Je tiens à remercier tout particulièrement Xavier Rodet, mon maître de stage, ainsi que Axel Roebel, Joseph Escribe et Philippe Bernat-y-vicens chercheurs de l'équipe dans laquelle j'étais intégré, qui ont su m'orienter tout au long de mon stage et me donner un peu de leur temps pour m'aider dans ma mission. Sans oublier toute l'équipe Analyse Synthèse de l'Ircam qui m'a permis d'effectuer mon stage dans d'excellentes conditions de travail. Je remercie également tout mon entourage, particulièrement mes collègues de promotion à qui je tiens à exprimer toute ma sympathie pour avoir partagé plusieurs années d'études enrichissantes ensemble. Finalement, je remercie ma famille, pour m'avoir apporté un soutien matériel plus que généreux, ainsi qu'une oreille attentive et des encouragements constants.

Table des matières

Table des figures

1. Introduction

Le stage de fin d'étude est l'une des étapes importante du cursus d'ingénieur. En effet, outre les différents stages effectués jusqu'ici, il est le premier correspondant sensiblement aux travaux effectués par un ingénieur. Les cours théoriques sont terminés, et le travail proposé demande les qualités requises d'un ingénieur : curiosité, rigueur, adaptabilité, sens de la communication. J'ai effectué mon stage dans un milieu particulier : celui de la recherche scientifique à l'Ircam, l'Institut de Recherche et de Coordination Acoustique / Musique. Cela m'a permis de découvrir un milieu de travail en perpétuelle ébullition, que ce soit pour la production de musique ou pour les recherches théoriques fondamentales. Ce rapport, est une synthèse du travail effectué pendant cinq mois à l'Ircam de février à juillet 2005. Nous verrons dans cette introduction des éléments clefs permettant au rapport de prendre tout son sens.

1.1. L'alignement de partition en bref :

> L'alignement de partition permet de lier les événements d'une partition musicale avec des instants sur l'axe temporel de l'audio d'une interprétation de cette partition.

1.2. Plan du rapport :

Ce rapport présentera dans un premier temps les supports informatique pour stocker des informations audio et le format MIDI qui sont utilisées pour l'alignement musical, puis nous rappellerons les objectifs du stage avec une description détaillée sur le problème d'alignement de partition. Nous verrons ensuite une exploration détaillée de l'algorithme DTW. Puis nous verrons le nouvel algorithme basé sur le DTW à court terme, qui est inventé dans ce stage. Nous verrons ensuite les travaux qui étaient éffectués en coopération avec différents projets avant de conclure.

1.3. Les fichier audios et midi

L'alignement de partition peut être divisé en deux catégories, le signal (audio) et les notes (midi). La première prend en considération les erreurs possibles dues à l'algorithme

d'analyse ; à l'inverse, la seconde suppose une bonne estimation des paramètres, provenant en général d'une estimation Midi de la hauteur, et se concentre sur les erreurs de l'interprète.

1.3.1. Fichiers audios

Il existe de nombreux supports informatiques pour stocker des informations audio. Parmi eux, les plus connus sont les WAV. Le nom technique du format WAV est le PCM, pour le Pulse Code Modulation.

Les caractéristiques principales des fichiers audios du type wav sont :

1. Fréquence d'échantillonnage, c'est à dire nombre d'échantillons par seconde (e.g 11025 qualité téléphone, 22050 qualité radio, 44100 qualité CD, 48000 qualité DAT)
2. Nombre de bits par échantillon (Lors de l'échantillonnage, chaque valeur d'amplitude est convertie sur une échelle déterminée par le nombre de bits utilisés, en général 8 ou 16, certaines cartes son ne lisent pas autre chose)
3. Mono ou Stéréo

Il existe d'autres formats comme MP3, AIFF, VQF et bien d'autres. Pour des besoins pratiques, nous convertirons tous les fichiers audios retenus en WAV qui est le format qui se manipule le mieux sous Matlab.

1.3.2. Fichiers midi

MIDI signifie Musical Instrument Digital Interface, c'est un format d'échange destiné aux instruments numériques.

Ce format n'échange pas la musique elle-même, mais uniquement une description des actions des musiciens. Ainsi, lorsqu'un CD musical, pour décrire une note, va donner l'intensité du signal à intervalles très courts (44100 fois par secondes - 44.1 kHz), dans le format MIDI cette note sera décrite par un chiffre donnant le ton de la note, et son intensité, et quelques autres informations. Le timbre de la note n'est donc pas communiqué, seul l'ordre de jouer cette note est transmis.

En plus, sont communiqués le numéro du canal sur lequel il faut l'interpréter (à chaque canal correspond un instrument) avec en plus éventuellement quelques informations (toujours très courtes, quelques octets) pour des informations de synchronisation, des glissendo, l'aftertouch (seconde pression sur une touche sans la relâcher), des vibratos...

Ce format nous a permis de nous procurer des partitions car nous connaissons les notes et leur durées, on peut donc se servir de cela comme une base de partition pour l'alignement de partitions. De plus il est possible de trouver relativement facilement des fichiers midi sur internet.

chapterL'alignement de l'Ircam : Etat de l'art

L'alignement de partition permet de lier les événements d'une partition musicale avec des instants sur l'axe temporel de l'audio d'une interprétation de cette partition. Parmi l'ensemble de ces événements, on s'interesse particulièrement au temps de début de chaque note ou accord, fins de notes, hauteur exacte, timbre, etc. L'alignement s'effectue en temps differé, en utilisant tout l'enregistrement audio et sa partition. L'alignement trouve des applications en synthèse (recoder exactement une interprétation d'une partition, avec toutes ses nuances, par exemple dans un fichier MIDI enrichi pour fournir une synthèse de haute qualité), en musicologie (étudier diverses interprétations d'une même oeuvre), en pédagogie (montrer à l'élève une image de son interprétation), en indexation (constitution de bases des données d'apprentissage), etc.

L'alignement de musique correspond donc à l'association d'évènements dans une partition avec des points temporels d'un signal audio. Le signal est donc segmenté selon les évènements de la partition. Une méthode d'alignement automatique a été développée, en utilisant la méthode de Dynamic Time Warping (DTW). C'est sur cette méthode que mon projet se base. Elle utilise la structure des pics spectraux, augmentée par un modèle d'attaques et de silence. Cette technique peut traiter des signaux audio considérés comme difficiles à aligner, comme la musique polyphonique, des trilles, ou des séquences rapides. Cette méthode a été décrite et présentée dans [Orio01b].

1.4. Les applications de l'alignement de partitions

Une grande partie de la recherche en informatique est consacrée à l'automatisation des processus réalisés par des humains. Les processus automatiques sont particulièrement utiles dans un certain nombre de situations. Par exemple, la segmentation d'une grande base de donnée d'enregistrements, qui peuvent durer plusieurs heures, ne peut pas être faite manuellement en raison de la trop grande quantité de données à traiter. La même situation s'applique pour les signaux compliqués (c'est à dire, d'ordres rapides, comprenant des notes avec legato) où la segmentation manuelle peut être pénible ou imprécise. L'alignement automatique des ordres de musique a un certain nombre d'applications, les plus importantes étant :

1. Segmentation d'une exécution en notes et étiquetage des notes avec l'information des temps de début et fin (points) pour construire des bases de données d'unités. En plus de la hauteur et la longueur des notes, il peut y avoir de l'information symbolique additionnelle attachée aux points, telle que la dynamique, l'articulation etc..
2. La comparaison de différentes exécutions pour la recherche en musicologie, par exemple ayant pour objectif l'étude des paramètres expressifs reliés à la synchronisation et au timing.

3. Indexation des médias continus par la segmentation pour la recherche de contenu. Tout le coût d'alignement entre les paires de documents peut être considéré comme mesure de distance (comme dans les premiers travaux sur la reconnaissance de la parole), permettant de trouver les meilleurs documents correspondant à la recherche effectuée, issus d'une base de données.

L'alignement est lié au problème de la synchronisation en temps réel entre les interprètes et les ordinateurs, habituellement appelé 'score following', pour des contraintes additionnelles de faible latence et seulement lorsqu'une connaissance locale de l'exécution est présente. L'alignement off-line peut être employé comme procédé de circuit fermé pour la formation des modèles statistiques en temps réel.

Lorsqu'on parle d'exécution, cela correspond à l'interprétation et à la réalisation d'une oeuvre musicale. Et c'est en cela que réside une part de la difficulté de l'alignement de partition, car chacun explore différemment la même oeuvre, mais les résultats informatiques doivent être optimum dans tous les cas.

1.5. Les étapes de l'alignement de partition :

1. Lecture du signal audio / Construction de la partition par lecture du fichier midi correspondant
2. Détection des percussions de type "kick" et "snare"
3. Alignement à partir de l'algorithme de DTW
4. Estimation précise des attaques des notes harmoniques alignées
5. Algorithme de "Beat control" pour préciser les trames d'attaques (Beat Control) suivi d'une nouvelle estimation des attaques de notes harmoniques
6. Création du nouveau fichier midi avec les notes (harmoniques et percussives) alignées

Tous les développements de l'alignement de partition ont été réalisés à l'Ircam par Nicola Orio, Diemo Schwarz, Férréol Soulez, Sébastian Durigon, Joseph Escribe puis par Phillipe Bernat-y-vicens, sous la direction de Xavier Rodet.

1. Orio, Schwarz : Algorithme de départ [Orio and Schwarz, 2001]
2. Soulez, Schwarz : Amélioration de l'Algorithme de départ [Soulez et al., 2003] et [Schwarz, 2004]
3. Durigon, Escribe : Détection des percussions, Estimation des onsets [Rodet et al., 2004]
4. Bernat-y-vicens : Amélioration de l'algorithme, Interface graphique, Départ de DTW à court terme

1.6. La partie d'alignement :

La partie de l'algorithme qui est traitée dans ce stage comprend les étapes 1 et 3. Les traitements associés sont présentés dans la figure 1.6 qui correspond aux étapes suivantes :

1. extraction des paramètres audio du signal (Analyse du signal) à partir de la performance (fichier audio) et de la partition
   a) Analyse du signal : FFT du signal
   b) Model des notes : Filtre passe bandes contenant les h premières harmonique des notes attendues d'après la partition.
   c) Trames de la performance : Traduction de la performance en seconde ⇒ performance en trames
   d) Trames de la partition : Traduction de la partition en seconde ⇒ partition en trames
2. Distance locale : à partir du modèle de note et du FFT du signal la distance locale entre tous les instants de la partition et de l'interprétation est calculé. C'est la grandeur d'entrée de l'algorithme DTW (explication de la distance local ⇒ chapitre 2)
3. Il y a trois différentes méthodes de calcul de la distance locale :
   a) attaque (AD)
   b) tenue (PSD)
   c) silence (SD)
4. A partir de la distance locale pour la trame courante dans le fichier audio l'alignement est fait par l'algorithme DTW (explication de l'algorithme DTW ⇒ chapitre 2)
5. On appelle chemin le resultat du DTW. Ce chemin construit une correspondance entre les trames dans le fichier audio et les trames dans la performance.

1.7. Sujet : Passage de l'algorithme DTW global à un DTW à court terme

Le programme d'alignement utilise un algorithme de Dynamic Time Warping (DTW) sur tout l'enregistrement audio. Ceci est optimal mais limite la taille des morceaux possibles et très coûteux en temps de calcul.

Le problème est donc de remplacer l'optimisation DTW globale, donc sur l'ensemble du morceau, par une optimisation à court terme, c'est à dire sur une fenêtre de durée bien inférieure, et de faire glisser cette fenêtre du début jusqu'à la fin du morceau afin d'obtenir à peu près le même résultat optimal.

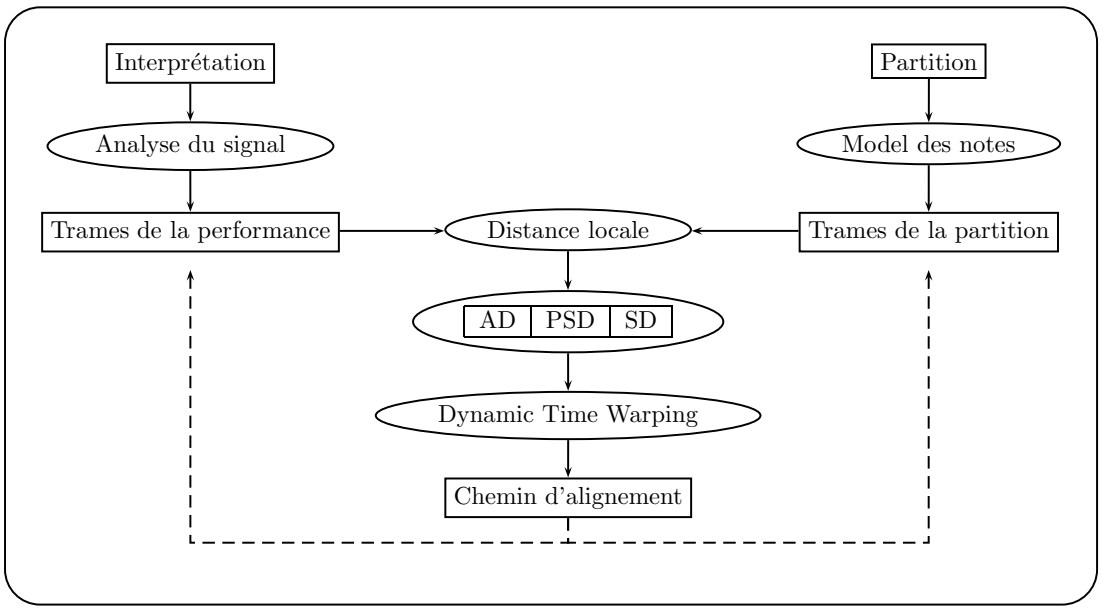


FIG. 1.1.: L'alignement musical : L'algorithme DTW

2. Dynamic Time Warping (DTW)

Le DTW a été largement utilisé dans la reconnaissance de la parole pour aligner des segments de parole de locuteurs différents. La stratégie appliquée par l'algorithme est très similaire à la technique employée pour la comparaison de textes. Le DTW trouve le meilleur appariement entre une référence (la partition) et un signal (l'interprétation) par le calcul d'une différence entre des vecteurs des caractéristiques pour chacun de ces signaux. Les algorithmes de comparaison de chaînes de caractère sont basés sur la correspondance exacte entre la référence et le signal et ne prennent pas en compte, entre autre, l'imprécision de l'estimation du pitch provenant d'accords ou d'erreurs de l'algorithme de détection de hauteur. En outre, le DTW peut être utilisé pour aligner des caractéristiques multidimensionnelles continues, par exemple les résultats extraits d'une analyse du signal, qui permettent de baser l'alignement de partition sur des paramètres acoustiques différents de la hauteur et ne requiert pas une segmentation préalable du signal.

En bref, l'algorithme DTW consiste en trois étapes :

1. Calcul des distances locales
2. Programmation dynamique pour obtenir l'optimum global
   a) Calcul des distances augmentées. On ne garde que les prédécesseurs minimaux de chaque point (principe de l'optimalité) ⇒ optimum local
   b) Backtrack pour trouver la distance minimale ⇒ optimum global
3. Résultat : chemin de recalage qui consiste en la correspondance des deux séquences

[Rabiner and Juang, 1993] [Di Martino, 1984]

2.1. Principe de DTW (recalage temporel)

Comme discuté précédemment, le problème d'alignement musical peut être considéré comme une comparaison de deux séquences $\mathcal{X} = (x_1, x_2, \ldots, x_N)$ (partition) et $\mathcal{Y} = (y_1, y_2, \ldots, y_M)$ (interprétation) de longueurs différents N et M, où l'on détermine les correspondances entre les points (x_n, y_m) *avec* $n = 1, 2, \ldots, N$ *et* $m = 1, 2, \ldots, M$ de façon à minimiser un certain critère d'erreur. Le DTW permet d'effectuer une synchronisation des échelles des temps de deux formes à comparer comme le montre la figure 2.1.

[Keogh, 2002]

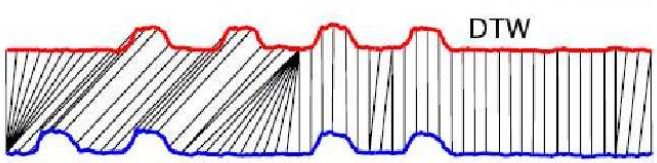


FIG. 2.1.: La distance DTW

2.2. Le chemin de recalage

Il s'agit donc de trouver parmi tous les Φ ainsi définis :

$$n = \Phi_x(k) \quad avec \quad k = 1, 2, \ldots, K \tag{2.1}$$

$$m = \Phi_y(k) \quad avec \quad k = 1, 2, \ldots, K \tag{2.2}$$

$$\Phi = (\Phi_x, \Phi_y) \tag{2.3}$$

où Φ indique que la fonction de recalage met en coïncidence le $\Phi_x(k)^{ieme}$ prélèvement de la forme $\mathcal{X}$ et le $\Phi_y(k)^{ieme}$ prélèvement de la forme $\mathcal{Y}$. La fonction optimale $\widehat{\Phi}$, au sens d'une métrique, réalise la coïncidence optimale entre les deux formes à comparer. Si l'on porte sur un axe horizontal le vecteur concernant les paramètres de la partition $\mathcal{X}$, et sur l'axe vertical le vecteur concernant les paramètres de l'interprétation $\mathcal{Y}$, une représentation de la fonction Φ est un chemin dans le plan ainsi défini. La figure 2.2 illustre un exemple de chemin de recalage.

2.3. Les contraintes imposées aux chemins de recalage

Monotonie

Afin que la fonction de recalage respecte l'évolution dans le temps du signal musical, celle-ci est soumise à conditions de monotonie exprimées par les relations suivantes :

$$\Phi_x(k+1) \geq \Phi_x(k) \tag{2.4}$$

$$\Phi_y(k+1) \geq \Phi_y(k) \tag{2.5}$$

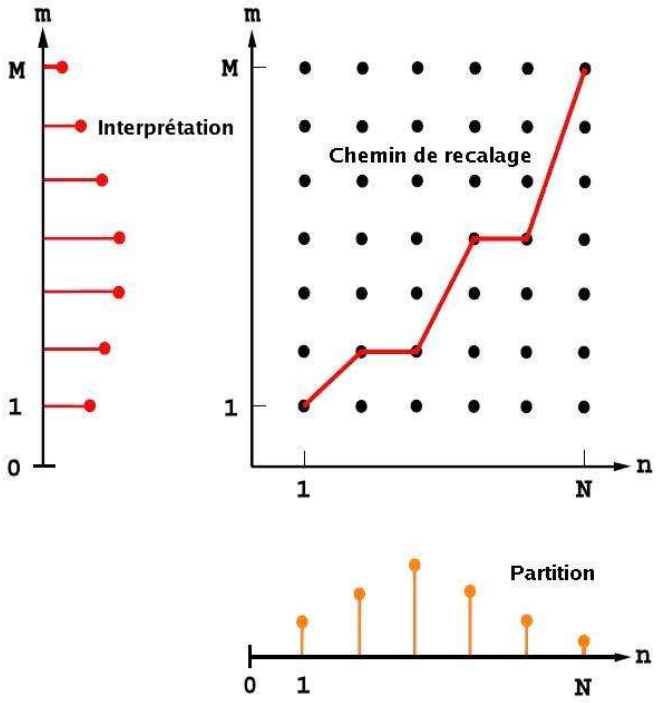
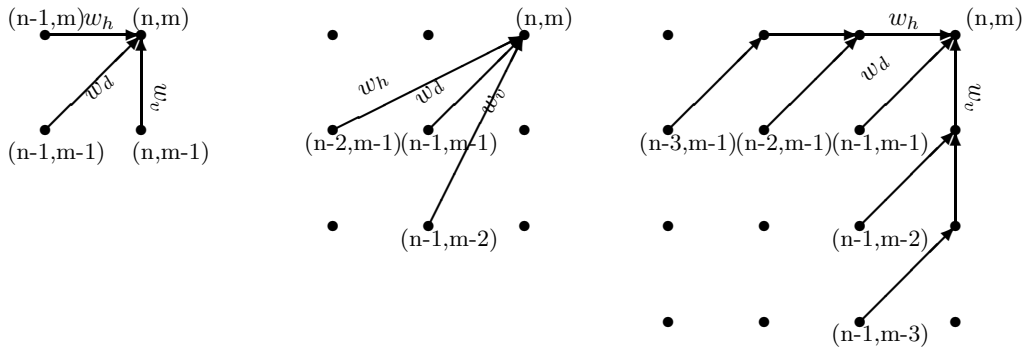


FIG. 2.2.: Chemin de recalage



Type I Type III Type V

FIG. 2.3.: Trois exemples de contraintes qui sont très souvent utilisées

Continuité

La fonction de recalage se doit aussi de ne pas effectuer des compressions ou dilatations irréalistes. Pour qu'il en soit ainsi, celle-ci est assujettie à des contraintes locales qui lui empêche d'effectuer certains déplacements locaux. On interdit souvent par example au chemin de recalage d'aller consécutivement deux fois dans la même direction si le sens du déplacement est horizontal ou vertical. La figure 2.3 montre trois exemples de contraintes qui sont très souvent utilisées.

$$\Phi_x(k) - \Phi_x(k-1) \leq 1 \tag{2.6}$$

$$\Phi_y(k) - \Phi_y(k-1) \leq 1 \tag{2.7}$$

(w_d, w_h et w_v) sont des poids associées aux chemins (diagonal, horizontal et vertical). Dans le cas d'alignement musical le Type V est le type de voisinage le plus adapté parce qu'il ne permet pas de déplacement horizontal ni vertical (notes oubliées ou ajoutées) mais il tolère suffisamment d'éloignement.

2.3.1. Limites

D'autre part, afin que la fonction de recalage tienne compte de l'ensemble des prélèvements de chacune des deux formes, des contraintes aux fronières lui sont imposées. Ces dernières

l'assujettissent à mettre en correspondance les prélèvements terminaux de chacune des deux formes.

Ces contraintes sont exprimées par les relations :

$$\Phi_x(1) = 1 \qquad \Phi_y(1) = 1 \tag{2.8}$$

$$\Phi_x(K) = N \qquad \Phi_y(K) = M \tag{2.9}$$

N et M désignant les nombres de coïncidences effectuées par le chemin de recalage.

2.3.2. Principe de la programmation dynamique appliquée à la recherche par le chemin de la fonction de recalage optimal

Le recalage temporel, comme nous l'avons vu, a pour but de déterminer la fonction de recalage $\Phi = (\Phi_x, \Phi_y)$ qui maximise les meilleurs coïncidences entre les deux formes à comparer ou encore, comme nous l'avions esquissé, qui minimise une certaine métrique.

Le problème de l'alignement temporel peut se résoudre simplement en explorant tous les chemins possibles. Malheureusement, le nombre de chemins possibles croit exponentiellement avec la longeur des signaux à comparer. Toutefois, le problème peut être résolu de manière efficace par un algorithme de comparaison dynamique qui va mettre en correspondance optimale les échelles temporelles des deux séquence.

2.4. Distance cumulée

La fonctionelle que l'on associe habituellement à une fonction de recalage Φ est donné par la relation :

$$d_\Phi(\mathcal{X}, \mathcal{Y}) = \sum_{k=1}^{K} d(\Phi_x(k), \Phi_y(k)) p(k) / P_\Phi \tag{2.10}$$

$$avec\ P_\Phi = \sum_{k=1}^{K} p(k) \tag{2.11}$$

$p(k)$ est un facteur de ponderation qui diffère suivant la transition locale $(\Phi_x(k-1), \Phi_y(k-1)) \Rightarrow (\Phi_x(k), \Phi_y(k))$ et P_Φ est un facteur de normalisation dont le rôle est de rendre $d_\Phi(\mathcal{X}, \mathcal{Y})$ indépendant de la longueur du chemin recalage.

La fonction de recalage Φ qui nous intéresse est celle qui minimise la distance cumulée $d_\Phi(\mathcal{X}, \mathcal{Y})$. La solution du recalage temporel est donc donnée par la relation suivante :

$$\Phi = Arg \min_{\Phi} d_\Phi(\mathcal{X}, \mathcal{Y}) \tag{2.12}$$

2.5. Distance cumulée minimale

En fait pour effectuer une reconnaissance, Φ ne nous intéresse pas directement. L'information importante est le taux de dissemblance donné par la relation :

$$d(\mathcal{X},\mathcal{Y}) \quad = \quad \min_{\Phi_x,\Phi_y} d_\Phi(\mathcal{X},\mathcal{Y}) \tag{2.13}$$

On définit $D(N,M)$ comme ceci :

$$M_\Phi d(\mathcal{X},\mathcal{Y}) \quad \triangleq \quad D(N,M) \tag{2.14}$$

$$= \quad \min_{\Phi_x,\Phi_y} \sum_{k=1}^{K} d(\Phi_x(k),\Phi_y(k))p(k) \tag{2.15}$$

L'équation 2.13 peut être résolue par la programmation dynamique à l'aide du principe d'optimalité local introduit par Bellman [Bellman, 1957] :

> *Soit $C_{(1,1)}^{(n,m)}$ le chemin optimal joignant le points $(1,1)$ et (n,m), alors pour tout point $(n',m') \in C_{(1,1)}^{(n,m)}$, le chemin de recalage $C_{(1,1)}^{(n',m')}$ est optimal.*

Désignons la distance cumulée optimal au point (n,m) associée à $C_{1,1}^{(n',m')}$ par $D(n,m)$:

$$D(n,m) \quad \triangleq \quad \min_{\Phi_x,\Phi_y,K'} \sum_{k=1}^{K'} d(\Phi_x(k),\Phi_y(k))p(k) \tag{2.16}$$

$$avec\ \Phi_x(1)=1,\ \Phi_y(1)=1 \quad et \quad \Phi_x(K')=n,\ \Phi_y(K')=m \tag{2.17}$$

D'après le principe d'optimalité local il vient :

$$D(n,m) \quad = \quad \min_{i'_x,i'_y}[D(n',m') + d((n',m'),(n,m)))] \tag{2.18}$$

où :

1. (n',m') appartient à un voisinage de (n,m) défini par la contrainte local utilisée
2. $d((n',m'),(n,m)))$ est la distance locale pondérée entre les points (n',m') et (n,m).

Le programme est réalisé en matlab. Les distances (locales et cumulées) sont gardés dans des matrices ldm(m,n) et adm(m,n). $ldm(m,n)$ correspond à la matrice des distances locales $d(n,m)$ et $adm(m,n)$ correspond à la matrice des distances cumulée $D(n,m)$. (index m lignes et index n colonnes).

2.6. Local Distance Model (LDM)

Les distances locales sont calculées à l'aide d'un modèle de note (attaque plus soutient). On construit un filtre passe bande contenant les h premières harmoniques des notes attendues d'après la partition. La distance entre les instants n de la partition et m de l'interprétation est :

2.6.1. Sustain

$$PSD(m,n) \quad = \quad 1 - \frac{\sum_i S_i P_i^2}{\sum_i P_i} \tag{2.19}$$

Avec : S : Passe bande correspondant a la partition P : Énergie de l'élément i de la FFT du signal

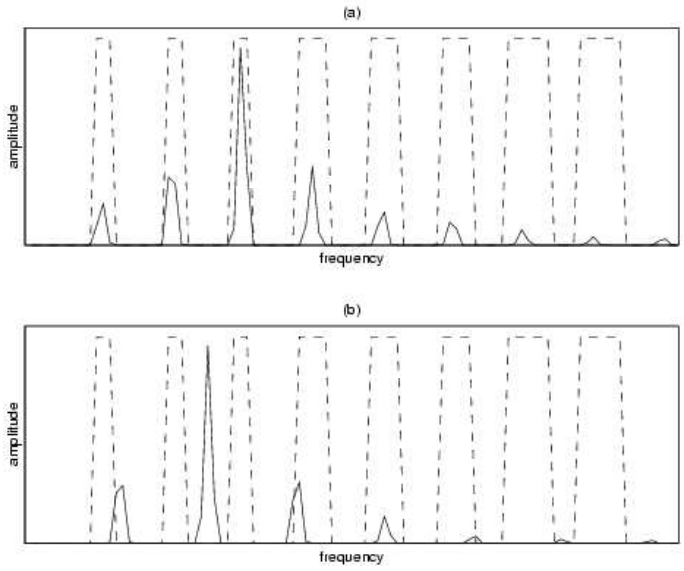


FIG. 2.4.: Passes bande générés d'après la partition avec une bonne (a) et une mauvaise (b) correspondance avec le spectre du signal.

Avec le modèle 'sustain' on obtient de trop grandes imprécisions sur le début des notes dûes à :

1. La réverbération des partiels de la note précédente
2. L'étalement de l'énergie sur tout le spectre lors de l'attaque.

3. La 'lenteur' de certaine attaques.
4. Dans les cas polyphoniques trop peu de partiels varient pour causer un changement de la structure du spectre détectable par le PSD.

2.6.2. Attaque

Le principe du modèle d'attaque utilise les variations d'énergie de chaque note. Lors du calcul de PSD, on somme les valeurs absolues des variations d'énergie dans chaqun des filtres harmoniques de la note dont on cherche le commencement. Cette variation n'est pas la dérivée de l'énergie mais la différence avec l'extremum local précedant, ce qui permet d'observer ces variations à plus long terme, ainsi même si l'attaque consiste en une lente montée de l'énergie elle sera quand même détectée. De plus, utiliser la somme des valeurs absolues dans les filtres permet de prendre en compte les baisses d'énergie lors de l'attaque due aux battements dans certaines bandes de fréquences ; la baisse d'énergie d'une fin de note n'étant pas synchrone sur tous les filtres à cause, entre autres, de la réverbération (les aiguës vont s'atténuer plus vite que les graves...) qui ne sera pas détectée comme une attaque.

Prise en compte de la variation d'énergie dans chacun des filtres : En début de note, on observe, dans chacun des filtres, la différence entre l'énergie courante et son extremum précédant.

$$AD(m,n) \quad = \quad mean_k \left(1 - tanh\left(\alpha\left(\sum_{i=1}^{F_n} \|\Delta_i^k\| - \Theta_a\right)\right)\right) \tag{2.20}$$

avec $\|\Delta_i^k\|$ la différence d'énergie en dB d'avec le maximum local précédent dans le bande filtre i de la note k, θ_α un seuil et α un facteur correctif.

2.6.3. Silence

A cause de la réverbération il est difficile d'identifier les courtes pauses. Les pauses inférieures à 100ms sont donc fusionnées avec la notes précédentes Les pauses plus longues sont détectées à l'aide d'un seuil (θ_s) sur l'énergie (E).

$$SD(m,n) \quad = \quad \begin{cases} E - \theta_s & \text{if } E \geq \theta_s \\ 0 & \text{if } E < \theta_s \end{cases} \tag{2.21}$$

$$avec\ E \quad = \quad log \sum_{i=1}^{N_{FFT}} P_i \tag{2.22}$$

2.6.4. ldm(m,n)

$$ldm(m,n) = \begin{cases} AD(m,n) & \text{if } n \in A \\ SD(m,n) & \text{if } n \in S \\ PSD(m,n) & \text{otherwise} \end{cases} \quad (2.23)$$

2.7. Accumulated Distance Model (ADM)

Comme nous avons vu précédemment le problème de l'alignement peut être résolu efficacement par la programmation dynamique. Il suffit alors, pour chaque point de l'espace d'évaluer la meilleure manière d'entrer dans cet état en respectant les contraintes et en minimisant la contribution à la distance globale.

Il suffit donc d'étudier les transitions autorisées et d'appliquer la relation récursive locale : (avec $\lambda = ldm(m,n)$)

Type I :

$$adm(m,n) = min \left\{ \begin{array}{l} adm(m-1,n-1) + w_d\lambda \\ adm(m-1,n) + w_v\lambda \\ adm(m,n-1) + w_h\lambda \end{array} \right\} \quad (2.24)$$

Type III :

$$adm(m,n) = min \left\{ \begin{array}{l} adm(m-1,n-1) + w_d\lambda \\ adm(m-2,n-1) + w_v\lambda \\ adm(m-1,n-2) + w_h\lambda \end{array} \right\} \quad (2.25)$$

Type V :

$$adm(m,n) = min \left\{ \begin{array}{l} adm(m-1,n-1) + w_d\lambda \\ adm(m-2,n-1) + w_v\lambda + w_d ldm(m-1,n) \\ adm(m-1,n-2) + w_h\lambda + w_d ldm(m,n-1) \\ adm(m-3,n-1) + w_v\lambda + w_d ldm(m-2,n) + w_v ldm(m-1,n) \\ adm(m-1,n-3) + w_h\lambda + w_d ldm(m,n-2) + w_h ldm(m,n-1) \end{array} \right\} \quad (2.26)$$

où adm(m,n) est la distance cumulée le long du chemin optimal allant de point (1,1) au point (m,n). adm(m,n) est évaluée sur tout le domaine qui est parcouru colonne par colonne ou ligne par ligne en partant du point (1,1).

L'algorithme de programmation dynamique général que nous venons d'expliquer met en évidence le fait qu'il faut en tout point (m,n) du plan de comparaison une distance cumulée partielle optimale D(n,m). Celle-çi d'après les relations de chapitre 2.7 nécessite

l'évaluation d'une distance locale $d(n, m)$ entre le n^{ieme} prélèvement de la forme $\mathcal{X}$ et le m^{ieme} prélèvement de la forme $\mathcal{Y}$. Cette dernière opération est très coûteuses en calcul. Ainsi est-il intéressant de limiter la zone de recherche du chemin de recalage optimal et par conséquent le nombre de points où il faut évaluer une distance locale. Myers C. S. Myers and Rosenberg [1980] a montré qu'en fonction de la contrainte locale à laquelle est assujettie la fonction de recalage il est possible de définir une zone en dehors de laquelle il est inutile de rechercher. Nous appelons la zone de recherche du chemin de recalage optimal corridor. En désignant par EMAX et EMIN respectivement la pente du chemin local de pente maximale et de pente minimale de la contrainte utilisée par exemple dans le cas de la contrainte Type II EMAX=2 et EMIM=1/2. La figure 2.7 visualise la zone dans le plan de comparaison en dehors de laquelle il est inutile de rechercher le chemin optimal dans le cas de la contrainte Type III.

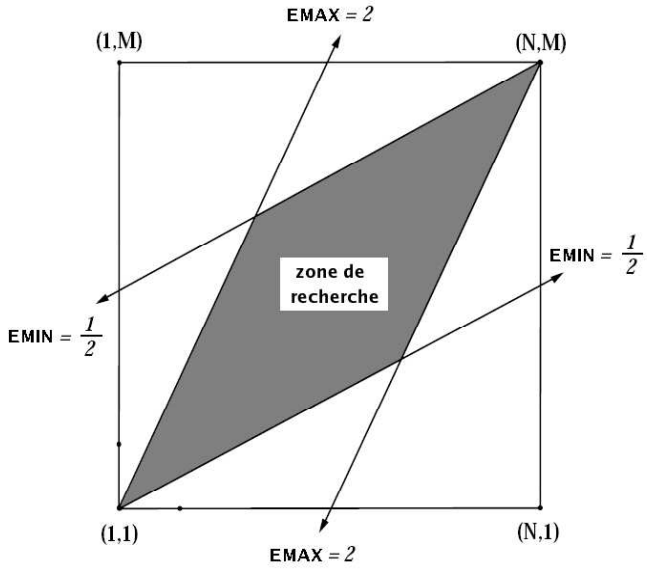


FIG. 2.5.: Corridor Type II

2.8. Path pruning

Le corridor a été mis en place pour diminuer le nombre de points calculés excluant tout les points des matrices adm et ldm ayant peut de chance d'être sur le chemin otimal. Il tient compte d'une part des points exclus par le type de voisinage (e.g. Les points dont le chemin pour les relier à (1,1) doit être supérieur à 3 dans le cas du voisinage 5 par exemple), d'autre part il écarte les points dont la valeur de distance augmentée (adm) de

leur précesseur est beaucoup plus importante que celle des points alentour et donc peu susceptible de faire partie du chemin optimal. On définit le seuil $\theta_P(m)$ comme suivant :

$$\theta_P(m) = 1.08 \min_n(adm(m-1,n)) \tag{2.27}$$

Cette tolérance permet d'accepter des trames en théorie trop éloignées de la partition.Cela donne une plus grande souplesse vis à vis d'un écart local assez grand entre la partition et l'enregistrement audio (dû à une qualité médiocre du fichier midi par exemple). En pratique la tolérance est de 8 %.

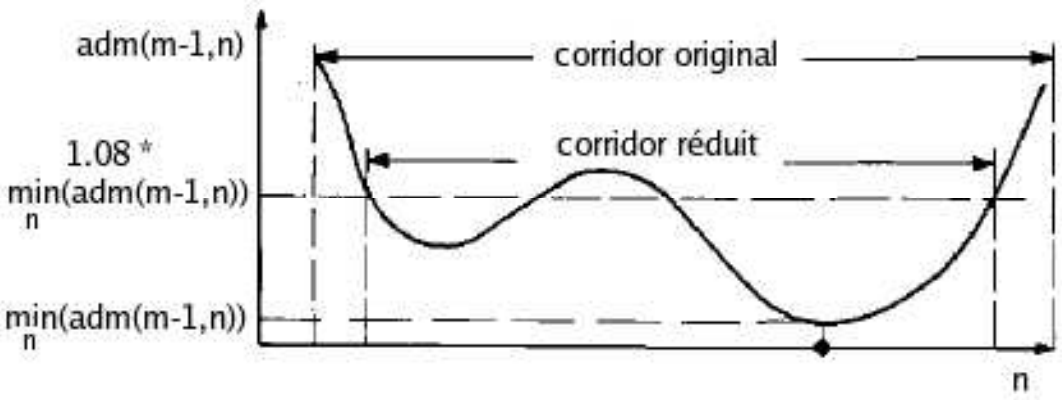


FIG. 2.6.: Path pruning

2.9. Shortcut path

Le stoquage du chemin est la partie la plus compliquée qui comprend de nombreuses variables. En fait il est impossible de stocker les coordonnées du prédecesseur de chaque point de la matrice dès que le nombre des trame deviens important. Comme seul le début de chaque note nous intéresse, seules les trois trames précédant, ceux ci sont stockeés.

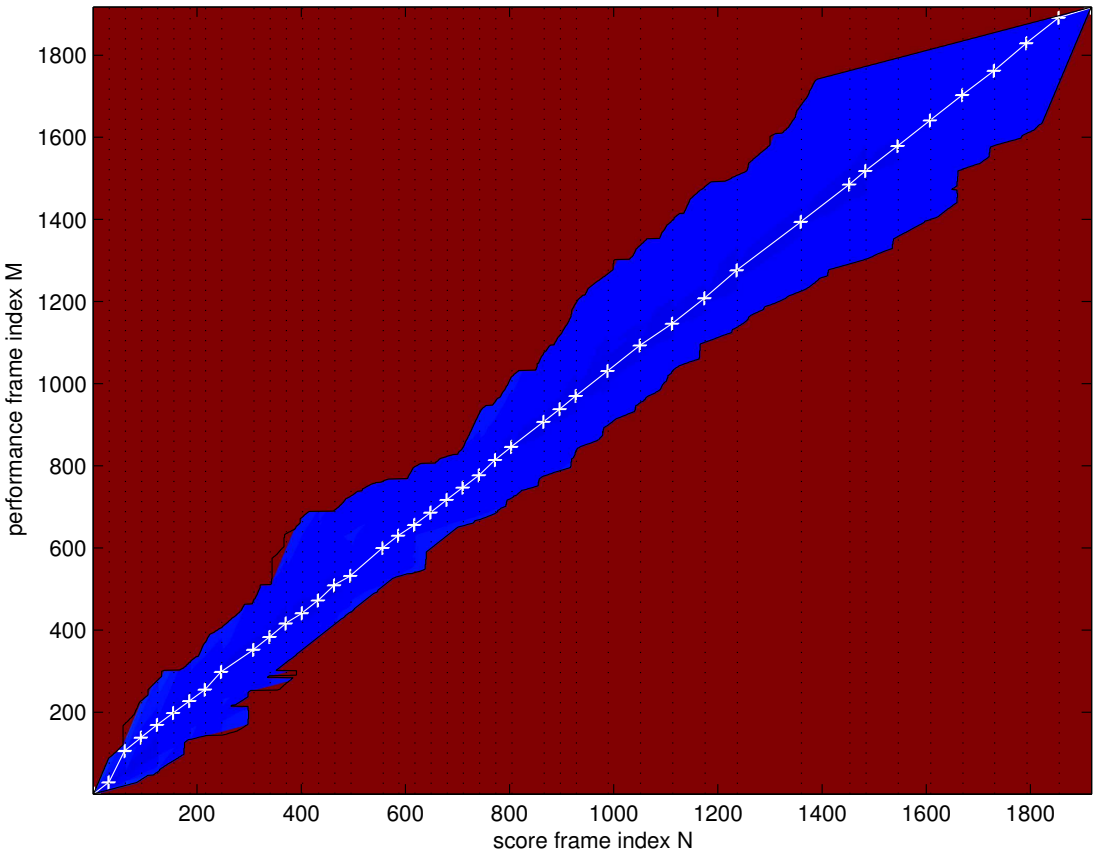


FIG. 2.7.: L'alignement musical utilisant l'algorithme DTW

3. Passage d'un algorithme DTW global à un DTW à court terme

3.1. Les algorithmes à optimums locaux :

[Di Martino, 1984]

L'algorithme de programmation dynamique général que nous avons expliqué au chapitre 2 se propose de déterminer parmi l'ensemble de tous les chemins de recalage définis par la contrainte locale utilisée, le chemin optimal. Pour ce faire un grand nombre de distances locales d(n,m) doivent être évaluées dans le plan de comparaison et un grand nombre de prédécesseurs doivent être garder. Avec les algorithme à optimum locaux le principe d'optimalité global du problème est abandonné afin d'essayer de minimiser considérablement le nombre de distances locales à calculer et minimiser les coûts de mémoire. La fonctionnelle qui est associée aux algorithmes à optimums locaux est la somme des minimums locaux dans des pavés du plan de comparaison prédéfinis ou bien dont la position est évolutive, normalisée par le nombre de pavés considérés. La figure 3.3 montre les differents types de pavés qui sont utilisés habituellement. Si l'on désigne par (n_{k_i}, m_{k_i}) un point de k^{ieme} pavé et par NP le nombre de pavés considérés de la comparaison entre deux formes $\mathcal{X}$ et $\mathcal{Y}$ il vient donc que les algorithmes à optimums locaux évaluent le taux de dissemblance $D(\mathcal{X}, \mathcal{Y})$ de la manière suivante :

$$D(\mathcal{X}, \mathcal{Y}) \quad = \quad \frac{1}{NP} \sum_{k=1}^{NP} \min_i d(n_{k_i}, m_{k_i}) \tag{3.1}$$

D'après equation 3.1 il est clair que plus la taille du pavé est faible et plus le nombre de distances locales à évaluer est dimuné. On voit donc qu'il faut dans un algorithme à optimums locaux effectuer un compromis sur ce paramètre afin de limiter au maximum le temps de calcul sans trop dégrader les performances de l'algorithme. Les algorithme à optimum locaux peuvent gagner par rapport à un algorithme de programmation dynamique classique jusqu'à un rapport dix en temps de calcul tout en conservant des taux de reconnaissance honorables. Les algorithmes à optimums locaux ont été développés pour la reconnaissance de parole pour aligner des segments de parole de locuteurs différents. Toutefois ces algorithmes sont en général peu robustes en ce sens qu'un locuteur peu entraîné aux systèmes de reconaissance vocaux, mettra souvent en défaut ces derniers.

En effet, ce type de locuteur aura tendance à engendrer des distortions temporelles qui nécessitent des chemins de recalage s'écartent sensiblement de la diagonale du plan de comparaison et donc ne pouvant être compensées par un algorithme à optimums locaux dont la stratégie est d'évaluer le score d'un chemin sous-optimal appartenant à un voisinage généralement proche de la diagonale.

3.2. Le nouvel algorithme d'alignement court terme

Comme nous avons vu dans les chapitres 2.3.2 et 2.7, la meilleure distance $adm(M,N)$ (ou $D(N,M)$) entre les deux séquences est obtenue par la détermination d'un chemin dans la matrice $adm(m,n)$ de façon à minimiser la somme des distances locales rencontrées pour aller d'un point initial (généralement $(1,1)$, correspondant au début des deux séquences) à un point final (généralement (N,M), correspondant à la fin des deux séquences).

En général il est théoriquement impossible de déterminer ce chemin sans calculer toute la matrice $adm(m,n)$ avec $n = 1,2,\ldots,N$ et $m = 1,2,\ldots,M$, parce ce que l'optimum local nommé $(\widehat{n_l},\widehat{m_l})$ ne tient pas compte des points (n,m) avec $\widehat{n_l} < n \leq N$ et $\widehat{m_l} < m \leq M$. Comme nous avons vu précédemment, la stratégie est donc d'évaluer le score d'un chemin sous-optimal appartenant à un voisinage généralement proche de la diagonale. Cette stratégie n'est pas bien adaptée à l'alignement musical, parce que la propriété d'un chemin appartenant à un voisinage généralement proche de la diagonale n'est pas nécessairement garantie.

La méthode proposée est basée sur le principe d'optimalité à court terme (chapitre 2.5). En utilisant les contraintes des limites (equations 2.8 et 2.9) on peut formuler le principe d'optimalité comme suivant :

> *Soit $C_{(1,1)}^{(N,M)}$ le chemin optimal global joignant le points $(1,1)$ et (N,M), alors pour tout point $(n,m) \in C_{(1,1)}^{(N,M)}$, le chemin de recalage $C_{(1,1)}^{(n,m)}$ est optimal.*

On note le chemin optimal global et les points associés au chemin optimal global comme suit :

$$C_{1,1}^{N,M} \triangleq \widehat{C} \tag{3.2}$$

$$(n,m) \in C_{1,1}^{N,M} \triangleq (\widehat{n},\widehat{m}) \tag{3.3}$$

Le problème est donc de trouver le point $(\widehat{n},\widehat{m})$ pour obtenir la partie $C_{(1,1)}^{(\widehat{n},\widehat{m})}$ d'alignement optimal global. Comme nous avons vu précédemment il est théoriquement impossible de déterminer le point $(\widehat{n},\widehat{m})$ à partir des optimums locaux $(\widehat{n_l},\widehat{m_l})$ parce qu'on

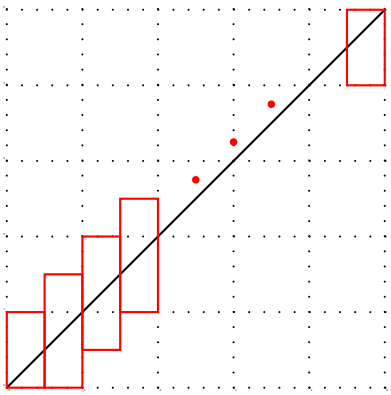


FIG. 3.1.: Pavés unicolonnes centrés sur la diagonale.

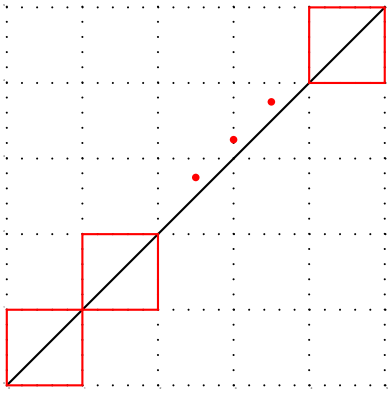


FIG. 3.2.: Pavés matriciels centrés sur la diagonale.

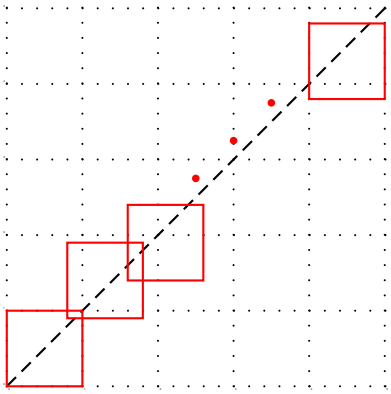


FIG. 3.3.: Pavés matriciels évolutifs : le sommet inférieure gauche du pavé i est positionné sur le minimum trouvé au pavé i-1. D'après HATON (Haton [1974])

ne peut que resoudre ce problème en calculant toute la matrice des distances cumulées pour faire le backtrack à partir du point (N, M).

La contrainte des limites (equation 2.8) assure que tous les chemins du plan de comparaison vont se rejoindre au plus tard au point $(1, 1)$. Supposant que dans le cas d'alignement musical il existe un $(\widehat{n}, \widehat{m})$ qui ne dépend pas de tous les points qui font partie du chemin $C^{(N,M)}_{(\widehat{n}+1,\widehat{m}+1)}$ il est possible de déterminer le $(\widehat{n}, \widehat{m})$ sans calculer toute la matrice $adm(M, N)$.

3.2.1. L'alignement local optimal : Type I

Dans cette partie nous montrons la développement d'un algorithme qui détermine le point $(\widehat{n}, \widehat{m})$, optimal au sens global. Pour cette première dérivation nous avons choisi la contrainte locale de Type I parce qu'il assure que tous les chemins $C^{(n_c,m'\leq m)}_{(1,1)}$ sont atteints à partir des point (n_c, m) (avec $rmin \leq n_c \leq rmax$, où $rmin$ est le bord du corridor à gauche et $rmax$ est le bord du corridor à droite pour le m fixé (figure 3.4)). Les Types II et V permettent des pas $(n-1, m-2)$ c'est à dire qu'il n'est pas sûr que tous les chemins $C^{(n_c,m'\leq m)}_{(1,1)}$ sont atteint à partir des point (n_c, m). Cette propriété de Type I est importante pour la suite de la premère partie.

Supposant qu'on utilise la contrainte locale de Type I et que l'on fait tous les backtracks à l'intérieur du corridor pour un m (index de l'audio) fixé, c'est à dire pour les point (n_c, m), on obtient tous les chemins possibles jusqu'au point m. Pour trouver la partie $C^{(\widehat{n},\widehat{m})}_{(1,1)}$ du chemin optimal $\widehat{C}$ on doit donc chercher le n_c correspondant au $\widehat{n}$.

A cause de la contrainte des limites tous les chemins possibles commencant au point $(1, 1)$ où respectivement des backtracks les chemins finissent au point $(1, 1)$. Il est clair que les chemins qui suivent la contrainte de continuité de Type I ne peuvent pas se croiser, parce que chaque predésseur du chaque point est uniquement défini et le Type I ne permet pas de sauter un point. C'est à dire que dès que deux chemins se rejoignent ils vont suivre le même chemin. En conséquence chaque chemin possible va rejoindre au plus tard le chemin $\widehat{C}$ au point $(1, 1)$. Il y a un dernier chemin qui va rejoindre le chemin $\widehat{C}$ au point de fusion et noté point G. Dès que on a trouvé le point G il ne reste que le chemin optimal. Ce point G correspond donc au point $(\widehat{n}, \widehat{m})$. Supposant que le point G est avant $(1, 1)$ on sait que la dernière partie du chemin $C^{(\widehat{n},\widehat{m})}_{(1,1)}$ est optimale.

A cause du noncroisement, il suffit donc de faire deux backtracks à partir des points $(rmin, m)$ et $(rmax, rmax)$ pour déterminer le point $(\widehat{n}, \widehat{m})$.

Nous redéfinissons le principe d'optimalité de la façon suivante :

$$C_{n',m'}^{rmin,m} \quad et \quad C_{n',m'}^{rmax,m} \tag{3.4}$$

$$\Rightarrow (n', m') = (\widehat{n}, \widehat{m}) \tag{3.5}$$

$$\Rightarrow C_{1,1}^{n',m'} \in \widehat{C} \tag{3.6}$$

$$\Rightarrow C_{1,1}^{n',m'} = C_{1,1}^{(\widehat{n},\widehat{m})} \tag{3.7}$$

3.2.2. L'alignement à court terme : Type V

Le non croisement des chemins est la propriété la plus importante de Type I. Si on peut montrer que cette propriété est aussi assurée pour le Type V, les hypothèses de la partie précédante sont aussi assurées. C'est à dire le STDTW peut se appliquer de la même façon au Type V.

La figure 3.4 montre touts les backtracks en intérieur du corridor pour un m (index de l'audio) fixe.

Si les chemins ne peuvent pas se croiser il suffit donc de ne faire que 2 backtracks (Figure 3.5) :

1. Backtrack rmin
2. Backtrack rmax

En respectant le principe d'optimalité les chemins ne peuvent pas se croiser. Dans l'annexe se trouve la preuve détaillée de cette propriété.

3.3. Principe

Les étapes de l'alignement de partition utilisant STDTW (figure 3.6)

1. Avancement dans l'audio de 500 trames (calcul des matrices ldm et adm et détermination des prédécesseurs optimal)
   a) backtrack (m,rim)
   b) backtrack (m,rmax)
   c) détermination de point de fusion
2. les variables importants sont :
   a) les prédécesseurs en intérieur des backtracks jusqu'au point de fusion
   b) la partie du chemin optimal determinée
   c) les corresponances associées à la dernière note qui sont aussi inclues dans le chemin optimal

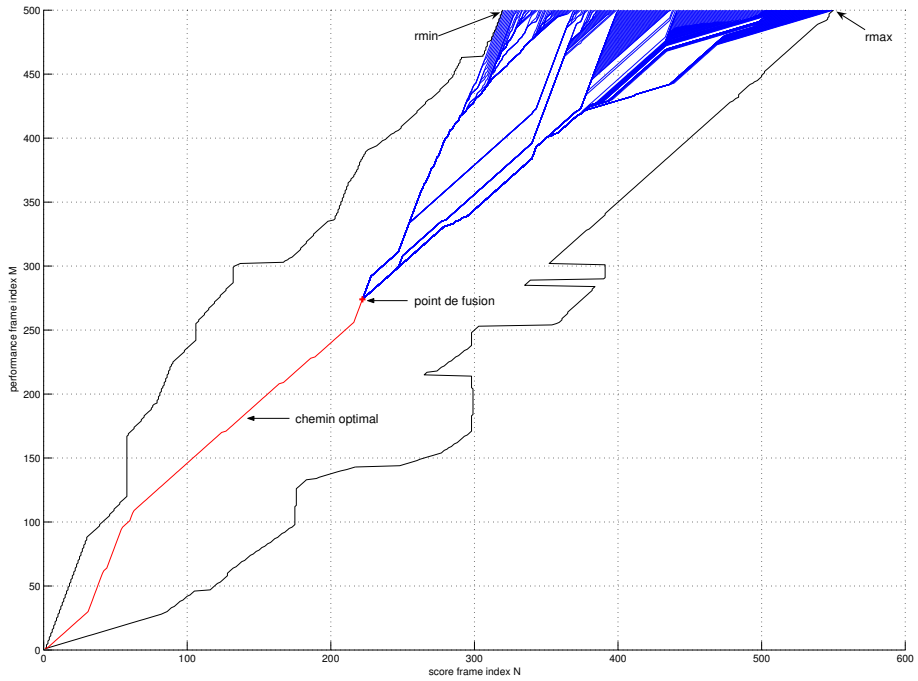


FIG. 3.4.: Les backtracks en intèrieure du corridor

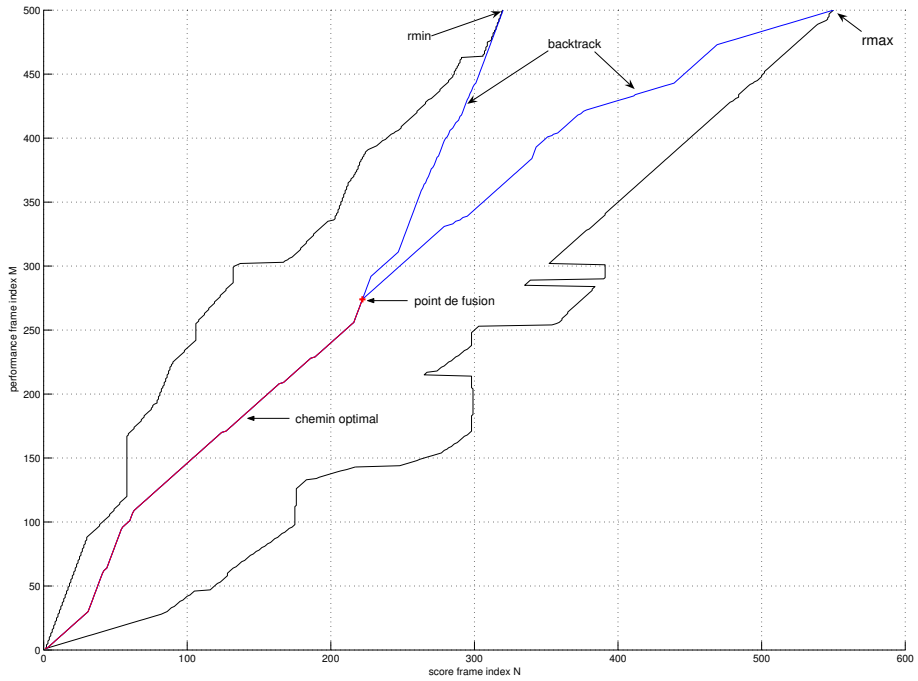


FIG. 3.5.: Les deux backtracks au bord du corridor

Si la taille de la matrice de prédécesseurs est trop importante on a les possibilités suivantes :

1. calcul de la matrice des prédécesseurs associée au short cut path
2. recommencer l'algorithme à la dernière note trouvée
3. diminuer la tolerance (2.8)

La figure (3.3) montre le resultat de l'alignement d'un morceau de jazz de 4.30 min.

3.4. Les programmes réalisés en Matlab

3.4.1. Programmes implémentés

Le programme d'alignement fait appel à differents fonctions codées en Matlab. Nous avons implémenté le nouvel algorithme STDTW sur Matlab et l'intégré dans le code existant. Les fonctions concernantes la partie DTW sont donc remplacés par les fonctions associés à l'algorithme STDTW.

Les fonctions consistent à les tâches suivantes :

1. admcalc5.m ⇒ coeur du programmme d'alignement STDTW
2. Short Time DTW Backpath.m ⇒ recherche du point de fusion (consiste à calculer des backtracks et de déterminer le point de fusion)
3. Short Time pred mat.m ⇒ gestion de la matrice des prédécesseurs

3.4.2. Réduction de l'occupation en mémoire vive

L'occupation de mémoire vive limitent le son à 3 minutes environ. Il est donc essentiel de réduire cette taille, même avec l'algorithme à court terme.

Environ 4 semaines ont été néccessaires à ce travial difficile (tableaux et cell-array Matlab incompatibles) pour réduire l'occupation en mémoire vive de la matrice de prédécesseurs et développer l'algorithme de backtrack et de point de fusion.

La programmation en Matlab a nécessité le passage en cell-array (ce serait beaucoup plus simple en C/C++ mais le logiciel est en Matlab).

Il faut donc faire toute une gestion des indices pour réintialiser l'algorithme une à chaque nouveau point de fusion G. Par contre cela permet d'arreter l'algorithme en un point G et de le reprendre ultèrieurement à ce point G à condition de commencer au dernier début théorique de note dans la partition avant ce point G. Cette possibilité est très interessante pour les fichiers longs, dont le temps de calcul d'alignement peut être très grand : on peut donc le faire progressivement par morceaux d'un point G à un autre.

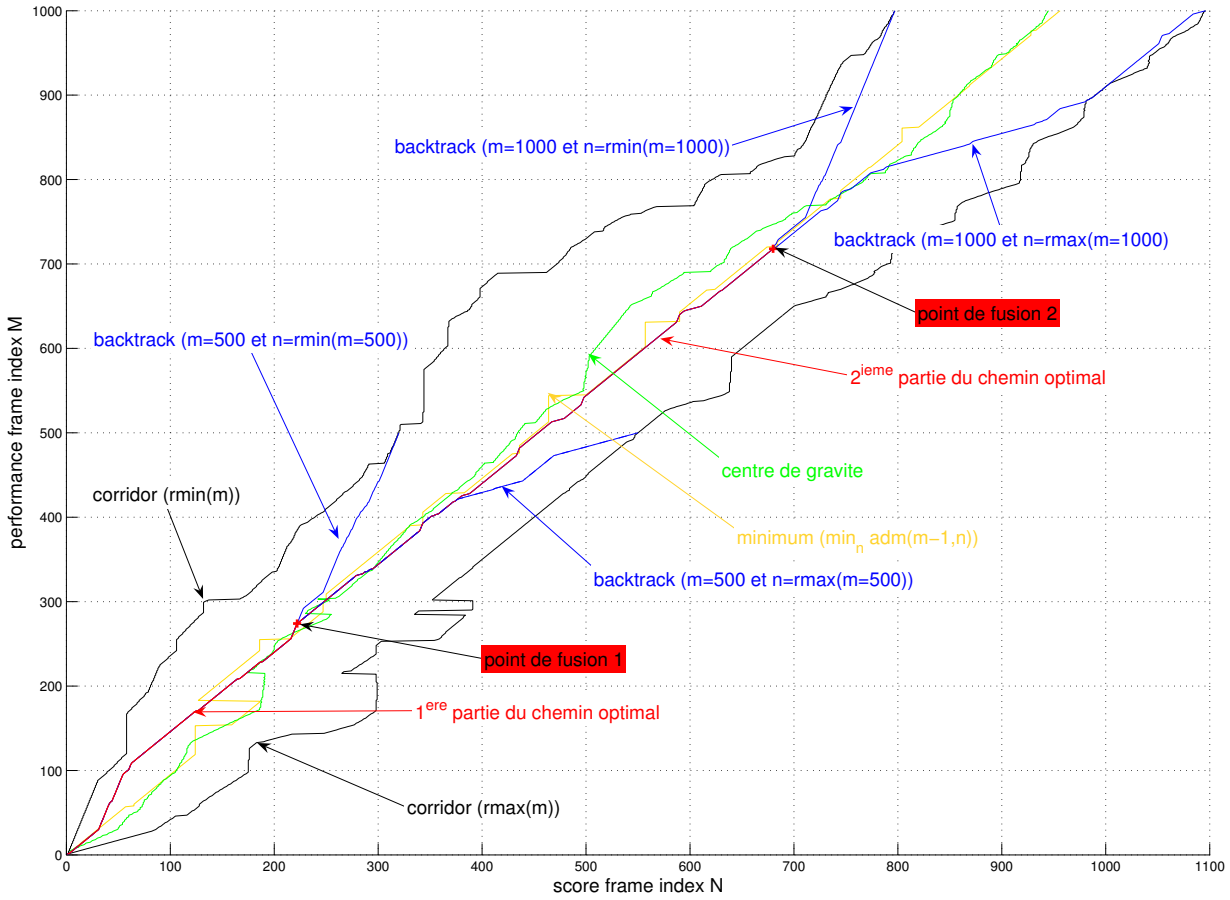


FIG. 3.6.: Principe : STDTW

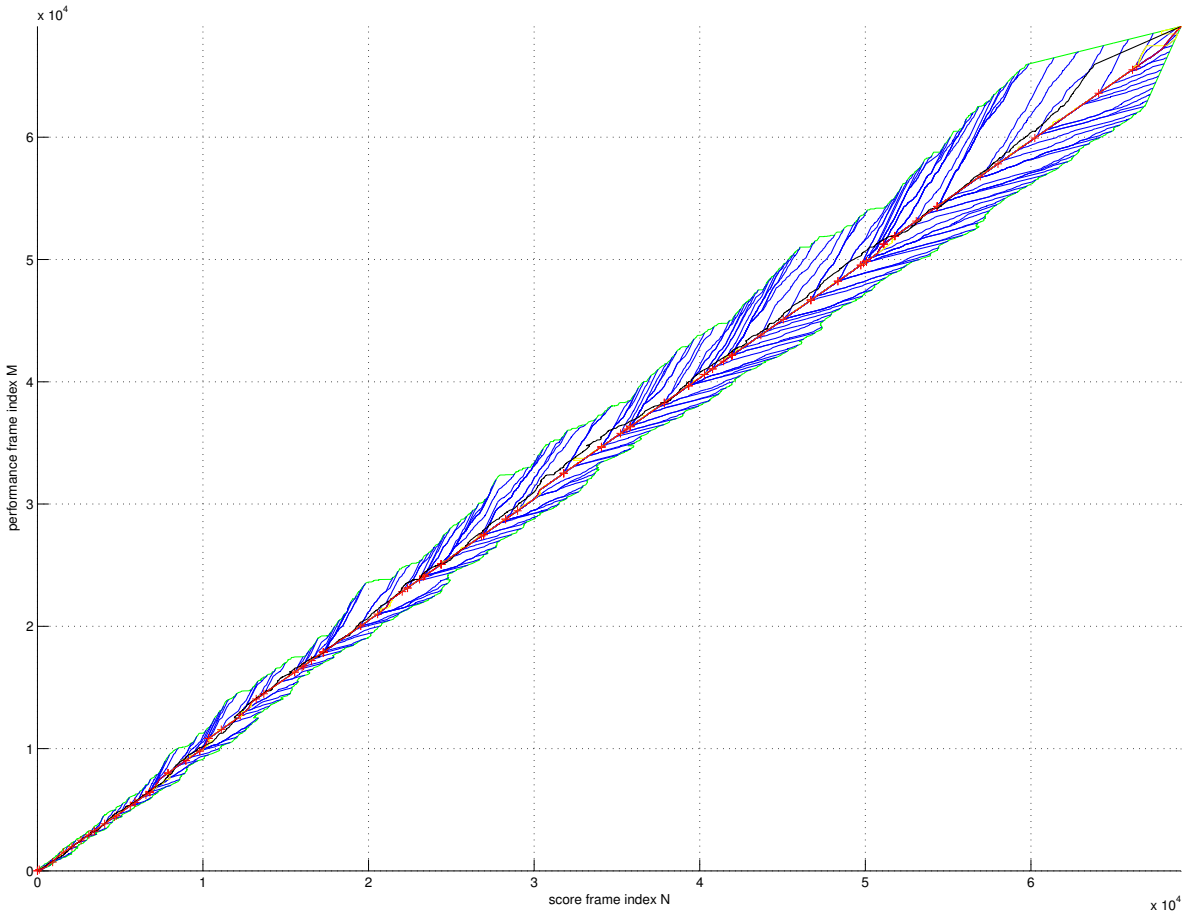


FIG. 3.7.: Alignement musical utilisant le nouvel algorithme STDTW

3.5. Performance de l'algorithme STDTW

Nous avons inventé un nouvel algorithme de DTW, nommé STDTW (Short Time Dynamic Time Warping). Il permet donc en théorie d'aligner des morceaux aussi long qu'on veut. Mais mieux encore, il est aussi optimal que l'algorithme global. Enfin, s'il était utilisé dans un contexte "temps réel" il fournirait le resultat de l'alignement avec un delai aussi faible que possible, c'est à dire dés que l'on a "entendu" suffisemment de l'audio pour être certain de l'alignement.

Nous avons fait des test sur une base de données des fichiers midi et fichier audios associés. Nous sommes arrivé d'aligner des morceaux de Classique de 8 minutes et des morceaux de Jazz de 7.5 min.

L'algorithme de la détéction des percussions doit être encore adapter à l'algorithme STDTW. Cette partie est prévu pour la suite de stage. Le calcul du shortcutpath n'est pas encore optimal parce qu'il nécessite encore beaucoup de temps de calcul. La détermination de la matrice de prédécesseur de shortcutpath est seulment fait en cas d'une taille importante de la matrice de prédécesseur entière (environ 500 MB).

4. Conclusion

Ce stage aura montré ma capacité à travailler dans l'Ircam sur un sujet scientifique complexe. Le stage a fait l'objet d'une présentation en salle de conférence lors d'un séminaire de recherche et création ayant pour objet l'alignement de partitions. Les travaux effectués tout au long de mon stage auront été réellement passionnants, et préparent de la meilleure façon possible l'entrée dans le monde professionnel. Ils ont nécessité un investissement personnel important, et une organisation rigoureuse, qualités requises pour un ingénieur. Le sujet aura été riche, et l'intégration dans l'équipe analyse synthèse excellente. Les résultats obtenus sont encourageants, et le travail est utilisable directement et riche en documentation. Finalement, ce stage ne s'arrête pas une fois ce rapport rendu : si la phase principale de développement voit globalement sa fin, le reste du temps sera consacré au perfectionnement de certaines procédures ainsi qu'à une documentation efficace du code matlab produit pour qu'il soit facilement compréhensible, utilisable et adaptable par d'autres programmeurs qui me succéderont. Ce stage de fin d'étude aura été très enrichissant et formateur. C'est la dernière étape avant l'entrée à proprement parler dans la vie d'ingénieur.

4.1. Résultats

1. Invention de l'algorithme SDTW (Short Time Dynamic Time Warping)
2. Implémentation de l'algorithme et intégration dans le projet existant en Matlab
3. Test de l'algorithme à partir d'un base de donnée des partitions (fichiers midis) et des performances (fichiers audios) de l'Ircam
4. Cooperation avec autres projets de l'Ircam.
    a) Estimation f0 (donner un résultat fiable d'un alignement musical pour un algorithme de transcription automatique d'un fichier audio à un fichier midi (sans partition) pour faire des comparaisons)
    b) La compositrice Patricia Alessandrini (Résynthèse à partir d'un fichier midi aligné)

4.2. Perspectives

1. Structure musicales :

   Le point de fusion peut probablement donner des informations sur la structure musicale du morceau

2. Détection des percussion :

   Adaption l' algorithme de la détéction des percussions à l'algorithme SDTW

3. Shortcutpath

   Amélioration de l'algorithme

4. Synthèse concatenative :

   Création d'une base de données qui contient des fichiers audio segmentés et indexés à l'aide de l'algorithme SDTW

5. Séminaire Ircam

   Prévu pour le 30. juin à l'Ircam Centre Pompidou

6. Prévu à long terme : Brevet

7. Prévu à long terme : Article

8. Presentation de l'algorithme de l'alignement à un colloque informatique musical (21.9.2005 à Bonn, congrès annuel de la societé d'informatique allemande)

A. Présentation de l'Ircam

Georges Pompidou initie en 1969 la création de l'Institut de Recherche et Coordination Acoustique/Musique dont il confie la direction au compositeur et chef d'orchestre Pierre Boulez. L'Ircam devient alors, et demeure aujourd'hui encore, un centre unique au monde, dédié à la recherche et la création musicale contemporaine.

L'Ircam est associé au Centre Pompidou et placé sous la tutelle du ministère de la Culture et de la communication. Depuis 1995, l'Ircam et le CNRS sont partenaires dans le cadre d'une unité mixte de recherche SMTS (Sciences et technologies de la musique et du son - UMR 9912).

Au départ, projet d'un homme, Pierre Boulez, compositeur, chef d'orchestre et théoricien de la musique, l'Ircam relaiera les utopies et conceptions esthétiques de son fondateur, faire se rencontrer art et science pour élargir l'instrumentarium et renouveler le langage musical.

A la fin des années 1970, l'Ircam propose la réflexion la plus avancée sur l'informatique musicale dans le monde.

Avec l'arrivée de Laurent Bayle en 1992, la structure s'ouvre artistiquement à d'autres formes, et travaille à la recherche de nouveaux publics, au travers notamment d'un rendez-vous, le festival Agora. D'autre part, afin d'accompagner le développement de l'informatique personnelle et des réseaux, la création du Forum permet une diffusion du savoir-faire Ircam dans le monde entier.

Depuis 2002, Bernard Stiegler, philosophe, en a pris la direction et réaffirme la vocation première de l'Ircam : la coordination entre recherche et création.

S'attachant à renouer les relations entre art et science, le projet de l'Ircam s'inscrit de plain-pied dans les problématiques contemporaines telles que les rapports entre industries culturelles et création.

S'appuyant fortement sur la présence des compositeurs et artistes invités à dialoguer avec ses équipes scientifiques, l'Ircam contribue au débat posé par les enjeux actuels, qu'ils soient théoriques, musicaux, esthétiques ou politiques.

L'Ircam, qui est le plus gros centre de recherche scientifique au monde entièrement dédié aux technologies pour la création musicale, accueille une importante population de chercheurs, près de 90 scientifiques.

Ces équipes, qui demeurent en relation étroite et permanente avec le monde des compositeurs en participant directement aux projets de création menés chaque année (375 oeuvres produites depuis 1977), ménent des recherches fondamentales sur les apports des mathématiques, de l'acoustique, de l'informatique et de la physique, appliqués à la

création musicale.

Leurs travaux autour de l'acoustique instrumentale, de l'analyse et de la synthèse des sons, des représentations numériques des structures musicales, du temps réel, de l'acoustique des salles, du design sonore, de la musicologie, des métadonnées musicales et du génie logiciel appliqué au traitement du son suscitent des échanges suivis avec la communauté scientifique internationale.

Parallélement à ces nouveaux éléments d'écriture et de lutherie mis à disposition des créateurs à travers ces recherches, et en étroite synergie avec eux, les terrains d'application sont nombreux.

Le financement aussi bien que la socialisation de ces travaux s'opérent notamment par la participation à des projets européens ou des partenariats développés avec le secteur industriel. Cette technologie peut concerner l'évolution des appareils de diffusion en haute fidélité, le son multi-canal en général, les normes de télécommunication, le design d'ambiance sonore des lieux publics, les logiciels d'éducation musicale, le traitement de la voix au cinéma, les nouveaus supports éditoriaux numériques, etc...

B. Preuve : non croisement dans le cas de type V

Soit un chemin optimal en backtrack du point (M, N) qui passe par (m, n). Nous voulons prouver qu'il ne peut pas croiser un autre chemin optimal dans l'étape vers les prédécesseurs directs de (m, n) (figure B.1).

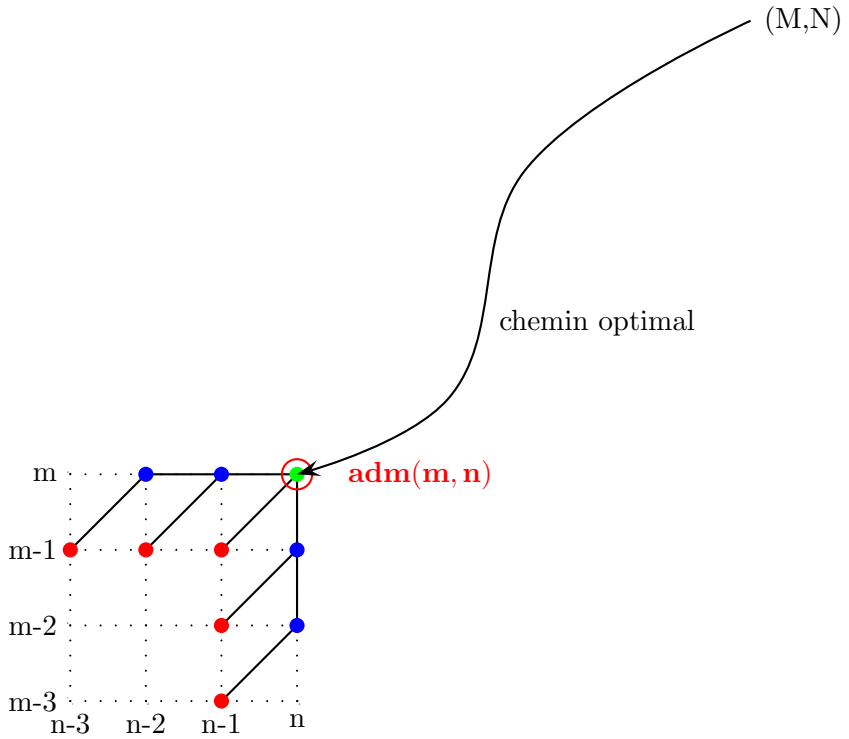


FIG. B.1.: Type V

La valeur $adm(m,n)$ est donnée par :

$$\mathbf{adm(m,n)} = \min \left\{ \begin{array}{l} adm(m-1,n-1) + w_d ldm(m,n) \\ adm(m-2,n-1) + w_v ldm(m,n) + w_d ldm(m-1,n) \\ adm(m-1,n-2) + w_h ldm(m,n) + w_d ldm(m,n-1) \\ adm(m-3,n-1) + w_v ldm(m,n) + w_d ldm(m-2,n) + w_v ldm(m-1,n) \\ adm(m-1,n-3) + w_h ldm(m,n) + w_d ldm(m,n-2) + w_h ldm(m,n-1) \end{array} \right\}$$

Il est clair que le croisement ne peut pas se produire en (m,n) ni en un de ses prédecesseurs (principe d'optimalité de Bellman). Les portions de chemin entre (m,n) et un prédecesseur direct peuvent être classé dans deux classes, classe 1 (figure B.2) et classe 2 (figure B.3). Les chemins de classe 1 ne peuvent pas être croisés qu'en l'un des points $(m,n-1)$ ou $(m,m-2)$. Ils ne peuvent donc être croisés que par un chemin de la classe 2 : le chemin de (i,j) à l'un de ses prédecesseurs dans la classe 2 $(j-1,i-1)$, $(j-1,i-2)$, $(j-1,i-3)$ croiserait le chemin de (m,n) à l'un des prédecesseurs dans la classe 1.

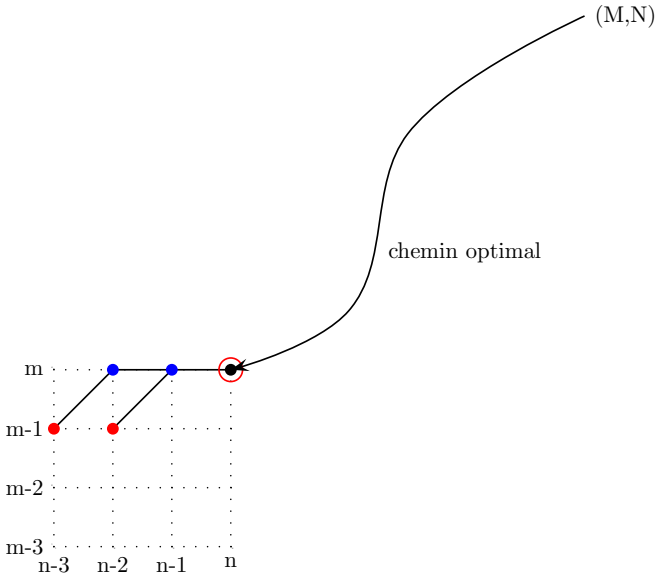


FIG. B.2.: classe 1

Il suffit donc d'examiner tous ces cas de croisement un à un. Les inégalités ci-après concernent les distances locales $p_{k,l}$ (matrice ldm) aux differents points (k,l) et les distances cumulées $\bar{p}_{k,l}$ (matrice adm) . La distance locale du point de fusion est notée C. Les inégalités se déduisent immédiatement des figures.

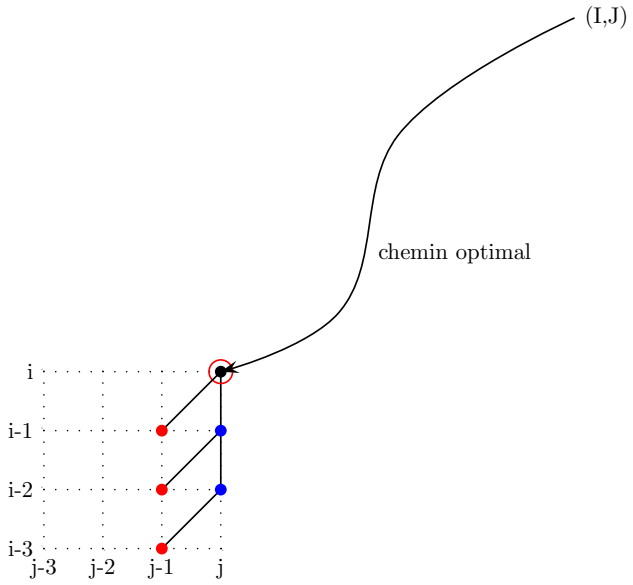


FIG. B.3.: classe 2

Chaque inégalité (sauf l'avant derniére croisement 10 figure B.13) se simplifie en C=0. Il n'y a aurait donc de croisement possible que si la distance locale au point de croisement était nulle, ce qui est exclu dans notre cas, car il est "presque impossible" (au sens probabiliste) que cette distance soit nulle. L'avant dernière (croisement 10 figure B.13) in égalité se simplifie en un inégalité $\bar{p}_{2,3} = p_{2,4} + \bar{p}_{1,3}$ tout aussi "presque impossible".

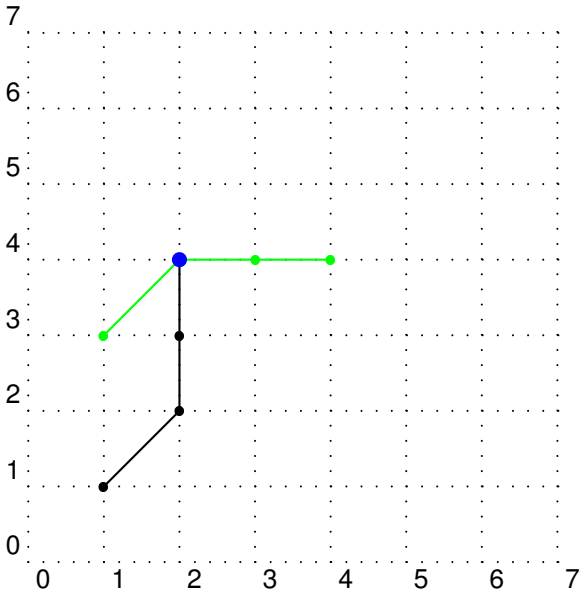


FIG. B.4.: Croisement 1

$$p_{4,4} + p_{3,4} + C + \bar{p}_{1,3} \leq p_{4,4} + p_{3,4} + \underbrace{p_{2,3} + p_{2,2} + \bar{p}_{1,1}}_{S} \tag{B.1}$$

$$C + \underbrace{p_{2,3} + p_{2,2} + \bar{p}_{1,1}}_{S} \leq C + \bar{p}_{1,3} \tag{B.2}$$

$$C + \bar{p}_{1,3} \leq S \leq \bar{p}_{1,3} \tag{B.3}$$

$$\Rightarrow C = 0 \tag{B.4}$$

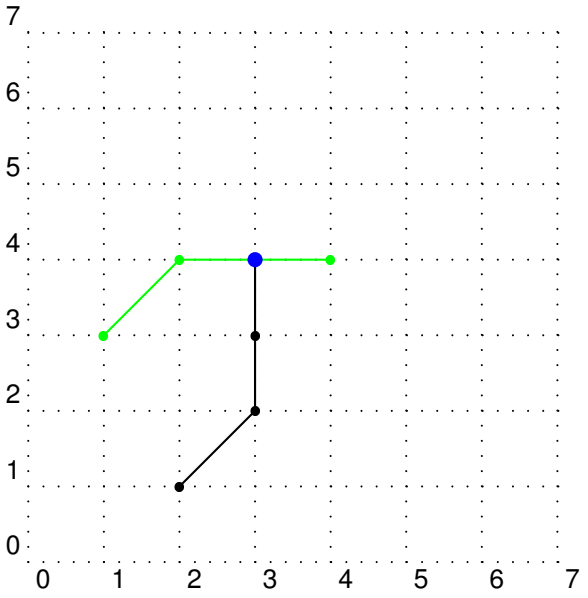


FIG. B.5.: Croisement 2

$$p_{4,4} + C + p_{2,4} + \bar{p}_{1,3} \leq p_{4,4} + \underbrace{p_{3,3} + p_{3,2} + \bar{p}_{2,1}}_{S} \tag{B.5}$$

$$C + \underbrace{p_{3,3} + p_{3,2} + \bar{p}_{2,1}}_{S} \leq C + p_{2,4} + \bar{p}_{1,3} \tag{B.6}$$

$$C + p_{2,4} + \bar{p}_{1,3} \leq S \leq p_{2,4} + \bar{p}_{1,3} \tag{B.7}$$

$$\Rightarrow C = 0 \tag{B.8}$$

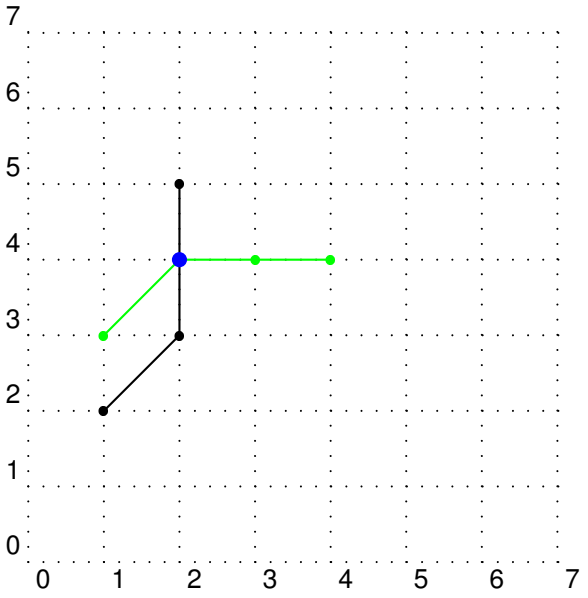


FIG. B.6.: Croisement 3

$$p_{4,4} + p_{3,4} + C + \bar{p}_{1,3} \leq p_{4,4} + p_{3,4} + \underbrace{p_{2,3} + \bar{p}_{1,2}}_{S} \tag{B.9}$$

$$p_{2,5} + C + \underbrace{p_{2,3} + \bar{p}_{1,2}}_{S} \leq p_{2,5} + C + \bar{p}_{1,3} \tag{B.10}$$

$$C + \bar{p}_{1,3} \leq S \leq \bar{p}_{1,3} \tag{B.11}$$

$$\Rightarrow C = 0 \tag{B.12}$$

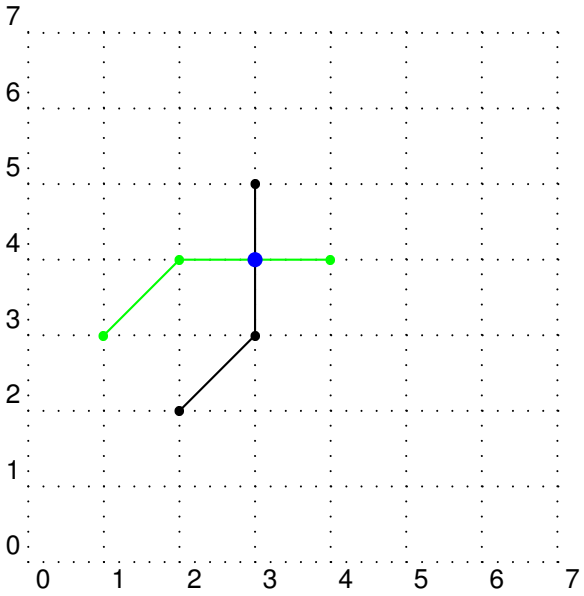


FIG. B.7.: Croisement 4

$$p_{4,4} + C + \underbrace{p_{2,4} + \bar{p}_{1,3}}_{S_1} \leq p_{4,4} + \underbrace{p_{3,3} + \bar{p}_{2,2}}_{S_2} \tag{B.13}$$

$$p_{3,5} + C + \underbrace{p_{3,3} + \bar{p}_{2,2}}_{S_2} \leq p_{3,5} + \underbrace{p_{2,4} + \bar{p}_{1,3}}_{S_1} \tag{B.14}$$

$$C + S_1 \leq S_2 \tag{B.15}$$

$$C + S_2 \leq S_1 \tag{B.16}$$

$$\Rightarrow C = 0 \tag{B.17}$$

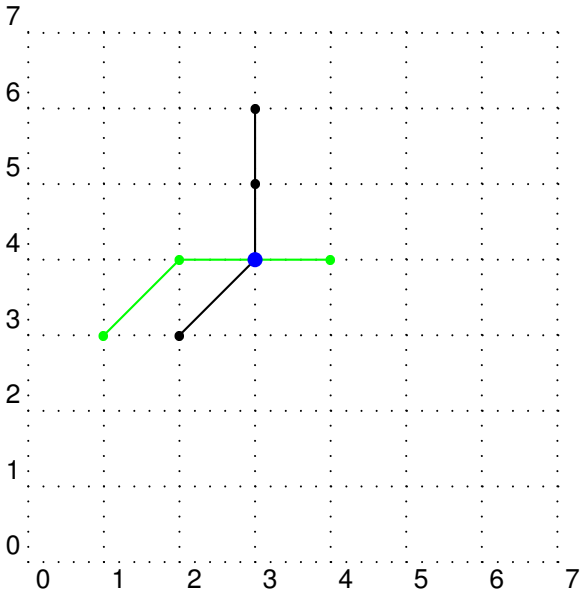


FIG. B.8.: Croisement 5

$$\tag{B.18}$$

$$p_{4,4} + C + \underbrace{p_{2,4} + \bar{p}_{1,3}}_{S} \leq p_{4,4} + C + \bar{p}_{2,3} \tag{B.19}$$

$$p_{3,6} + p_{3,5} + C + \bar{p}_{2,3} \leq p_{3,6} + p_{3,5} + \underbrace{p_{2,4} + \bar{p}_{1,3}}_{S} \tag{B.20}$$

$$C + \bar{p}_{2,3} \leq S \leq \bar{p}_{2,3} \tag{B.21}$$

$$\Rightarrow C = 0 \tag{B.22}$$

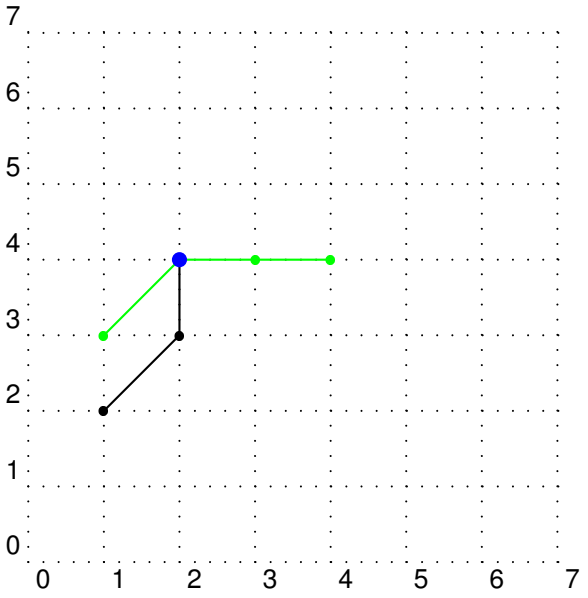


FIG. B.9.: Croisement 6

$$p_{4,4} + p_{3,4} + C + \bar{p}_{1,3} \leq p_{4,4} + p_{3,4} + \underbrace{p_{2,3} + \bar{p}_{1,2}}_{S} \tag{B.23}$$

$$C + \underbrace{p_{2,3} + \bar{p}_{1,2}}_{S} \leq C + \bar{p}_{1,3} \tag{B.24}$$

$$C + \bar{p}_{1,3} \leq S \leq \bar{p}_{1,3} \tag{B.25}$$

$$\Rightarrow C = 0 \tag{B.26}$$

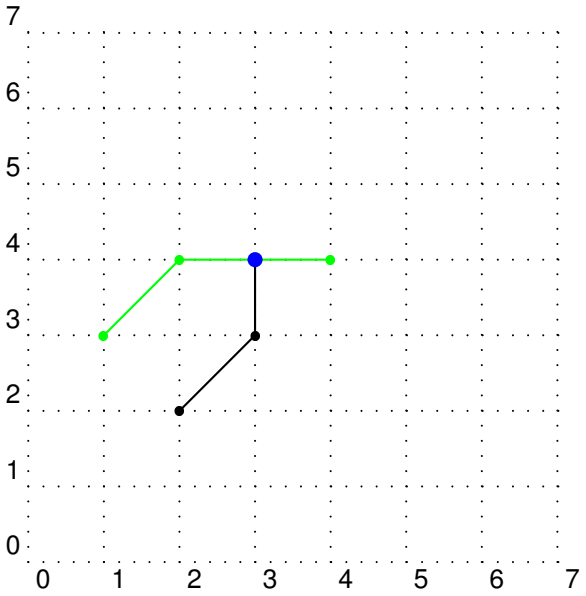


FIG. B.10.: Croisement 7

$$p_{4,4} + C + p_{2,4} + \bar{p}_{1,3} \leq p_{4,4} + \underbrace{p_{3,3} + \bar{p}_{2,2}}_{S} \tag{B.27}$$

$$C + \underbrace{p_{3,3} + \bar{p}_{2,2}}_{S} \leq C + p_{2,4} + \bar{p}_{1,3} \tag{B.28}$$

$$C + p_{2,4} + \bar{p}_{1,3} \leq S \leq p_{2,4} + \bar{p}_{1,3} \tag{B.29}$$

$$\Rightarrow C = 0 \tag{B.30}$$

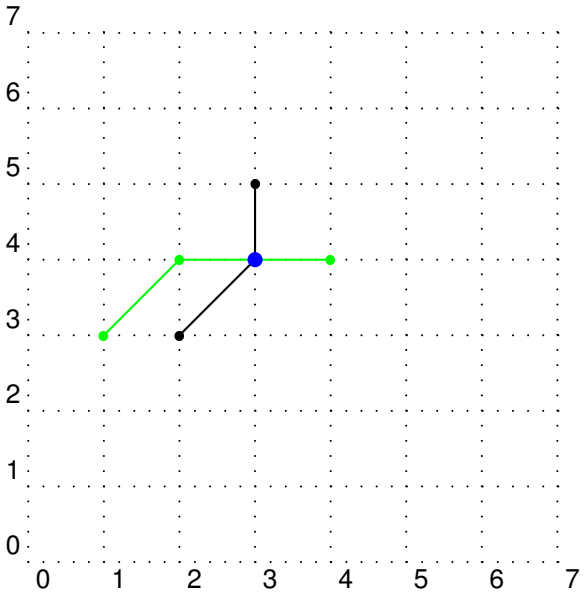


FIG. B.11.: Croisement 8

$$p_{4,4} + C + \underbrace{p_{2,4} + \bar{p}_{1,3}}_{S} \leq p_{4,4} + C + \bar{p}_{2,3} \tag{B.31}$$

$$p_{3,5} + C + \bar{p}_{2,3} \leq p_{3,5} + \underbrace{p_{2,4} + \bar{p}_{1,3}}_{S} \tag{B.32}$$

$$C + \bar{p}_{2,3} \leq S \leq \bar{p}_{2,3} \tag{B.33}$$

$$\Rightarrow C = 0 \tag{B.34}$$

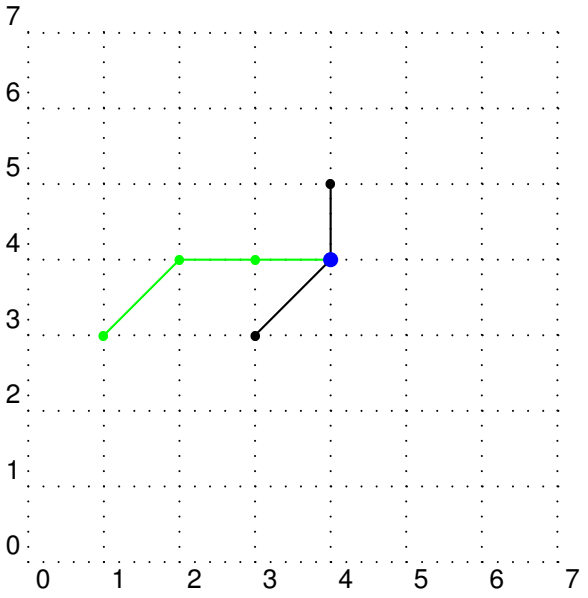


FIG. B.12.: Croisement 9

$$C + \underbrace{p_{3,4} + p_{2,4} + \bar{p}_{1,3}}_{S} \leq C + \bar{p}_{3,3} \tag{B.35}$$

$$p_{4,5} + C + \bar{p}_{3,3} \leq p_{4,5} + \underbrace{p_{3,4} + p_{2,4} + \bar{p}_{1,3}}_{S} \tag{B.36}$$

$$C + \bar{p}_{3,3} \leq S \leq \bar{p}_{3,3} \tag{B.37}$$

$$\Rightarrow C = 0 \tag{B.38}$$

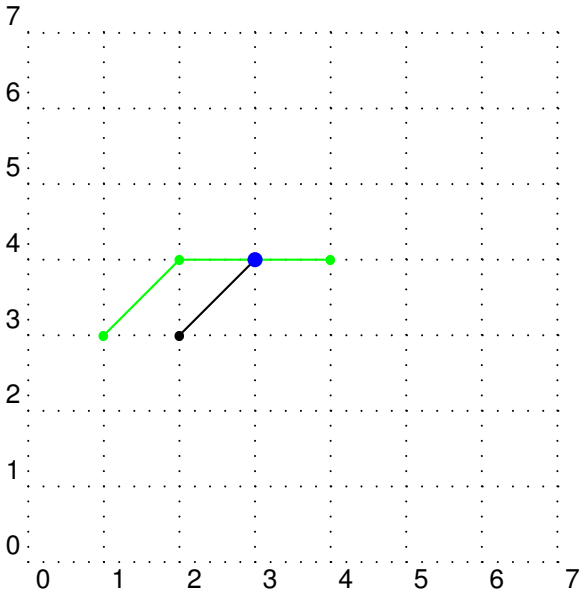


FIG. B.13.: Croisement 10

$$p_{4,4} + C + p_{2,4} + \bar{p}_{1,3} \leq p_{4,4} + C + \bar{p}_{2,3} \tag{B.39}$$

$$C + \bar{p}_{2,3} \leq C + p_{2,4} + \bar{p}_{1,3} \tag{B.40}$$

$$p_{2,4} + \bar{p}_{1,3} \leq \bar{p}_{2,3} \tag{B.41}$$

$$\bar{p}_{2,3} \leq p_{2,4} + \bar{p}_{1,3} \tag{B.42}$$

$$\Rightarrow \bar{p}_{2,3} = p_{2,4} + \bar{p}_{1,3} \tag{B.43}$$

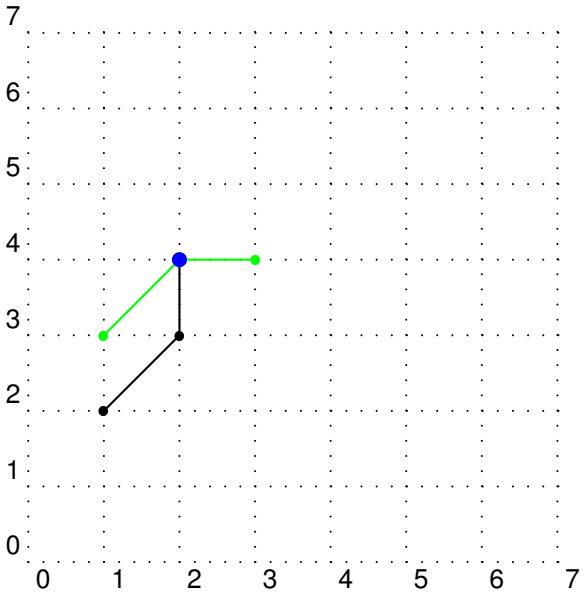


FIG. B.14.: Croisement 11

$$p_{3,4} + C + \bar{p}_{1,3} \leq p_{3,4} + \underbrace{p_{2,3} + \bar{p}_{1,2}}_{S} \tag{B.44}$$

$$C + \underbrace{p_{2,3} + \bar{p}_{1,2}}_{S} \leq C + \bar{p}_{1,3} \tag{B.45}$$

$$C + \bar{p}_{1,3} \leq S \leq \bar{p}_{1,3} \tag{B.46}$$

$$\Rightarrow C = 0 \tag{B.47}$$

Bibliographie

R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

L. R. Rabiner C. S. Myers and A. E. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. In *IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-28 , No. 6, pp. 623-635*, 1980.

J. Di Martino. *Contribution à la reconnaissance globale de la parole : mots isolés et mots enchaînés*. Thèse de docteur ingénieur, Univerité de Nancy I, Nancy, France, 1984.

J. P. Haton. *Contribution à la Analyse, la Paramétrisation et la Reconnaissance de la Parole*. Thèse d'etat, Univerité de Nancy I, Nancy, France, 1974.

E. Keogh. Exact indexing of dynamic time warping. In *Proc. 28th International Conference on Very Large Database (VLDB)*, Hong Kong, China, 2002. URL `http ://citeseer.ist.psu.edu/keogh02exact.html`.

Nicola Orio and Diemo Schwarz. Alignment of monophonic and polyphonic music to a score. In *International Computer Music Conference (ICMC)*, Havana, Cuba, Septembre 2001. URL `http ://mediatheque.ircam.fr/articles/textes/Orio01b`.

Lawrence R. Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, NJ, 1993. ISBN 0-13-015157-2.

Xavier Rodet, Joseph Escribe, and Sébastien Durigon. Improving score to audio alignment : Percussion alignment and precise onset estimation. In *ICMC*, 2004. URL `http ://mediatheque.ircam.fr/articles/textes/Rodet04a/`.

Diemo Schwarz. *Data-Driven Concatenative Sound Synthesis*. Thèse de doctorat, Université Paris 6 - Pierre et Marie Curie, Paris, France, 2004. URL `http ://mediatheque.ircam.fr/articles/textes/Schwarz04a`.

Ferréol Soulez, Xavier Rodet, and Diemo Schwarz. Improving polyphonic and poly-instrumental music to score alignment. In *International Symposium on Music Information Retrieval (ISMIR)*, Baltimore, USA, Octobre 2003. URL `http ://mediatheque.ircam.fr/articles/textes/Soulez03a`.