TABLES DES MATIERES

REMERCIEMENTS ..... **VII**

RESUME ..... **XI**

ABSTRACT ..... **XIII**

TABLES DES MATIERES ..... **XV**

0 CHAPITRE 0 : INTRODUCTION GENERALE **0**
0.1 Motivation **1**
0.2 Problématique et objectifs de la thèse **2**
0.2.1 Problématique 2
0.2.2 Objectifs 2
0.3 Organisation de la thèse **3**

1 CHAPITRE 1 : ETAT DE L'ART **6**
1.1 Introduction **7**
1.2 Partie vidéo : Mesures de similarité **7**
1.2.1 Qu'est ce qu'un document vidéo ? 8
1.2.2 Analyse vidéo : où sommes nous actuellement ? 8
1.2.2.1 La segmentation vidéo 8
1.2.2.1.1 La segmentation temporelle 8
1.2.2.1.2 Segmentation spatio-temporelle 9
1.2.2.2 Indexation vidéo 9
1.2.2.2.1 Indexation de bas niveau 10
1.2.2.2.2 Analyse sémantique 10
1.2.2.3 Situation actuelle 10
1.2.3 Qu'est ce qu'une mesure de similarité 11
1.2.3.1 Une mesure de similarité pour quoi faire ? 11
1.2.3.2 Quelques définitions 12
1.2.3.3 Les différentes approches de comparaisons et de mesures 12
1.2.4 Etat de l'art des mesures de similarité 13
1.2.4.1 Mesures s'inspirant de celles appliquées aux images 13
1.2.4.2 Mesures intégrant la dimension du temps 13
1.2.4.3 Les méthodes avec modèles à priori 14
1.2.4.4 Méthodes d'échantillonnage 14
1.2.4.5 Mesure de comparaison générique 15
1.2.4.6 Mesures permettant d'identifier les copies sans modèles à priori 15
1.2.4.6.1 Particularité 15

1.2.4.6.2 Les extractions de répétitions 16
1.2.4.6.3 Recherche dans les bases de données 17
1.2.5 Recherche sans considération de l'ordre temporel 17
1.2.5.1 Mesures permettant la classification en genre 18
1.2.6 Limites 18
1.3 Séries chronologiques et méthodes de comparaison 19
1.3.1 Définitions 19
1.3.2 Notation 19
1.3.3 Formulation du problème de la comparaison de séries chronologiques 20
1.3.4 Les solutions par utilisation de distances 20
1.3.4.1 Distance de Minkowski 20
1.3.4.2 Changement d'espaces de représentation 21
1.3.5 Programmation dynamique 22
1.3.5.1 Définition 22
1.3.5.2 La comparaison par déformation temporelle dynamique (DTW) 22
1.3.6 La distance d'édition 24
1.3.6.1.1 La distance d'édition pour les séquences de texte 24
1.3.6.1.2 La distance d'édition et les séquences numériques 25
1.3.6.2 La plus longue sous séquence commune (PLSC ou LCSS) 25
1.3.6.2.1 La PLSC pour les séries chronologiques 26
1.3.6.2.2 Transformations 26
1.3.6.3 Les méthodes d'approximations du calcul de la PLSC 28
1.4 Conclusion 28

2 CHAPITRE 2 : MATRICE DE COMPARAISON 30
2.1 Introduction 31
2.2 Stratégie de comparaison des caractéristiques audiovisuelles 32
2.3 Comparaison de deux séries temporelles 33
2.3.1 Notations et conventions 33
2.3.2 IQR : algorithme de l'intersection quadratique récursive 34
2.3.3 ESSV : algorithme d'extraction des séquences similaires de taille variable 36
2.3.4 CC : algorithme de calcul du taux de couverture 37
2.3.5 DiSC : algorithme de comparaison Dichotomique des Séries Chronologiques 39
2.3.6 Extension pour deux séries de tailles quelconques 39
2.3.7 Comparaison du CC avec PLSC 39
2.3.7.1 Algorithme PLSC 40
2.3.7.2 Comparaison théorique de la complexité 40
2.3.7.3 Défaut de l'algorithme CC 41
2.3.7.4 Comparaison expérimentale de complexité et de performance 42
2.3.8 Comparaison des séquences par morceaux 44
2.4 Calcul des enveloppes morphologiques de séries chronologiques 45
2.4.1 Définitions de morpho mathématiques 45
2.4.2 Deux opérateurs morphologiques : la dilatation et l'érosion 46
2.4.3 La construction de l'enveloppe morphologique 46
2.5 Algorithme de comparaison pour une caractéristique audiovisuelle 47
2.5.1 Typologie des courbes 47
2.5.2 Comparaison par intersection d'enveloppes 51
2.5.3 Adaptation de l'algorithme CC 52
2.5.4 Le structurant morpho mathématique 52
2.5.5 Entre l' «efficacité » et la « précision » 53
2.5.5.1 Le paramètre tMax 53
2.5.5.2 Le paramètre tMin 53

2.5.6 Comparaison du PLSC et du CC après adaptation des paramètres 54
2.5.7 Exemples et résultats 55
2.6 Matrice de comparaison 56
2.6.1 Principe 57
2.6.2 Construction 57
2.6.3 Remarques 58
2.6.4 Fusion inter-caractéristiques 58
2.6.5 Exemples et résultats 60
2.7 Conclusion 61

3 CHAPITRE 3 : MESURE DE SIMILARITE 64
3.1 Introduction 65
3.2 La similarité entre deux documents vidéo 66
3.2.1 Une autre définition pour un document vidéo 66
3.2.2 Définition de l'événement 66
3.2.3 Définition de l'espace des événements 67
3.2.4 Exemple 67
3.2.5 Taille des événements 69
3.2.6 Nécessité d'une mesure de similarité 69
3.2.7 Relativité de la similarité 70
3.2.7.1 Contexte de la mesure 70
3.2.7.2 Contenu versus composition temporelle 70
3.2.7.3 La détection de l'invariance 70
3.2.8 Transitivité de la similarité 70
3.3 Mesure de similarité de style 71
3.3.1 Interprétation de la matrice 71
3.3.2 Densité et répartition des votes 72
3.3.2.1 Première mesure intuitive 73
3.3.2.2 Pondération des votes 74
3.3.3 Identification de scénarios 74
3.3.3.1 Décalage constant du temps 74
3.3.3.2 Décalage variable du temps 75
3.3.3.3 Synchronisme symétrique 75
3.3.3.4 Synchronisme asymétrique 76
3.3.3.5 Cas général : combinaison de scénarios 76
3.3.4 La fonction de pondération 77
3.3.5 Normalisation des poids 78
3.3.6 Normalisation des diagonales 79
3.3.7 Définition de la mesure de similarité 80
3.4 Pseudo distance de similarité 80
3.4.1 Pourquoi une distance ? 80
3.4.2 Proposition d'une pseudo-distance 80
3.4.3 Mesure de similarité versus pseudo-distance de similarité 81
3.5 Conclusion 81

4 CHAPITRE 4 : APPLICATIONS 84
4.1 Introduction 85
4.2 Méthodologie 86
4.2.1 Extraction des caractéristiques 86
4.2.2 Lecture des matrices de similarités 86
4.3 Caractéristiques utilisées 87
4.3.1 Caractéristiques vidéo 87

4.3.1.1 Outil d'extraction baseindexvid ... 87
4.3.1.2 Les deux couleurs dominantes ... 87
4.3.1.3 La luminance moyenne ... 89
4.3.1.4 Le contraste ... 90
4.3.1.5 Les orientations et granularités de texture ... 91
4.3.1.6 Le taux d'activité ... 91
4.3.2 Caractéristiques audio ... 92
4.3.2.1 Outil d'extraction des caractéristiques audio ... 92
4.3.2.2 Modulation de l'énergie à 4 Hertz ... 92
4.3.2.3 Modulation de l'entropie ... 93
4.3.2.4 Paramètres de segmentation ... 94
4.4 Conception et mise en œuvre technique ... 95
4.4.1 Décodage vidéo ... 95
4.4.2 Outils d'extraction et de comparaison ... 95
4.4.3 Parallélisation ... 95
4.5 Expérience 1 : Etude du style d'un film de cinéma ... 96
4.5.1 Description et but ... 96
4.5.1.1 Conditions de l'expérience ... 97
4.5.2 La matrice de Matrix ... 97
4.5.3 Analyse diagonale de la matrice ... 98
4.5.4 Analyse de l'effet « PostProduction » ... 101
4.5.5 Evaluation Technique ... 101
4.6 Expérience 2 : Structuration des flux de télévision ... 101
4.6.1 Macro structuration ... 102
4.6.2 Description et but ... 102
4.6.3 Paramétrisation de tMin et tMax ... 102
4.6.4 Analyse des résultats ... 103
4.6.4.1 Analyse diagonale de la matrice ... 103
4.6.4.2 Analyse globale ... 107
4.6.5 Evaluation Technique ... 107
4.7 Expérience 3 : Classification en genre et étude de la composition temporelle des journaux télévisés ... 108
4.7.1 But et description ... 108
4.7.2 Définition de l'expérience ... 109
4.7.2.1 La recherche de la diagonale de référence ... 109
4.7.2.2 La définition des scénarios ... 110
4.7.2.2.1 Scénario 1 ... 110
4.7.2.2.2 Scénario 2 ... 110
4.7.2.2.3 Scénario 3 ... 110
4.7.2.2.4 Scénario 4 ... 110
4.7.3 Résultats ... 110
4.7.3.1 Illustrations graphiques ... 110
4.7.3.2 Interprétations des graphiques ... 111
4.7.4 Extraction automatique de l'organisation structurelle des enregistrements ... 115
4.7.5 Evaluation Technique ... 116
4.8 Conclusion ... 117

5 CHAPITRE 5 : CONCLUSION GENERALE ... 118
5.1 Contributions ... 119
5.2 Perspectives ... 120

BIBLIOGRAPHIE ... 123

ANNEXE A : QUELQUES MATRICES DE COMPARAISON 141

ANNEXE B : PARALLELISATION 149

ANNEXE C : ELEMENTS SUR LE FILM MATRIX RELOADED 155

0 Chapitre 0 : Introduction générale

Chapitre 0

INTRODUCTION GENERALE

Être capable de dire que deux documents audiovisuels se ressemblent beaucoup, est une affirmation qui n'est pas uniquement subjective. Les travaux de cette thèse concernent le problème de la comparaison des documents audiovisuels par la mise en évidence d'éléments communs.

0.1 Motivation

Dans le domaine en pleine expansion de la vidéo numérique, les documents audiovisuels disponibles sont maintenant présents en quantité importante même dans les foyers. En effet, l'évolution rapide de la technologie a permis au grand public de s'équiper de matériels multimédias sophistiqués. Le marché actuel laisse supposer que d'ici quelques années, les lecteurs enregistreurs numériques associés à des disques durs de capacité conséquente, destinés à conserver sur le plus ou moins long terme des programmes télévisés, seront largement répandus. Différents problèmes seront susceptibles de se présenter :

- Les programmes préférés seront noyés au milieu des flux enregistrés.
- Il y aura plusieurs genres de programmes enregistrés, correspondants aux différents centres d'intérêt d'une ou de plusieurs personnes.

A ces deux problèmes s'ajoute celui de l'encombrement progressif du disque qui devra conduire au développement de stratégies de nettoyage ou de classement automatique des enregistrements. Pour aussi classique soit-il, ce problème n'est pas résolu. Comment indexer et classer les enregistrements pour les sélectionner en vue de leur suppression ou de leur sélection ultérieure.

La comparaison des contenus dans le domaine de l'audiovisuel trouve sa place dans de nombreux contextes applicatifs. Elle est une opération de base pour différents types d'analyse des contenus, en complément de la détection, de l'identification et de la localisation.

0.2 Problématique et objectifs de la thèse

0.2.1 Problématique

Des approches classiques de comparaison des documents audiovisuels se basent essentiellement sur l'ensemble des caractéristiques de bas niveaux en les considérant comme des vecteurs multidimensionnels.

D'autres approches se basent sur la similarité des images composant la vidéo sans tenir compte de la composition temporelle du document ni de la bande son.

Le défaut que l'on peut reprocher à ces méthodes est qu'elles restreignent la comparaison à un simple opérateur binaire robuste au bruit. De tels opérateurs sont généralement utilisés afin d'identifier les différents exemplaires d'un même document.

L'originalité de notre démarche réside dans le fait que nous introduisons la notion de la similarité de style pour la comparaison des documents vidéo. Ces critères sont plus souples, et n'imposent pas une similarité stricte de toutes les caractéristiques étudiées à la fois.

0.2.2 Objectifs

Le premier objectif est de définir un mécanisme permettant la comparaison des documents audiovisuels. La comparaison en elle-même n'est pas un objectif applicatif à atteindre dans une phase d'analyse vidéo. C'est plutôt un outil à utiliser pour accomplir certaines tâches.

En conséquence, ces mécanismes, pour qu'ils soient utilisables, doivent être définis dans les conditions les plus générales possibles. Nous entendons par conditions générales les points suivants :

- la méthode de comparaison, en sa conception ne doit pas dépendre de l'application. Cette conception doit être indépendante
    - du genre des documents traités,

- d'un ensemble spécifique de caractéristiques audiovisuelles,
- de la durée des documents traités.

– le résultat de la comparaison doit pouvoir être quantifiable. Nous devons pouvoir extraire une information numérique à partir du processus de comparaison. Lorsque nous comparons un document avec un ensemble de documents, nous devons être capables de comparer les résultats de ces comparaisons.

Le deuxième objectif est de définir une mesure de similarité pour les documents vidéo. Cette mesure doit en revanche être paramétrable pour répondre à des observations particulières, notamment, lorsque les documents comparés ne présentent qu'une ressemblance partielle. Cette ressemblance partielle se manifeste selon trois aspects :

– qualitative : c'est la ressemblance des segments vidéo pour un certain sous-ensemble variable de caractéristiques audiovisuelles,

– quantitative : c'est la ressemblance entre un sous-ensemble de segments vidéo appartenant à chaque document.

– temporelle : il s'agit de prendre en compte l'ordonnancement des segments similaires dans les deux documents.

0.3 Organisation de la thèse

Cette thèse comporte quatre chapitres.

- Le ***chapitre 1***, « Etat de l'art », concerne essentiellement les :
  a. Mesures de similarité pour les documents vidéo, et les
  b. Méthodes de comparaison des séries chronologiques.
- Dans le ***chapitre 2***, « Matrice de comparaison », nous introduisons la méthode de comparaison que nous proposons. Trois algorithmes sont au cœur de cette méthode :
  a. L'algorithme de l'Intersection Quadratique Récursive conçu pour être appliqué aux séries chronologiques en général. Il a pour but de rechercher toutes les séquences semblables à l'intérieur de deux séries.
  b. L'algorithme d'Extraction des Séquences Similaires de tailles Variables à partir de deux séries.
  c. L'algorithme du Calcul de Couverture qui estime la ressemblance entre deux séquences comparées.

Ensuite nous présentons les résultats de la comparaison de deux documents vidéo sous la forme d'une Matrice de Comparaison.

- Nous proposons dans le ***chapitre 3***, intitulé « Mesure de similarité », un ensemble de réflexions sur la similarité entre les documents vidéo. En particulier nous évoquons deux points essentiels :

    a. le degré de ressemblance, et

    b. l'organisation temporelle des éléments semblables.

  Nous définissons ensuite une mesure de similarité qui peut prendre en compte plusieurs scénarios de comparaison. Une pseudo distance est proposée à la fin de ce chapitre.

- Le ***chapitre 4*** s'intitule « Applications ». Avant de détailler les trois expérimentations principales qui font l'objet de ce chapitre, nous revenons sur l'environnement de développement de nos outils, ainsi que sur la description des caractéristiques audio et vidéo utilisées dans le processus de comparaison.

Nous concluons ce manuscrit de thèse par une synthèse des contributions que nous avons apportées. Enfin, nous évoquons les principaux points abordés qu'il serait souhaitable d'approfondir et d'où nous tirerons des pistes pour définir les perspectives majeures pour de futurs travaux.

1 Chapitre 1 : Etat de l'Art

Chapitre 1

ETAT DE L'ART

1.1 Introduction

Ce chapitre ne prétend pas contenir une présentation exhaustive des méthodes développées par l'ensemble des équipes de recherche qui ont travaillé sur la vidéo, mais vise plutôt à dresser une classification des divers thèmes abordés dans ce document, dans le but de les situer dans leur cadre de mise en oeuvre.

Ces thèmes ou points essentiels sont les mesures de similarités sur les documents vidéo et les méthodes de comparaison sur les séries temporelles (dites aussi chronologiques). Nous allons essayer, tout au long de ce manuscrit, de nous rapporter à ce chapitre, à chaque fois que nous parlerons d'un thème abordé pour tirer un bilan et effectuer des comparaisons.

Deux grandes parties structurent donc ce chapitre. La première consiste à dresser un état de l'art de l'analyse des documents vidéo du point de vue du contenu en présentant plus particulièrement les méthodes de comparaison et les mesures de similarité des documents. La deuxième se focalise essentiellement sur les liens entre les travaux sur les séries chronologiques et la proposition que nous formulerons dans le chapitre suivant.

1.2 Partie vidéo : Mesures de similarité

L'analyse de la vidéo à travers ses caractéristiques bas niveau a déjà fait l'objet de nombreuses propositions, et représente l'approche la plus immédiate puisque l'on dispose de moyens concrets, plus ou moins simples, pour extraire des caractéristiques audiovisuelles pour démarrer cette étude.

1.2.1 Qu'est ce qu'un document vidéo ?

Sans rentrer dans les différents types de codage et de compression, un document vidéo peut être vu comme étant la combinaison de deux modes : la vidéo et l'audio, représentés dans un espace discret de temps associé, en général, à une fréquence d'échantillonnage plus élevée que les changements d'états qu'il reflète, ce qui permet au spectateur de le percevoir comme étant continu.

Pour simplifier, on peut le définir comme un ensemble de séquences d'images synchronisées avec une ou plusieurs bandes son, formant un tout complet. Ses niveaux hiérarchiques sont : le document complet, l'unité narrative, la scène, le plan, puis l'image. Cette dernière, n'ayant pas un équivalent structurel sur la bande son (la « slice » audio associée à une image dans les fichiers mpeg n'a pas d'autre sens que celui d'être un fragment sonore sur une durée d'1/25ème de seconde) et ne reflétant pas non plus la notion du temps, ne peut former à elle seule un document vidéo : au moins deux images sont nécessaires pour cela.

Jusqu'à nos jours c'est le document le plus riche, en matière de sens. Il est composé des seuls deux sens transportables à distance, parmi les cinq avec lesquelles l'homme peut communiquer. Nous comprenons donc pourquoi ce genre de documents prend toute cette importance dans notre vie quotidienne et, par conséquence, dans nos laboratoires.

1.2.2 Analyse vidéo : où sommes nous actuellement ?

La segmentation vidéo comprend la segmentation temporelle, telle que la détection des changements de plan et la détection d'effets de transition spéciaux, et la segmentation spatio-temporelle, telle que la segmentation en objets et leurs suivis. L'indexation vidéo comprend l'indexation bas niveau, telle que l'utilisation des caractéristiques ou des descripteurs comme la couleur, la texture, la forme et le mouvement; l'indexation de niveau sémantique, dite aussi haut niveau, telle que la classification de plans, la segmentation en unités narratives, la détection et l'identification des personnes, et la construction des résumés vidéo, tel que les résumés par images clés ou par détection d'événements importants [Tekalp 04].

1.2.2.1 La segmentation vidéo

On distingue deux types de segmentation vidéo : la segmentation temporelle et la segmentation spatio-temporelle.

1.2.2.1.1 La segmentation temporelle

La segmentation en plans est la technique de segmentation temporelle des enregistrements vidéo la plus répandue et la plus utilisée. Les méthodes de détection de changements de plan localisent les images, à travers lesquelles de grandes différences sont observées dans un certain espace de caractéristiques [Gargi 00, Lienhart 01, Hanjalic 02]. L'espace de caractéristiques se compose habituellement d'une combinaison de couleur et de mouvement. Les changements de plan peuvent être instantanés (cuts) ou apparaître sur plusieurs images, appelés les effets de transition progressifs, tels que les fondus et les volets. Il est plus facile de détecter des cuts que des effets progressifs.

La méthode la plus simple pour la détection des cuts est d'analyser les variations d'intensité des pixels entre les images successives. Si un nombre prédéterminé de pixels montre des différences plus grandes qu'une certaine « valeur seuil », alors l'occurrence d'un cut peut être déclarée.

Une approche légèrement différente consiste à diviser chaque image en blocs rectangulaires, à opérer des évaluations statistiques dans chaque bloc indépendamment, et à vérifier alors que le nombre de blocs qui ont globalement été modifiés est supérieur à un seuil. Les deux approches peuvent être sensibles au bruit et à la compression. Cependant, il existe de nombreuses solutions qui s'appliquent à la vidéo de manière générique avec une précision plus qu'acceptable [Gargi 00, Lienhart 01, Hanjalic 02].

La micro-segmentation est une segmentation temporelle à une échelle encore plus petite que celle du plan. Elle est basée sur la segmentation en événements, en mouvements de caméra, en entrée-sortie d'objets ou de personnages [Joly 96]. Par opposition, la macro-segmentation effectue une segmentation qui se rapproche de la composition sémantique des documents (segmentation en séquences, en chapitres, en programmes) [Aigrain 95].

1.2.2.1.2 Segmentation spatio-temporelle

La segmentation en objets n'est pas un problème facile, principalement parce que la définition des objets vidéo exige habituellement une interprétation sémantique de la scène. Il n'est généralement pas possible de définir de tels objets, sémantiquement significatifs, en termes de caractéristiques de bas niveau, tels que des paramètres de mouvement ou de couleur. Par conséquent, la segmentation et le suivi d'objets sémantiques dans une scène sans contrainte peuvent exiger l'intervention interactive de l'utilisateur. Cependant, dans quelques circonstances bien contraintes, des objets sémantiques peuvent être segmentés et suivis entièrement automatiquement. Par exemple, dans les systèmes de vidéo surveillance [Courtney 97, Foresti 02], où la caméra est stationnaire, des objets dans la scène peuvent être extraits par des méthodes simples de soustraction et de détection de changement d'arrière plan.

1.2.2.2 Indexation vidéo

Parmi les grandes familles d'outils d'indexation dédiés spécifiquement aux contenus vidéo, on trouve les outils d'indexation de bas niveau, et ceux effectuant une analyse sémantique.

1.2.2.2.1 Indexation de bas niveau

Des descripteurs de bas niveau, tels que la couleur, la texture, la forme, et le mouvement, peuvent être associés aux plans ou aux objets. La couleur des images choisies peut être décrite par l'histogramme de couleur ou par les couleurs dominantes [Manjunath 02]. Les paramètres de mouvement de caméra et le taux d'activité décrivent le mouvement au niveau du plan [Manjunath 02, Tan 00]. Le mouvement des objets peut être décrit par des trajectoires [Dagtas 00].

Les sommaires d'images clés et les sommaires de segments importants sont généralement employés dans des applications commerciales. Les images clés, qui se rapportent à une ou plusieurs images représentatives dans un plan, fournissent une représentation compacte. Plusieurs méthodes existent pour choisir automatiquement les images clés par l'analyse des caractéristiques de bas niveau [Dimitrova 02, Antani 02].

1.2.2.2.2 Analyse sémantique

L'information sémantique peut être représentée par des annotations structurées ou du texte libre, ou par des modèles sémantiques. Les annotations peuvent être manuelles, ou extraites automatiquement à partir du sous-titrage, par la détection et l'identification de visages, de décors, ou d'actions modélisés spécifiquement. Les modèles sémantiques peuvent décrire des entités, telles que des objets et des événements, et des relations entre elles, qui rendent possible le traitement des requêtes complexes. Certains modèles sémantiques sont considérés comme prolongements des modèles d'Entité Relation (ER) développés pour des documents par les communautés de recherche de base de données et documentaire.

L'analyse sémantique de la vidéo induit généralement l'utilisation des caractéristiques cinématographiques et basées Objet. Les caractéristiques cinématographiques ont pour origine l'application de règles classiques de montage et de production, telles que les procédés de réalisation des effets de transitions. Les différentes règles cinématographiques peuvent s'appliquer à différents genres. Par exemple, les films d'action, les séries de télévision, les journaux d'infos, et toutes les émissions de sports ont différentes caractéristiques cinématographiques [Hampapur 02b, Sundaram 02, Ekin 03]. Les méthodes de segmentation et d'identification d'objets peuvent également être employées pour la détection des événements importants [Satoh 01].

1.2.2.3 Situation actuelle

Alors que de bons résultats peuvent être obtenus pour certaines tâches d'analyse vidéo telle que la détection de changement de plans sur la vidéo générique, la plupart des tâches automatiques d'analyse – et en particulier celles qui portent sur des aspects sémantiques, fonctionnent mieux sur des contenus spécifiques.

Par la suite, nous allons traiter, toujours dans le domaine de l'analyse automatique de la vidéo, le thème spécifique de la comparaison des documents vidéo.

1.2.3 Qu'est ce qu'une mesure de similarité

En général, c'est une fonction qui quantifie le rapport entre deux objets comparés en fonction des points de ressemblance et de différence. Bien entendu, ces deux objets doivent appartenir à une même classe sémantique. Il n'existe pas de définition standard en ce qui concerne la comparaison des documents vidéo.

1.2.3.1 Une mesure de similarité pour quoi faire ?

Dans le cas le plus basique, la mesure de similarité est une fonction binaire qui affirme ou nie la ressemblance de deux documents vidéo. Elle peut servir de prédicat d'égalité, c'est-à-dire qu'elle indique si deux contenus sont deux copies d'un même document. Cette notion stricte d'égalité peut être étendue pour s'adapter à la nature complexe d'un document vidéo. On considérera égaux deux documents vidéo identiques au bruit de transmission près, à la fréquence de codage près, aux petits changements dus à des variations de montage près, à la longueur près, et ainsi de suite.

L'intérêt pour ce genre de mesure est multiple. Citons :

- Les problèmes de droit de propriété : plus présents aujourd'hui en raison de l'accès, de l'édition et de la diffusion faciles des vidéos sur le web. Bien qu'on ait proposé la technique de filigranage (ou watermarking) à cette fin, cette mesure est nécessaire pour les copies qui n'ont pas pu profiter de cette technique [Hampapur 01].

- La recherche de documents vidéo : l'objectif est d'améliorer la performance des moteurs de recherche des documents multimédia qui reposent actuellement uniquement sur des informations textuelles liées, et de les rendre aussi efficaces que pour le texte. [Shan 98, Tan 99, Wu 00, Naphade 01].

- L'amélioration de la tolérance aux fautes des moteurs de recherche [Cheung 00]. Lorsqu'une vidéo demandée n'est plus accessible sur un site donné, dû à un problème d'expiration de lien, des répliques peuvent être consultées ailleurs. En outre, en trouvant des copies similaires sur le web, les utilisateurs peuvent choisir le meilleur endroit d'accès avec la meilleure vitesse de téléchargement pour faciliter leur tâche de récupération.

- Le développement de systèmes de surveillance d'émission de certains contenus pour la validation de contrats de diffusion.

Ce type de mesure de similarité peut être plus ou moins tolérant jusqu'à permettre d'identifier tout simplement des documents du même genre comme nous le proposerons

dans les chapitres suivants. Nous pouvons ainsi l'utiliser comme un outil de classification et d'extraction.

1.2.3.2 Quelques définitions

La similarité entre deux documents vidéo, telle qu'elle est vue par [Cheung02] et [Hoi 03], est donnée comme étant le pourcentage d'images ou de plans semblables partagés par ces documents. Cette mesure est semblable à celle appliquée aux documents textuels [Broder 97, Shivakumar 98], mesure qui consiste à calculer le pourcentage des mots semblables partagés. Cependant, sur la vidéo, cette mesure est plus délicate à mettre en oeuvre. Pour mesurer la similitude des images de deux séquences vidéo, l'approche typique, selon les auteurs, est de représenter chaque image par un vecteur de caractéristiques multidimensionnel (basé sur un ensemble d'attributs, tels que la couleur, la texture, la forme et le mouvement). Ensuite, la similarité entre images est calculée par une fonction de similarité appliquée sur les vecteurs correspondants.

Cependant cette définition de la similarité entre documents vidéo n'est pas partagée par tous les chercheurs. Son principal défaut est la non prise en compte de la bande son et de la dimension temporelle.

L'ordre temporel a une importance indéniable. Il s'illustre à différente échelles : dans l'ordre des unités narratives, celui des plans, ou encore celui des images dans un même plan. On peut se contenter d'étudier la relation d'ordre uniquement sur les images dans l'ensemble du document.

L'audio pouvant être d'une aide et d'une influence aussi importante que la bande visuelle, ce mode prend de plus en plus sa place dans ce genre de mesures [Herley 04].

1.2.3.3 Les différentes approches de comparaisons et de mesures.

Pour trouver un bon équilibre entre l'exactitude de la méthode de comparaison et le coût de sa mise en œuvre en termes de temps de calcul et de ressources informatiques, différentes résolutions (spatiales et temporelles) peuvent être étudiées et adaptées selon les applications.

Par exemple, pour retrouver un document vidéo unique ou distinctif dans des bases de données, l'appariement des images-clés peut être suffisant. Cependant, en recherchant dans un programme de télévision les segments vidéo semblables à un segment donné, une mesure plus sophistiquée est nécessaire.

Ce raisonnement s'applique de la même manière quand il s'agit d'effectuer une recherche de similarité partielle, où un sous-ensemble de caractéristiques d'un segment vidéo doit être apparié et non le segment en entier. Un exemple de ce dernier cas peut être la recherche de séquences vidéo filmées dans un même studio. L'arrière plan ou les couleurs dominantes sont comparables, mais les circonstances, le contexte, la personne filmée peuvent différer.

1.2.4 Etat de l'art des mesures de similarité

1.2.4.1 Mesures s'inspirant de celles appliquées aux images

La recherche des contenus visuellement similaires est un thème central dans le domaine de la recherche des images basée sur le contenu (CBIR). Pendant la dernière décennie, de nombreux algorithmes ont été proposés pour identifier des contenus visuels semblables du point de vue de la couleur, de la texture, de la forme, ou du mouvement. Ces algorithmes regroupent des caractéristiques mesurées d'une image ou d'une vidéo dans un vecteur multidimensionnel. La similarité entre deux contenus peut alors être mesurée à l'aide d'une métrique définie sur l'espace vectoriel ainsi défini.

Beaucoup d'efforts de recherches sont menés pour trouver des caractéristiques efficaces pour divers traitements sur l'image et la vidéo [Bhat 98, Hampapur 00, Castelli 01, Hampapur 01, Ma 02, Wang 03]. Puisque le nombre d'images d'une séquence vidéo est habituellement très grand, une approche typique pour mesurer la similitude est basée sur la recherche des images-clé ou des plans principaux les plus proches dans les deux séquences. Cette technique est appelée l'algorithme des plus proches voisins (NN) [Wu 00]. Ce genre de techniques a quelques inconvénients.

Premièrement, elles dépendent de la détection de changement de plans pour segmenter la vidéo. Deuxièmement, les choix des images clés et de leur nombre dépendent de plusieurs paramètres. Troisièmement, et c'est un point primordial, de telles représentations ignorent en grande partie l'action dans une vidéo, ce qui a été en partie corrigé dans certains travaux qui ont associé une information concernant le taux d'activité ou de mouvement aux images clés [Kobla 96].

En considérant qu'un document vidéo est un ensemble de plans ayant un contenu différent, il peut être représenté par un ensemble ou une série chronologique de vecteurs de caractéristiques, en prenant habituellement un vecteur par plan.

1.2.4.2 Mesures intégrant la dimension du temps

Etendre la mesure de similitude à la vidéo présente comme premier défi la définition d'une mesure simple. De multiples propositions existent dans la littérature. Par exemple, dans [Naphade 01, Adjeroh 98, Lienhart 97b], la distance de déformation que nous présenterons dans le paragraphe 3.5.2 est employée pour mesurer l'alignement temporel des différentes séquences.

Shan et Lee dans [Shan 98], ont proposé un algorithme pour mesurer la similarité des plans, basé sur la similarité des séquences d'images ou d'images clés au lieu d'ensembles d'images. [Flickner 95, Yeung 95, Zhang 97, Zhong 95] ont évoqué différents scénarios d'alignement des ces séquences représentées dans un espace de caractéristiques visuelles. Ils ont ainsi proposé différents algorithmes de mesure de similarité basés sur la programmation dynamique (cf. paragraphe 3.5) et inspirés des algorithmes d'appariements des génomes dans la bioinformatique.

Tan, Kulkarni et Ramadge, dans [Tan 99], ont proposé une mesure de similarité des documents vidéo en prenant en compte leur structuration spatio-temporelle. Ils modélisent un document vidéo en une série chronologique multidimensionnelle sur laquelle ils appliquent les techniques d'alignement de la programmation dynamique pour estimer le taux de similarité (cf. paragraphe 3.5). Ils proposent ensuite une série de contraintes à choisir selon l'application. Ces contraintes ont pour but d'adapter la mesure proposée aux exigences de l'application. Elles concernent essentiellement les tailles relatives des deux documents comparés, l'importance accordée à l'ordre temporel de leurs éléments composants, et le recouvrement.

1.2.4.3 Les méthodes avec modèles à priori

Lienhart [Lienhart 97] a proposé un travail dans lequel il détecte un ensemble connu de films publicitaires. Etant donné un ensemble de k films publicitaires (chacun étant représenté par une séquence d'images), il calcule leur « fingerprint » en se basant sur le vecteur de cohérence de couleur [Pass 96]. Des approches semblables ont été publiées dans [Hampapur 00] et [Jaimes 03].

Toutes ces méthodes visent principalement à améliorer la précision de la mesure de similitude mais ne s'attaquent pas au problème de l'efficacité.

Une visée commune de toutes les recherches ci-dessus est l'appariement de vecteurs de caractéristiques entre deux séquences vidéo. Parfois, un sous-échantillonnage peut s'avérer nécessaire pour rendre les mesures plus aisément manipulables du point de vue informatique.

1.2.4.4 Méthodes d'échantillonnage

On distingue deux types de techniques de dimensionnement de la représentation des documents vidéo pour l'estimation d'une similitude :

- Les techniques haut niveau qui approximent les vecteurs de caractéristiques par une distribution statistique. Ces techniques sont utilisées pour la classification et l'analyse sémantique, car elles sont adaptatives et robustes à l'égard des petites perturbations. Néanmoins, elles sont bâties sur une forme restreinte de modèles de densité tels que des modèles gaussiens, ou des mélanges de gaussiennes, et ont recours à des méthodes de calcul intensif comme celle de la maximisation de l'espérance pour l'estimation des paramètres [Greenspan 02, Iyengar 98, Vasconcelos 01]. En conséquence, elles peuvent ne pas être applicables pour traiter une énorme quantité et une riche diversité de contenus visuels.

- Les techniques de premier ordre qui modélisent un contenu vidéo à travers un petit ensemble de vecteurs de caractéristiques représentatifs. Une approche consiste à calculer les vecteurs représentatifs "optimaux" en réduisant au minimum la distance

entre la vidéo et sa représentation originale. Si la métrique est euclidienne de dimension finie et la distance est la somme métrique de carré, la méthode de moyens peut être employée [MacQueen 67].

1.2.4.5 Mesure de comparaison générique

A notre connaissance, les travaux pionniers sur la définition d'une mesure de similarité vidéo générique - dans le sens « indépendant du domaine d'application », sont ceux menés par Lienhart, Effelsberg et Jain dans [Lienhart 97 b].

Ils considèrent les trois aspects suivants :

- Le niveau de résolution temporelle de la comparaison : celle-ci s'applique soit au document entier, à l'unité narrative ou scène, au plan, ou à la séquences d'images. Ils supposent que la comparaison de deux grands documents de plusieurs heures ne repose pas sur les mêmes principes que celle de la comparaison de deux clips de quelques secondes chacun.

- L'ordre temporel des éléments qui est, selon les auteurs, d'une importance variable en fonction des circonstances de l'application.

- La durée de chaque élément.

Pour chacun des nivaux de résolution, les auteurs proposent des méthodes de comparaison et des mesures de similarité. Contrairement à l'approche que nous présenterons dans les chapitres suivants, les auteurs considèrent chaque caractéristique visuelle comme étant spécifique et associée à une mesure de distance appropriée.

Au niveau le plus bas, les auteurs proposent de résoudre le problème de l'appariement avec des méthodes dérivées de la distance d'édition et de la programmation dynamique. Tandis qu'au niveau le plus haut ; les auteurs décomposent le document en un ensemble d'éléments, que ce soit scènes ou plans, et proposent un algorithme pour trouver le graphe de correspondance des différents éléments composant les deux documents comparés. A ce niveau, ils considèrent que l'ordre des éléments peut être changé, voire même croisé.

Par ailleurs, les auteurs évoquent l'intérêt de la combinaison de plusieurs caractéristiques dans le processus de la comparaison, et de la pondération de ces caractéristiques selon un paramétrage correspondant aux préférences de l'utilisateur.

1.2.4.6 Mesures permettant d'identifier les copies sans modèles à priori

1.2.4.6.1 Particularité

Ce genre de mesures est très spécifique. Il s'adresse à la détection de copies d'un même document dans de grands ensembles de données audiovisuelles.

Il est impossible, dans ce contexte, de gérer des signatures de grande dimension telles que des séries chronologiques (cf. paragraphe 1.3) et la similarité exacte n'est guère une exigence. On ne vise qu'à retrouver des documents identiques à seulement quelques modifications près (de montage, de changement de l'ordre ou de réduction du nombre des plans, de bruit de transmission, ou d'encodage différent). Les mesures sont en conséquences simplifiées.

Le principal intérêt de ce genre de mesures est la rapidité de calcul. Aidé par le fait qu'elles cherchent des contenus identiques au bruit et au codage près, ce qui soulève néanmoins quelques difficultés, ces mesures exploitent un échantillonnage temporel approprié et des techniques simples et rapides. Un exemple de ces travaux de recherche est celui de [Herley 04] qui se base sur une seule caractéristique sous échantillonnée provenant du flux audio.

Les performances de ces mesures sont remarquables ainsi que la rapidité de leur mise en œuvre. Mais le problème commun de ce genre de méthodes réside dans le fait que l'évaluation est effectuée sur la recherche de différentes versions synthétisées artificiellement, d'une part, et que de nombreuses contraintes du paradigme expérimental favorisent la réussite de l'expérience d'autre part.

1.2.4.6.2 Les extractions de répétitions

Herley, dans [Herley 04], extrait les répétitions à partir de flux audiovisuels. Il identifie les répétitions sans connaissance a priori et sans apprentissage.

Herley propose de détecter, par opposition aux approches avec modèles à priori, [Lienhart 97], des séquences *inconnues* qui se répètent dans le flux audio(visuel) en l'absence de toute hypothèse sur ce que pourrait être une telle séquence comme, par exemple, un événement se distinguant des événements qui l'entourent. Ces répétitions, selon lui, peuvent être des films publicitaires, des chansons, des génériques, et mêmes des programmes entiers. Il impose néanmoins une certaine durée aux objets qu'il cherche (30 secondes). Il extrait une caractéristique du flux audio en temps réel. Les répétitions des séquences sont supposées être à l'identique selon cette caractéristique, au bruit de rediffusion près, et les objets successifs ne sont pas supposés se recouvrir.

Son approche est la suivante. Des objets audio ou vidéo se répètent aléatoirement de temps en temps dans le flux. La partie restante est considérée comme séparant les répétitions les unes des autres. Il associe à chaque objet une probabilité de répétition et étudie du point de vue probabiliste les chances de répétitions de ces objets en fonction de la taille de la fenêtre temporelle observée, des tailles des objets répétés, et de la fréquence moyenne de répétition des objets dans le flux. En ce qui concerne la recherche de répétition elle-même, il divise le flux en N blocs et compare séquentiellement le bloc courant avec tous les autres. Dans le cas où il y a identification d'une répétition (par corrélation simple), il ajoute le bloc à une liste dédiée. Une amélioration consiste à vérifier l'existence des blocs d'abord dans la liste construite petit à petit, et ensuite dans le flux divisé en blocs. Cette liste de blocs constitue l'ensemble des objets dont la répétition est reconnue.

La caractéristique utilisée est issue de la caractéristique de l'audio. L'auteur estime que cette méthode s'applique indirectement au contenu vidéo puisque la répétition dans la vidéo engendre aussi une répétition dans le flux audio qui l'accompagne. La caractéristique est réechantillonnée à raison de 11 valeurs par seconde, ce qui supposé suffisant pour localiser la ressemblance entre deux éléments comparés. L'auteur détermine ensuite les points limites du morceau répété en opérant un glissement des blocs comparés en avant et en arrière pour localiser la divergence indiquant le début et la fin de l'itération.

La fonction de comparaison est apparemment binaire ; les deux morceaux sont identiques, ou pas. Une telle approche est intéressante dans certaines applications. Citons par exemple le cas d'un enregistrement audiovisuel de plusieurs heures, supervisé automatiquement, duquel on filtre toutes les répétitions pour réduire l'espace de stockage.

1.2.4.6.3 Recherche dans les bases de données

Nous distinguons deux types d'approches : celles qui se concentrent sur la recherche d'une séquence vidéo (contenant un élément de caractérisation décrit) à l'intérieur d'une base de données moyenne, et celles qui sont plutôt dédiées à la recherche d'un document vidéo complet à l'intérieur de grands ensembles de données.

Pour la recherche des copies d'un même document dans les grandes bases de données, Hoi, dans [Hoi 03], propose un cadre pour la détection de similitudes en deux phases, basé sur deux genres de signatures ayant différentes granularités : les signatures grossières et les signatures fines. Les deux phases de la recherche sont :

1. la recherche rapide et efficace des documents similaires
2. la détection ou la vérification de la similarité effective

Les signatures brutes sont produites en se basant sur l'histogramme de densité des points (chaque point représente une image ou un plan dans un espace multi caractéristique), et en transformant l'espace original des points en un espace bidimensionnel subdivisé en pyramides [Berchtold 98]. Dans une première phase de comparaison, basée sur ce genre de signature, la majeure partie des séquences vidéo non comparables à la requête, est rapidement filtrée.

Les signatures fines sont obtenues en calculant une forme simplifiée de l'évolution des caractéristiques des séquences vidéo. Lors de la seconde phase, la mesure de similarité est obtenue en comparant ces formes simplifiées. De ce fait, Hoi tient compte, dans la comparaison de l'ordre temporel des séquences.

1.2.5 Recherche sans considération de l'ordre temporel

Un peu avant Hoi, Cheung et Zakhor, dans [Cheung 02] avaient proposé une mesure de similarité efficace (c'est-à-dire d'un coût de calcul modéré), qui se range dans ce même

contexte, mais qui ne prend pas en compte la dimension temporelle, c'est-à-dire de l'ordre des images dans l'espace des séquences vidéo.

Dans [Cheung 02], on projette les images des deux séquences vidéo comparées dans un espace bidimensionnel et on cherche à identifier les ressemblances en découpant cet espace en sous-espaces et en estimant les sous-espaces communs aux deux séquences. Cheung et al. proposent ensuite des techniques d'approximation probabilistes pour accélérer la vitesse de calcul de cette mesure en limitant la perte d'efficacité.

En dehors des applications bien précises comme celles que nous venons de citer, nous pensons que ces méthodes ne seront pas les meilleures pour aborder l'évaluation de la similarité de style où la ressemblance exacte entre les deux contenus n'est plus un critère à observer.

1.2.5.1 Mesures permettant la classification en genre

Il existe de très nombreuses références sur les problèmes de classification des vidéos en genres. Citons ici en particulier [Bruno 04] qui abordent la définition d'une mesure de similarité entre documents vidéo par l'intermédiaire des descripteurs audiovisuels vus comme des séries chronologiques. Dans leurs travaux, tous les descripteurs d'un document sont regroupés pour devenir une série multidimensionnelle. Ils comparent ensuite deux séries en utilisant la technique des machines à vecteurs supports (SVM) [Mukherjee 97]. Leur approche est particulièrement utile dans le domaine de la reconnaissance de mouvement dans les séquences vidéo (par exemple aller/retour) et classification de vidéos en genre (sports/ journal télévisé (présentateur)).

1.2.6 Limites

Le fait de représenter chaque image dans la vidéo par un vecteur multidimensionnel ne se prête pas à la comparaison de style. Ce point de vue se défend par le fait que les images peuvent comporter seulement un sous-ensemble de caractéristiques similaire entre deux séquences vidéo. Or, ce sous ensemble peut être suffisant pour signaler une similarité. Par exemple deux films peuvent être considérés comme similaires uniquement parce qu'ils ont tous les deux un rythme rapide qui caractérise leur structuration.

La mesure de similarité que nous cherchons à établir doit rendre compte de la similarité du style. Cette notion est relativement complexe et sa définition peut s'appuyer sur une très grande diversité de caractéristiques. Considérant que l'ensemble des caractéristiques qu'il est possible d'analyser automatiquement dans un contenu audiovisuel est un ensemble ouvert, nous chercherons à élaborer une méthode générique quant aux caractéristiques utilisées. De ce fait, la mesure sera certes dépendante de l'ensemble choisi, mais elle sera en contrepartie applicable dès lors que de telles caractéristiques seront disponibles, et adaptable en fonction des contenus et des applications. Cette généricité de la comparaison vis-à-vis des traitements de bas niveau est un atout qu'aucune autre approche ne propose.

D'une manière générale, le concept de similitude sur lequel nous allons bâtir notre proposition se rapproche le plus de celui développé dans [Lienhart 97b] qui s'appuie sur les notions de niveaux de comparaison et d'importance accordée à l'ordre et à la durée segments.

1.3 Séries chronologiques et méthodes de comparaison

1.3.1 Définitions

Les séries chronologiques sont exploités dans des domaines très variés tels que l'analyse de séquences vidéo, la mobilité des animaux, l'identification de langue de signe, l'utilisation des téléphones mobiles, les analyses thérapeutiques, le traitement du signal ou encore le décodage du génome. Cette diversité applicative suscite donc de nombreux travaux de recherche au point de faire des séries chronologiques un dénominateur commun entre les communautés de la bioinformatique, du commerce, du multimédia, de l'astronomie, etc.

Pour cette raison, dresser une synthèse de l'état de l'art dans ce domaine, se révèle une tâche difficile, malgré la richesse des ouvrages et des revues scientifiques sur ce thème. On trouve ainsi nombreuses variantes d'un même algorithme selon qu'on l'applique au traitement de texte ou à la bioinformatique. Nous avons essayé, dans cette section, de donner les lignes essentielles et de citer certaines des références les plus marquantes dans ce domaine afin de rendre compte l'évolution générale de cette thématique.

Pour commencer, voici quelques définitions :

Définition. *Série chronologique.* Ce sont les valeurs successives d'une variable dans le temps. On distingue les séries chronologiques continues et discrètes. Graphiquement, une série chronologique est représentée avec le temps en abscisse et les valeurs de la fonction en ordonnée [Nasa].

Définition. *Séquence.* Une séquence est une sous partie d'éléments contigus d'une série chronologique.

Définition. *Sous séquence.* Une sous séquence d'une séquence est une sous partie d'éléments contigus de la séquence. Par suite, une sous séquence est aussi une séquence par rapport à la série chronologique

1.3.2 Notation

Un séquence ou une série chronologique est un ensemble ordonné de valeurs réelles. Le *i*ème élément d'une séquence X est noté $X[i]$. Une sous-séquence de X, notée $X[i, j]$, est le sous ensemble de X, composé des éléments de X, allant de l'ordre i jusqu'à l'ordre j. La longueur de la séquence $X[i, j]$ est égale à $j - i + 1$. La relation $<$ définit une relation d'ordre

totale sur les éléments de X avec $X[i] < X[j]$ *si et seulement si* $i < j$. On suppose que l'unité du temps est la même à travers toutes les séquences.

1.3.3 Formulation du problème de la comparaison de séries chronologiques

Le problème est tout simplement le suivant. Etant données deux séries chronologiques X et Y, de longueur respectives n et m, il s'agit de définir une méthode qui permet leur comparaison et d'en tirer une information de distance mesurant leur similarité. Différentes formulations du problème sont à aborder. Citons par exemple le cas particulier où X et Y sont de même longueurs, c'est-à-dire $n=m$. Un autre cas consiste à considérer X comme une requête à apparier avec toutes les séquences dans une base de données. Ces séquences peuvent être toutes de la longueur de X, ou d'une longueur variable. Dans ce dernier cas, et lorsque les dimensions varient, il s'agit d'extraire les sous séquences, dans les séquences de la base de données, semblables à X.

On peut aussi considérer le problème de normalisation des séquences avant le traitement, de la translation, du ré-échantillonnage, qui peuvent être mis en oeuvre par des fonctions de transformation, généralement linéaires. Il s'agit alors de trouver la fonction de transformation qui optimise la similarité.

1.3.4 Les solutions par utilisation de distances

1.3.4.1 Distance de Minkowski

L'approche la plus simple pour définir la similitude entre deux séquences est de ranger chaque séquence dans un vecteur et puis d'employer une p-distance pour définir la mesure de similarité.

Considérons deux séquences $X = x_1, x_2, ..., x_n$ et $Y = y_1, y_2, ..., y_n$. La p-distance entre deux vecteurs n-dimensionnels X et Y est définie par :

$$L_p(\bar{x}, \bar{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (1)$$

Pour p=2, on mesure la distance euclidienne ; pour p=1, c'est la distance de Manhattan. Diverses approches ont employé, adapté ou étendu cette métrique [Agrawal 93, Yi 00, Gionis 99, Goldin 95, Keogh 01, Kahveci 01, Chan 99, Faloutsos 94, Rafiei 97, Shahabi 00, Singh 98].

Cependant, une p-distance peut par elle-même être insuffisante pour décrire la similitude. Pour les séries chronologiques, cette manière de mesurer la similitude ou distance n'est pas appropriée, puisque les séquences peuvent contenir des « valeurs parasites», avoir différentes échelles ou un décalage constant d'amplitude, et des fréquences d'échantillonnage différentes. Les valeurs parasites sont des valeurs qui sont des erreurs de transmission ou de

mesures et ne sont en aucun cas révélatrice du contenu réel. Elles doivent idéalement être omises lors de la comparaison des séquences. Ce type de valeur, s'il est pris en compte, peut engendrer une grande distance de similarité entre deux séquences presque identiques.

D'autres distances plus performantes, tenant compte de la variance des coefficients par exemple comme la distance de Mahalanobis, peuvent être envisagées.

Néanmoins dans ce contexte, à notre connaissance, [Agrawal 93] est le premier travail qui propose une solution pour l'évaluation d'une similitude entre séquences temporelles qui repose sur les principes énoncés ci-dessus. Dans [Agrawal 93], on suppose que toutes les séquences sont de la même longueur, et chaque séquence est considérée comme un point dans un espace N dimensionnel. Puis, deux séquences sont considérées semblables quand la distance euclidienne entre les deux points est inférieure à une valeur seuil ε.

Faloutsos et al. ont étendu la méthode proposée dans [Agrawal 93] pour localiser les sous-séquences pouvant être appariées à une séquence donnée ou à une sous séquence d'une séquence donnée [Faloutsos 4].

Diverses métriques non Euclidienne ont été employées pour calculer la similitude des séquences. Ainsi, le modèle de Landmark par Perng, Wang, et Zhang [Perng 00] choisit seulement un sous-ensemble de valeurs, correspondant aux points maximaux, pour représenter les séquences correspondantes. Les auteurs définissent la distance à partir de deux tuples de valeurs, l'un représentant les coordonnées temporelles et l'autre l'amplitude de ces maximums. Cette distance peut être rendue invariante face à quelques transformations de base (décalage, changement d'échelle en temps et en amplitude, déformation de l'axe du temps, déformation ou re-dimensionnement non uniforme de l'amplitude).

1.3.4.2 Changement d'espaces de représentation

La transformée de Fourier discrète (TFD) est employée pour l'extraction de caractéristiques des séries chronologiques, puisqu'elle préserve l'ordre donné par la distance euclidienne entre les séquences. Le fait que cette transformation préserve la distance, et qu'elle autorise une comparaison plus ou moins fine selon le nombre de coefficients de Fourier utilisé, rend la TFD particulièrement attrayante pour l'indexation. Cependant, elle ne peut être employée que seulement pour des séquences de même longueur. En outre, elle n'est pas très efficace pour le traitement de séquences comportant des éléments non corrélés (tels que des vecteurs aléatoires).

Agrawal, Faloutsos, et Swami [Agrawal 93] ont développé une des premières solutions de ce genre. Ils ont transformé les séquences dans le domaine fréquentiel en employant la TFD. Plus tard, ils ont réduit la complexité de la comparaison en ne gardant que seulement les quelques premiers coefficients de Fourier.

Chan et Fu [Chan 99] ont utilisé la transformation en ondelettes de Haar pour réduire le nombre des cœfficients et ont comparé cette méthode à celle de la TFD. Ils ont constaté que leur méthode donnait de meilleurs résultats.

En ce qui concerne l'égalité des longueurs, Faloutsos et al. [Faloutsos 94] ont généralisé le travail de [Agrawal 93] pour permettre l'appariement des sous séquences. Les séquences de données peuvent être de différentes longueurs mais la séquence faisant office de requête doit être plus petite que n'importe laquelle de ces séquences.

Ces premiers travaux présentent les limitations suivantes pour être utilisés dans des applications pratiques :

- la mesure de similarité utilisée pour comparer les séquences est généralement la distance euclidienne. Cette distance est sensible au bruit.
- les problèmes de différence d'échelle, d'amplitude et de translation sur l'axe de temps n'ont pas été traités.
- le problème d'un appariement partiel des séquences n'est pas traité.

C'est pour palier ces problèmes qu'ont été introduites les solutions par programmation dynamique.

1.3.5 Programmation dynamique

1.3.5.1 Définition

Définition. *Programmation Dynamique*. C'est un ensemble de méthodes d'optimisation pour répondre à un problème de planification posé lorsque des décisions doivent intervenir à des périodes discrètes et qu'à chaque période, un nombre fini (petit) d'options de décisions peuvent être prises [pestmanagement]. C'est une classe de méthodes pour un problème de décision séquentiel avec une structure de coût composé.

Richard Bellman est l'un des principaux fondateurs de cette approche [cps]. Une ou plusieurs variantes de la programmation dynamique sont à la base de toutes les techniques d'alignement existantes [calliope].

1.3.5.2 La comparaison par déformation temporelle dynamique (DTW)

Une de ces méthodes est basée sur la technique de gestion de la déformation sur l'axe du temps (DTW – Dynamic Time Warping). Elle a d'abord été employée pour apparier des signaux pour la reconnaissance de la parole [Sakoe 78]. Berndt et Clifford [Berndt 94] ont proposé d'employer cette technique pour mesurer la similitude de séries temporelles dans le domaine de la fouille de données. D'autres travaux récents ont également employé cette mesure de similitude dans la fouille de données [Keogh 00, Park 00b].

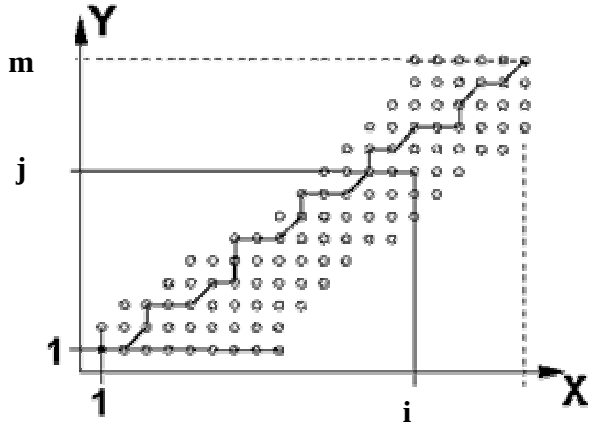


Fig. 1. Figure illustrant un chemin parcouru entre deux vecteurs de longueur différente. [Delfabro 01]

La DTW, est un algorithme qui permet de trouver un appariement optimal entre deux séquences. Le principe de base consiste à trouver un chemin selon certaines règles pour minimiser l'ensemble des distances entre les vecteurs. [Delfabro 01]

Plus précisément, pour une séquence de test X et pour chaque séquence de référence Y_k, on considère une matrice D de dimension ($N * J(k)$) (où N et $J(k)$ sont respectivement le nombre de vecteurs dans la séquence de test et de référence). A chaque entrée (n, j) de cette matrice, on associe la distance locale $d(x_n, y_j^k)$.

Pour rechercher la meilleure distance $D(X, Y_k)$ entre la séquence de test X et la séquence de référence Y_k, il suffit alors de rechercher le "chemin" dans cette matrice D pour aller du point initial (1, 1), correspondant au début des deux séquences, au point final (N, $J(k)$), correspondant à la fin des deux séquences en progressant par voisinage de cellule en cellule de façon à minimiser la somme des distances locales rencontrées.

La mise en oeuvre de cet algorithme se fait de façon simple en calculant, pour chaque entrée (i, j) la distance cumulée $D(i, j)$ correspondant à la distance optimale que l'on obtient au fur et à mesure de la progression. On peut facilement montrer que cette distance peut se calculer en utilisant la récurrence suivante :

$$D(i,j) = d(i,j) + \min_{p(i,j)} \{D(p(i,j))\} \quad (2)$$

avec :

- $p(i, j)$: ensemble des prédécesseurs possibles de l'élément (i, j)
- $D(i, j)$: distance globale
- $d(i, j)$: distance locale

Les prédécesseurs peuvent être choisis afin d'obtenir une trajectoire monotone et plausible. Celle-ci doit rester le plus près possible de la diagonale. On peut également associer des pénalités aux prédécesseurs les moins probables afin de favoriser certains profils de chemin. L'algorithme implémenté tient compte des zones où le calcul des distances ne fait pas sens. C'est pour cette raison en particulier que la zone supérieure gauche et la zone inférieure droite ne sont pas calculées. Les distances locales associées sont initialisées à une valeur très élevée afin que le chemin trouvé ne puisse pas y passer.

1.3.6 La distance d'édition

Les approches précédentes utilisent la distance de déformation de temps dynamique (DTW), et la distance euclidienne pour estimer la similitude entre deux séries de temps, Ces distances sont relativement sensibles au bruit. A l'inverse, les mesures qui sont robustes aux données bruitées, violent généralement le principe de l'inégalité triangulaire en ne considérant pas les parties les plus différentes des objets. Cependant, elles doivent être considérées dans la mesure où elles sont conformes à certains mécanismes de la perception humaine. En particulier, la comparaison de différents types de données (images, trajectoire etc..), conduit souvent à une étude des parties qui sont semblables, quitte à prêter moins d'attention aux éléments de grande dissimilitude.

1.3.6.1.1 La distance d'édition pour les séquences de texte

La *distance édition* entre deux séquences alphanumériques (dites aussi *texte*) est définie dans [Crochemore 94] comme étant le nombre d'opérations minimum nécessaires pour changer une requête donnée en un motif prédéfini. Les opérations sont *effacer*, *insérer*, et *changer*. Si l'on suppose que le coût de chaque opération est 1, la distance d'édition entre deux séquences textuelles est le nombre minimum d'opérations nécessaires pour changer une des deux séquences comparées pour obtenir l'autre.

Soit $D(i, j)$ la distance minimale entre les deux séquences $X[i]$ et $Y[j]$. On définit récursivement cette distance comme suit :

$$D(i,j)=\begin{cases} j & \text{si } i=0 \\ i & \text{si } j=0 \\ D(i-1,j-1) & \text{si } i,j>0 \text{ et } X[\mathrm{i}]=Y[j] \\ \min\begin{Bmatrix} D(i-1,j-1)+1, \\ D(i-1,j)+1, D(i,j-1)+1 \end{Bmatrix} & \text{sinon} \end{cases} \qquad (3)$$

Les valeurs dans la dernière partie de cette formule correspondent respectivement aux opérations *changer*, *effacer* et *insérer*. La formule (3) induit un algorithme de recherche de la distance d'édition entre deux textes X et Y de longueur N et M avec une complexité d'ordre $O(mn)$.

Une forme particulière de l'algorithme de la distance d'édition renvoie la longueur de la plus longue sous séquence commune entre deux séquences (PLSC) [Crochemore 94]. Cette forme considère que $D(i, j)$ est le nombre minimal d'effacements, et d'insertions, mais pas d'échanges, nécessaires pour transformer un texte $X[i]$ en $Y[j]$. C'est une version restreinte de la distance d'édition où les échanges ne sont pas permis.

Beaucoup de travaux de recherche ont été menés pour déterminer les sous séquences de textes qui s'apparient approximativement à une chaîne donnée [Califano 93, Roytberg 92, Vingron 89, Wang 94, Wu 92].

1.3.6.1.2 La distance d'édition et les séquences numériques

Les séquences de textes se composent normalement d'un nombre réduit de symboles discrets, ce qui conduit à proposer des mesures de similitude et les méthodes de recherche différentes pour traiter des séries numériques de différentes longueurs.

Dans ce cas, on considère que :

1. au lieu de vérifier l'égalité stricte entre les éléments X[i] et Y[j] des deux séquences X et Y (respectivement), on vérifie qu'ils sont à une *distance d'appariement* l'un de l'autre, c'est-à-dire que *distance*($X[i]$, Y[j]) $\leq \delta$,
2. l'opération *changer* n'est pas permise dans le calcul de cette distance d'édition, et finalement,
3. pour l'opération *insérer*, les nouveaux éléments sont calculés par interpolation. Par exemple la valeur $(X[i] + X[i+1])/2$ sera insérée entre $X[i]$ et $X[i+1]$.

La formule suivante calcule la distance d'édition entre les deux séquences X et Y :

$$D(i,j) = \begin{cases} j & \text{si } i = 0 \\ i & \text{si } j = 0 \\ D(i-1, j-1) & \text{si } i,j > 0 \text{ et distance}(X[i], Y[j]) \leq \delta \\ \min\{D(i-1, j)+1, D(i, j-1)+1\} & \text{sinon} \end{cases} \quad (4)$$

Le chemin entre les éléments appariés peut être alors établit en exploitant cette formule [Bozkaya 97].

1.3.6.2 La plus longue sous séquence commune (PLSC ou LCSS)

Needleman et Wunsch [Needleman 70] ont proposé cette technique à l'origine pour trouver les similitudes entre des séquences de protéine. De nombreux travaux de recherches ont été déjà conduits sur l'emploi d'algorithmes de comparaison de séquences en bioinfor-

matique. Smith et Waterman [Smith 81] ont ainsi identifié des sous séquences moléculaires communes en recherchant la plus longue sous-séquence commune.

L'algorithme de la plus longue sous séquence commune PLSC, est un algorithme de programmation dynamique qui recherche la plus longue sous séquence commune entre des préfixes qu'on allonge tant qu'une similarité est reconnue. Sa complexité est d'ordre O(nm) (où n et m sont les longueurs des deux séquences).

1.3.6.2.1 La PLSC pour les séries chronologiques

Agrawal et al. dans [Agrawal 95] exploitent ce modèle de similarité en l'adaptant aux séquences temporelles. D'après [Agrawal 95], deux séquences temporelles seraient semblables si elles ont assez de paires de sous séquences ordonnées non recouvertes semblables. Deux sous séquences sont considérées semblables, à leur tour, si l'une peut être incluse dans une enveloppe d'une largeur donnée autour de l'autre. Le modèle laisse également la possibilité d'ignorer les valeurs parasites dans les sous séquences comparées. Les sous séquences appariées ne sont pas nécessairement alignées le long de l'axe de temps.

Plus formellement, Soit $X' = x_{i_1},...,x_{i_l}$ et $Y' = y_{j_1},...,y_{j_l}$ les plus longues sous séquences dans X et Y respectivement (X et Y chacune de longueur n), où

(a) pour $1 \le k \le l - 1$, $i_k < i_{k+1}$ et $j_k < j_{k+1}$, et

(b) pour $1 \le k \le l$, $x_{i_k} = y_{j_k}$

Nous définissons la similarité entre X et Y par $\text{sim}(X, Y) = l / n$.

Il faut noter que les deux séquences peuvent ne pas avoir la même longueur. Dans ce cas, il suffit d'ajouter des nombres factices à la séquence la plus courte pour qu'elle ait la même longueur que l'autre.

1.3.6.2.2 Transformations

Bollobás et al., dans [Bollobás 97], évoquent le besoin d'une fonction de distance qui présente des propriétés permettant de prendre en compte :

- différents échantillonnages.
- l'existence de valeurs parasites.
- le traitement de séquences de différentes longueurs. La distance euclidienne traite des séquences de longueur égale. Dans le cas de différentes longueurs nous devons décider s'il faut tronquer la série la plus longue, ou concaténer des zéros à la série la plus courte. Dans le cas général, son utilisation est compliquée et la notion de distance devient vague.

– l'efficacité. Elle doit être en juste proportion expressive mais suffisamment simple, afin de permettre une estimation efficace de la similitude.

Donc si besoin, une des deux séquences peut subir un décalage temporel et un changement d'échelle du type $f(x) = a \times x + b$, convenablement avant de procéder à la comparaison.

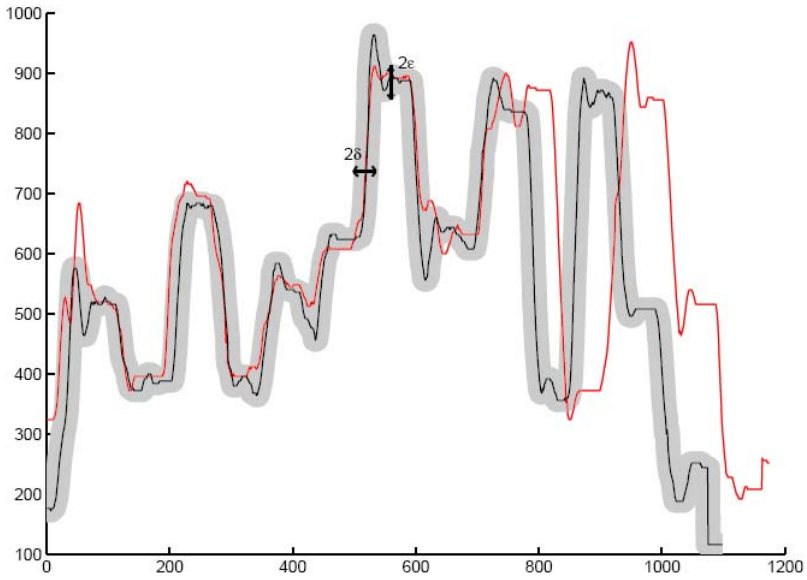


Fig. 2. La notion de comparaison par PLSC avec ε comme degré de liberté d'amplitude et δ comme degré de liberté temporelle. Les éléments dans l'enveloppe grise sont appariés. [Vlachos 02]

Soit $\delta > 0$ une constante entière, $0 < \varepsilon < 1$ une constante réelle et f une fonction linéaire de la forme $f: y = ax + b$ appartenant à la famille (infinie) des fonctions linéaires L. étant donné les deux séquences $X = x_1, x_2, ..., x_n$ et $Y = y_1, y_2, ..., y_n$, soient $X' = (x_{i_1},...,x_{i_l})$ et $Y' = (y_{j_1},...,y_{j_l})$ les plus longues sous séquences appartenant respectivement à X et Y telle que :

1. pour $1 \le k \le l\text{-}1$, $i_k < i_{k+1}$ et $j_k < j_{k+1}$,
2. pour $1 \le k \le l$, $|i_k - j_k| \le \delta$, et
3. pour $1 \le k \le l\text{-}1$, $y_{j_k} / (1 + \varepsilon) \le f(x_{i_k}) \le y_{j_k} (1 + \varepsilon)$.

Soit $S_{f,\varepsilon,\delta}(X, Y)$ défini par l / n. Alors $\mathrm{Sim}_{\varepsilon,\delta}(X, Y)$ est défini par $\max_{f \in L}\{S_{f,\varepsilon,\delta}(X,Y)\}$.

Lorsque $\mathrm{Sim}_{\varepsilon,\delta}(X, Y)$ est proche de 1, les deux séquences sont considérées comme très similaires. La constante δ est définie pour s'assurer que les positions temporelles des éléments appariés dans les deux séquences ne sont pas trop éloignées.

La constante ε, dite tolérance relative, pouvant être accompagnée d'une autre constante ε_2 dite absolue, permet un appariement approximatif des amplitudes. Quant à la fonction f, c'est une fonction de transformation linéaire (mais elle peut aussi être quadratique ou autre selon l'application) introduite pour résoudre le problème de facteur d'échelle.

1.3.6.3 Les méthodes d'approximations du calcul de la PLSC

La complexité de l'algorithme PLSC est de l'ordre de $O(n^2)$ pour des séquences de tailles n. Du fait que l'algorithme initial date depuis un certain moment, la littérature comporte une variété d'approximations pour le PLSC (voir table 1). Ces travaux ont approximé l'algorithme de PLSC pour la recherche d'une sous séquence de taille minimale égale à p (paramètre fixé), où r est le nombre de paires $(i, j) \in [1, n] \times [1, n]$ pour lesquelles $X[i] = Y[j]$. Nous citons :

Référence	Complexité de l'algorithme
[Hunt 77]	$O((r + n) \log n)$
[Hirschberg 77]	$O(p\ n + n \log n)$
[Hirschberg 77]	$O((n + 1 - p)\ p \log n)$
[Naskatsu 82]	$O(n\ (n - p))$
[Hebrard 84]	$O(p\ n)$

Table 1. Complexité en temps d'exécution pour certaines approximations de la PLSC [Simon 88]

Si l'on considère que p varie entre 0 et n, et que r varie entre 0 et n^2 nous pouvons remarquer que la complexité de ces algorithmes dans les pires cas ne descend pas au dessous de $O(n^2)$. De plus tous ces algorithmes présupposent une taille minimale à la séquence recherchée et des conditions sur l'alphabet de la séquence.

L'existence d'un algorithme linéaire pour accomplir cette tache n'a pas été démontrée, [Simon 88]. Cependant on peut trouver des approximations dont la complexité est de l'ordre de $O(n^2 / \log n)$ [Paterson 94]. En revanche, cet algorithme suppose un alphabet fini et n'est pas donc applicable pour les séries numériques.

1.4 Conclusion

Dans ce chapitre nous avons abordé deux grands axes de recherche : les mesures de similarité des documents vidéo, et les mesure de similarité des séries chronologiques. Nous avons

classifiés les mesures de similarité vidéo et cité au fur et à mesure leurs points forts et faibles ainsi que le domaine de leur utilisation.

Pour la comparaison des séries temporelles, nous avons dressé l'axe chronologique de l'évolution de ce problème ainsi que les solutions proposées.

Dans le chapitre suivant, nous présentons notre méthode de comparaison des documents vidéo basée sur la comparaison des caractéristiques audiovisuelles individuellement.

2 Chapitre 2 : Matrice de comparaison

Chapitre 2

MATRICE DE COMPARAISON

2.1 Introduction

Pour évaluer la similarité entre documents vidéo, il faut rechercher les éléments communs. Chaque vidéo est représentée par l'ensemble ses caractéristiques audiovisuelles. Ces caractéristiques sont des séries chronologiques, comme le montrent les exemples présents dans les figures 1 et 2. Le problème se ramène alors à la recherche des séquences similaires entre les deux ensembles des séries chronologiques.

On distingue deux stratégies pour procéder. La première est de fusionner chaque ensemble de caractéristiques en une seule série chronologique multidimensionnelle, et ensuite de procéder à la recherche des séquences communes. La deuxième est de fusionner les résultats issus de chaque couple de caractéristiques comparé à part. La première possibilité est réservée à la comparaison de documents très semblables, voire des versions d'un même document. En dehors de cette application, la fusion en prétraitement pénalise la similarité partielle. Nous entendons par similarité partielle la similarité sur un sous ensemble variable des caractéristiques. Nous choisissons alors la deuxième stratégie.

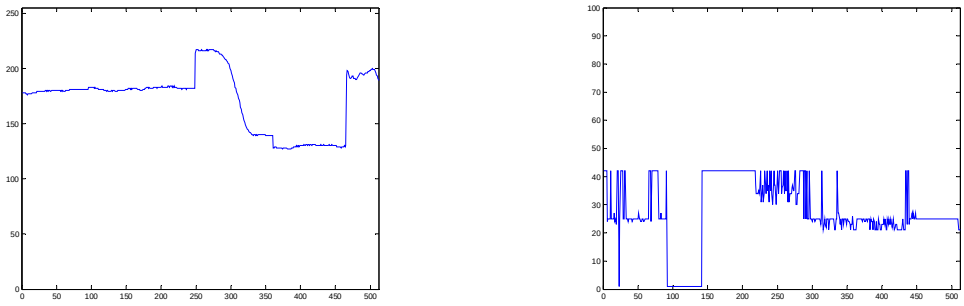


Fig. 1. Exemples de caractéristiques audiovisuelles extraites à partir d'un segment vidéo. Sur la base d'une mesure par image, le graphique de gauche représente la luminance moyenne, et celui de droite concerne la saturation dominante.

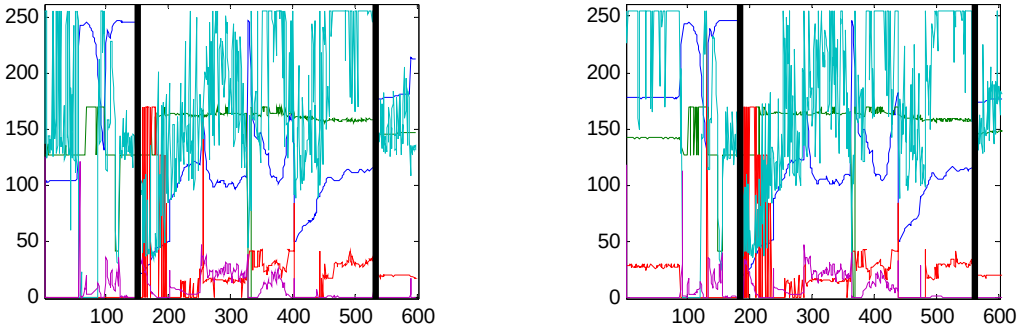


Fig. 2. Un ensemble de caractéristiques audiovisuelles extraites à partir de deux segments vidéo. Dans les vidéos, les séquences correspondantes aux intervalles entre les deux barres noires sont identiques : il s'agit d'un générique d'un jeu télévisé. Dans les graphes, on voit bien que, à l'intérieur des paires de barres noires, les caractéristiques sont appariables deux à deux.

2.2 Stratégie de comparaison des caractéristiques audiovisuelles

Les séquences communes pour un couple de séries chronologiques recherchées, sont les représentations des segments vidéo présentant des comportements semblables. Le problème essentiel se résume en les points suivants :

- les emplacements respectifs des séquences communes sont inconnus. Ils peuvent être n'importe où dans les documents
- les tailles de ces séquences ne sont pas connues non plus.

Comme ces séquences numériques représentent des séquences vidéo, nous pouvons supposer que leur taille est bornée. Mais nous ne la fixons pas. Elle varie donc dans un intervalle.

Pour trouver toutes les séquences candidates à la ressemblance, nous proposons un algorithme de recherche quadratique récursive. Une fois que nous disposons des candidats, nous présentons notre algorithme de calcul de couverture pour évaluer leur similarité effective. Ensuite, pour obtenir un schéma de comparaison globale de deux documents vidéo, nous présentons le principe de la matrice de comparaison. L'agrégation des résultats obtenus sur toutes les caractéristiques, chacune séparément, se fera via une fusion inter- matricielle ou inter caractéristiques. Nous terminons ce chapitre avec une analyse de l'information portée par cette matrice de comparaison.

2.3 Comparaison de deux séries temporelles

Dans ce paragraphe, nous présentons notre méthode conçue pour comparer deux séries chronologiques dans le but d'y extraire tous les couples de séquences similaires dont la taille est comprise entre certaines bornes.

Tout d'abord, nous proposons un algorithme général pour la détection de toutes les paires de séquences semblables d'une longueur donnée; l'algorithme d'intersection quadratique récursive (IQR). Ensuite, un second algorithme est présenté ayant pour but l'extraction de tous les couples de séquences similaires de longueurs variables (ESSV) à partir de deux séquences données.

Pour mesurer la similarité entre les séquences, nous proposons un troisième algorithme basé sur la programmation dynamique. Cet algorithme calcule la couverture (CC) de deux séquences qui est l'estimation du pourcentage des sous séquences ordonnées appariées. Nous comparons cet algorithme dont l'approche est dichotomique avec l'algorithme classique d'extraction de la plus longue sous séquence (LCSS).

2.3.1 Notations et conventions

Soit à comparer deux séries numériques X et Y de tailles respectives n et m. Nous avons : $X=[x_1, x_2, \ldots, x_n]$ et $Y = [y_1, y_2, \ldots, y_m]$.

En raison de l'approche dichotomique utilisée dans cet algorithme, les séries chronologiques doivent avoir une longueur égale à une puissance de deux. Nous supposons dans un premier temps, que les séries comparées ont la même longueur et que cette longueur est une puissance de deux. Nous verrons ensuite comment on peut étendre la méthode pour qu'elle soit applicable à deux séries de longueurs différentes et quelconques. Nous considérons donc pour l'heure le cas où $m = n$.

Considérons maintenant un prédicat d'intersection, *pot_sim*, qui peut être appliqué sur deux séquences de telle sorte que ces séquences soient considérées potentiellement similaires si elles ont une intersection non vide :

$$pot\_sim : S^2 \rightarrow B = \{VRAI, FAUX\} \text{ tel que,} \quad (1)$$

$$pot\_sim(I, J) = VRAI \Leftrightarrow \cap (I, J) \neq \varnothing,$$

où S est l'espace des séquences, et I et J sont deux séquences. Ce critère d'intersection est choisi de façon à effectuer un filtrage rapide des séquences ne présentant aucune similarité, et à cerner le champ d'investigation sur lequel nous appliquerons une fonction ultérieure qui s'occupera de la vérification de la similarité de manière plus détaillée.

2.3.2 IQR : algorithme de l'intersection quadratique récursive

L'algorithme de l'Intersection Quadratique Récursive (IQR) est un algorithme général, qui peut être appliqué dans tous les cas où on cherche à comparer des éléments décomposables, et à identifier des sous-éléments semblables. Nous le présentons comme suit.

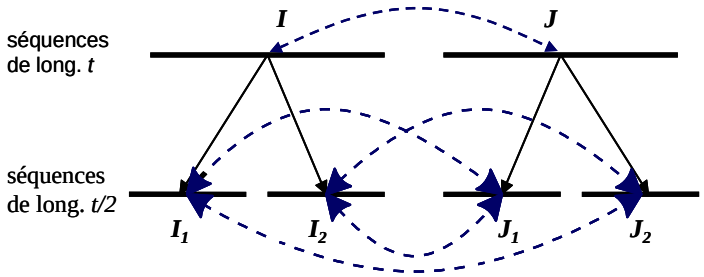


Fig. 3. Représentation de la comparaison quadratique (au niveau $t/2$) : processus principal de l'algorithme IQR. Les traits horizontaux représentent les séquences, alors que les lignes pointillées indiquent les comparaisons.

Nous procédons récursivement en commençant par les séries entières :

1. Les deux séquences, soient I et J, sont comparées à l'aide du critère d'intersection.

2. Si elles sont *potentiellement similaires*, c'est à dire $\cap (I, J) \neq \varnothing$, alors aller à l'étape suivante, sinon arrêter.

3. Si la longueur des séquences atteint une certaine valeur, *tMax*, arrêter et passer à l'algorithme suivant pour vérifier leur similarité et en extraire les sous séquences similaires, sinon continuer l'étape suivante.

4. Chaque séquence est coupée en deux sous séquences de longueurs égales, soit I en I_1 et I_2, et J en J_1 et J_2, puis

5. Une comparaison quadratique est effectuée ; cela consiste à comparer chacune des sous séquences résultantes de la division dichotomique de la première séquence avec les deux autres sous séquences de la deuxième, soit I_1 avec J_1 et avec J_2, et I_2 avec J_1 et avec J_2 (figure 3). C'est-à-dire pour chaque couple à comparer, on recommence à partir de l'étape 1 et ainsi de suite.

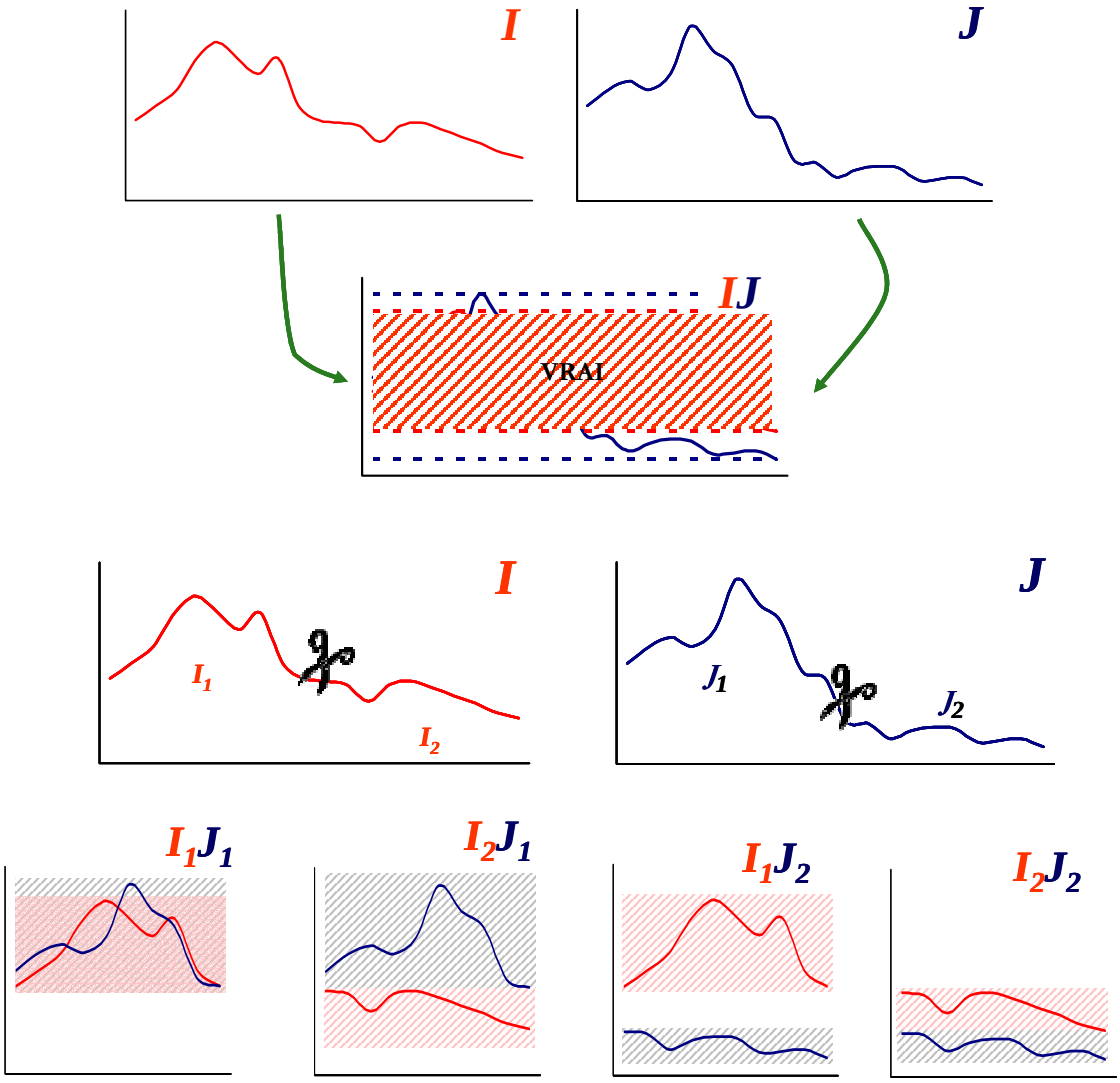


Fig. 4. Un exemple de la comparaison par l'IQR selon deux niveaux. Les trois derniers couples de séquences comparées I_1J_2, I_2J_1 et I_2J_2 ne présentent pas d'intersections de leur champ de variation. La comparaison en profondeur doit continuer alors seulement au niveau de I_1J_1.

Dans le cas où les séquences comparées ont une intersection vide, la comparaison s'arrête. Cet arrêt, à un niveau *t*, économise les opérations de comparaison inutiles et par suite, valide le choix pour lequel nous n'avons pas découpé, dès le début, les deux séries comparées en séquences de longueur *tMax*, et essayé toutes les combinaisons de comparaisons possibles. Dans l'exemple présenté dans la figure 4, deux niveaux de comparaison sont montrés. Pour le second niveau, trois parmi les quatre comparaisons engendrées sont nulles. Il s'agit de I_1J_2, I_2J_1 et I_2J_2. En conséquence, l'algorithme continue seulement à comparer I_1J_1 d'une façon récursive.

Les couples de séquences identifiés par cet algorithme sont les séquences potentiellement similaires ou pouvant contenir des sous séquences similaires. L'extraction des séquences similaires est le travail de l'algorithme qui suit.

2.3.3 ESSV : algorithme d'extraction des séquences similaires de taille variable

Nous avons fixé pour objectif l'extraction de toutes les séquences similaires de taille variant entre deux bornes *tMax* et *tMin*. Nous avons en entrée les couples de séquences de plus grande taille possible recherchée, c'est-à-dire *tMax*. Il s'agit alors, dans un premier temps, de vérifier leur similarité deux à deux. Nous remplaçons alors le critère d'intersection utilisé dans l'algorithme IQR par un nouveau critère reposant sur une analyse plus fine. A cette fin, nous utilisons une mesure appelée *taux de couverture* ou *couv*. Pour deux séquences comparées, lorsque ce taux dépasse un seuil *T*, leur similarité est affirmée. Le choix du seuil *T* dépend de l'exactitude demandée pour la mesure de similarité et du domaine d'application.

Nous avons donc :

$$sim : S^2 \rightarrow B = \{VRAI, FAUX\} \text{ tel que,} \qquad (2)$$

$$sim\ (I, J) = VRAI \Leftrightarrow couv\ (I, J) \geq T$$

Les séquences similaires peuvent contenir aussi des sous séquences similaires, dans un ordre quelconque, qui nous intéressent. Pour les identifier, nous effectuons la comparaison quadratique, quelque soit le taux de couverture des couples de séquences comparées en identifiant à chaque niveau les couples similaires. Nous arrêtons lorsque nous obtenons des séquences de tailles *tMin*. De ce fait, toutes les comparaisons possibles de séquences de taille comprise entre *tMin* et *tMax* sont effectuées.

Les étapes dans ESSV sont donc les suivantes :

1. Soient *I* et *J* les deux séquences comparées à l'aide de la fonction *couv*,
2. Si elles sont *similaires*, c'est-à-dire *couv* $(I, J) \geq T$, alors elles sont identifiées, puis,
3. Si les longueurs des séquences atteignent la borne inférieure *tMin*, arrêter, sinon continuer à l'étape suivante,
4. Chaque séquence est coupée en deux sous séquences de longueurs égales, soit *I* en I_1 et I_2, et *J* en J_1 et J_2, puis
5. Une comparaison quadratique est effectuée en recommençant avec chaque couple à partir de l'étape 1 et ainsi de suite…

À la fin de cet algorithme tous les couples des séquences semblables sont identifiés. Leurs longueurs varient entre *tMin* et *tMax*.

Les deux principales différences entre l'ESSV et l'IQR s'observent :

- lorsque deux séquences ne vérifient pas le critère de comparaison :
    - dans l'ESSV, la comparaison quadratique continue ; en effet comme la mesure de similarité est rigoureuse à ce stade, nous recherchons les sous séquences qui, elles, pourraient être similaires.
    - dans l'IQR, nous arrêtons la comparaison quadratique ; le critère étant très lâche, toute similarité potentielle dans les niveaux inférieurs aurait été reflétée dans le niveau courant.
- lorsque les deux séquences vérifient le critère de comparaison, la comparaison quadratique continue dans les deux algorithmes, mais
    - dans l'algorithme l'IQR, les séquences potentiellement similaires sont identifiées juste au dernier niveau, c'est-à-dire lorsqu'ils ont une longueur *tMax*.
    - dans l'algorithme ESSV, les séquences similaires sont identifiées à chaque niveau, de *tMax* en descendant vers *tMin*. Cette analyse exhaustive est menée dans le but d'identifier tous les couples de séquences similaires de longueurs variables.

Cette analyse quadratique présente le défaut d'un découpage automatique arbitraire des intervalles comparés (comme conséquence du découpage dichotomique), mais il remédie dans le même temps au problème des différents alignements possibles de deux séquences. Ainsi, nous pouvons simultanément tester si une chaîne est comparable à une autre et si la fin de la première est comparable avec le début de la seconde et ainsi de suite. Ceci nous servira plus tard dans la phase de fusion inter caractéristique pour identifier des zones d'appariements forts même en l'absence d'appariements strictement identiques sur toutes les caractéristiques.

2.3.4 CC : algorithme de calcul du taux de couverture

L'algorithme ESSV utilise le taux de couverture pour mesurer la similarité de deux séquences à comparer de taille *t*, telle que $tMin \leq t \leq tMax$.

Définition. *Taux de couverture* : le taux de couverture de deux séquences, *I* et *J*, de même longueur, est le pourcentage du nombre de couples d'éléments ordonnés de *I* et *J* appariés.

$$\text{couv}(I,J) = \frac{\text{nombre d'éléments appariés ordonnés de } (I,J)}{\text{nombre total de couples ordonés de } (I,J)} \quad (3)$$

Exemple. La figure 5 montre le calcul du taux de couverture entre deux séquences *I* et *J*. Les traits verticaux représentent des appariements entre les éléments similaires.

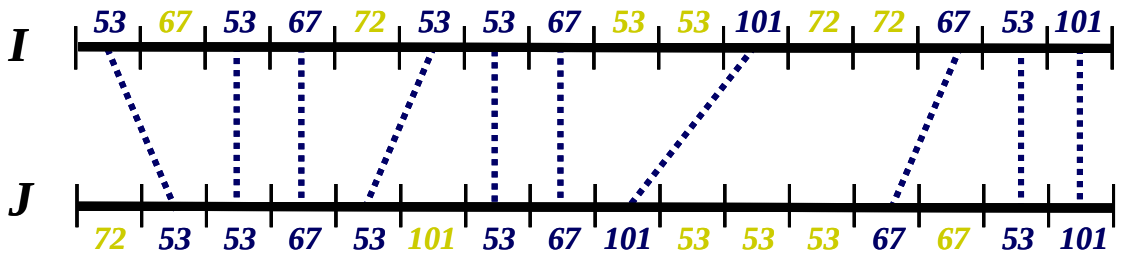


Fig. 5. Exemple de calcul du taux de couverture pour deux séquences *I* et *J*.

L'appariement se fait dans l'ordre de l'appartenance de ces éléments aux séquences *I* et *J*, sans croisement, mais tout en permettant des décalages et des sauts. Chaque séquence est composée de 16 éléments. En total, nous pouvons compter 10 appariements. Ce qui donne un taux de couverture égal à 10/16 × 100.

La fonction estime ce taux pour le meilleur alignement entre les séquences. Cette fonction est définie ainsi, lorsque *t* est la taille des sous séquences à chaque itération.

$$couv(I,J) = \begin{cases} 0 & \text{si } \cap(I,J) = \phi \\ 100 & \text{si } \cap(I,J) \neq \phi \text{ et } t = 1 \\ \frac{1}{2}\max \begin{pmatrix} couv(I_1,J_1) + couv(I_2,J_2), \\ couv(I_1,J_2), \\ couv(I_2,J_1) \end{pmatrix} & \text{si } \cap(I,J) \neq \phi \text{ et } t > 1 \end{cases} \quad (4)$$

où I_1 (resp. I_2) est la première (resp. la seconde) moitié de *I*, et J_1 (resp. J_2) est la première (resp. la seconde) moitié de *J*. (figure 6)

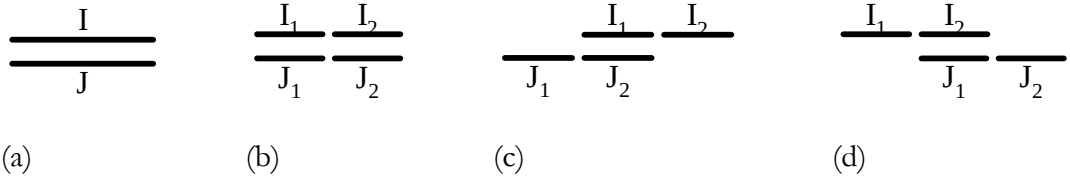


Fig. 6. Illustration des phases principales de l'algorithme de calcul du taux de couverture. (a) illustre la comparaison de deux séquences *I* et *J*. Selon l'équation (4); (b) illustre le premier argument de l'appel de *max*, (c) le second et (d) le troisième. Notons qu'on ne permet pas les comparaisons croisées pendant cette deuxième étape.

Cette fonction récursive permet de gérer les déformations et les raccourcis de la dimension temporelle, en recherchant le meilleur alignement des couples de séquences, à chaque itération et en tolérant un pourcentage de (100-T) de sous séquences non couplées.

Inspirée de l'algorithme récursif IQR, cette fonction n'effectue pas de parcours en profondeur quand l'intersection globale de *I* et *J* est vide. Quand l'intersection est vide entre deux sous séquences, c'est qu'il n'y a pas d'éléments à apparier.

2.3.5 DiSC : algorithme de comparaison Dichotomique des Séries Chronologiques

Nous obtenons ainsi finalement un algorithme général. L'algorithme de comparaison dichotomique des séries chronologiques (DiSC) qui extrait les séquences similaires à partir de deux séries chronologiques de même taille égale à une puissance de deux devient alors :

```
Début DiSC (I, J)
   si (intersection_non_vide(I,J)) alors
        si (t<=tMax) alors
             si (taux_de_couverture(I,J)>=T) alors
             le couple (I,J) est identifié[1] ;
             fin si
   fin si

   si (t>tMin)alors appel quadratique
            I est découpée en deux sous-séquences de même taille I1, I2
            J est découpée en deux sous-séquences de même taille J1, J2
            DiSC (I1, J1);
            DiSC (I2, J2);
            DiSC (I1, J2);
            DiSC (I2, J1);
   fin si
Fin DiSC
```

2.3.6 Extension pour deux séries de tailles quelconques

La division dichotomique impose que la taille de chacune des deux séries comparées soit égale à une même puissance de deux. Une extension à l'algorithme dans le but de le rendre applicable à des séries de taille différentes et quelconque serait de concaténer à la fin de chacune des séries des valeurs qui n'interviennent pas dans l'algorithme (menant à une intersection vide) et qui leur donne la longueur souhaitée, c'est à dire la puissance de deux immédiatement supérieure à la taille de la plus longue des deux séries.

On pourra ajouter par exemple des valeurs égales à la valeur maximale dans les deux séries plus une marge, à la première série et à la valeur minimale moins une marge à la deuxième.

2.3.7 Comparaison du CC avec PLSC

Un algorithme classique dans le domaine des séries chronologiques est l'algorithme de la recherche de la plus longue sous séquence commune à deux séquences, PLSC, en anglais « longest common subsequence », LCSS. Nous aurions pu utiliser cet algorithme pour calculer le taux de couverture de deux séquences qui est effectivement la longueur de la plus longue sous séquence commune. Dans ce paragraphe, nous justifions notre choix de l'utilisation de notre algorithme CC, en se basant essentiellement sur l'argument de rapidité

[1] Se référer à la section « matrice de comparaison » pour savoir comment est identifié le couple de séquences similaires.

de calcul, et d'économie en nombre d'opérations. Cet argument est fondamental lorsque la quantité de comparaisons à effectuer entre différentes séquences est imposante (ce qui est supposé être le cas dans les applications ciblées).

2.3.7.1 Algorithme PLSC

Nous rappelons brièvement l'algorithme de la PLSC, en le présentant de manière à pouvoir le comparer avec notre algorithme. Le lecteur pourra se rapporter à sa description formelle dans le paragraphe 1.3.6.2.1.

Il consiste à calculer une matrice L[0:t , 0:t] où t est la longueur de I et J, L[i,j] est la PLSC entre I[1 :i] et J[1 :j] en récurrence selon la formule :

$$L[i,\ j] = \begin{cases} 0, & \text{si } i = 0 \text{ ou } j = 0 \\ L[i-1,\ j-1] + 1 & \text{si } I[i] = J[\,j] \\ \max\left(L[i-1,\ j],\ L[i,\ j-1]\right) & \text{sinon} \end{cases} \qquad (\mathbf{5})$$

Le calcul de la couverture avec la méthode de la PLSC appliquée à notre cas, où I et J ont la même longueur sera :

```
Initialisation de L
Pour i=1..t faire
   Pour j=1..t faire
            Si I[i]=J[j] alors
               L[i,j]=L[i-1,j-1]+1;
            Sinon
               Si L[i-1,j]>L[i,j-1] alors
                    L[i,j]=L[i-1,j];
               Sinon
                    L[i,j]=L[i,j-1];
               Fin si
            Fin si
   Fin pour
Fin pour
couverture = L[t,t] / t
```

Le temps requis pour l'exécution de l'algorithme PLSC peut être justifié par les besoins, notamment lorsque ceux-ci se doivent d'être précis et rigoureux. Bien entendu, il existe aussi des approximations de cet algorithme. Dans notre travail, nous n'avons besoin que de l'estimation du taux de couverture et nous accordons une plus grande importance à la rapidité du calcul vue la grande quantité d'opérations à exécuter.

2.3.7.2 Comparaison théorique de la complexité

La complexité théorique de l'algorithme CC est la suivante :

$$\textit{complexité} \quad = \sum_{k=0}^{\log 2(t)} 4^k \qquad (\mathbf{6})$$

Où $t = 2^l$ est la taille des séquences à comparer, en puissance de deux. Cette complexité, qui peut être exprimée en $O(4^l)$ est bien la même, théoriquement, que celle de l'algorithme classique de la PLSC.

Or, cette complexité théorique étant évaluée dans le pire des cas, nous allons montrer que dans la pratique la complexité de CC est nettement inférieure à celle du PLSC.

Considérons les cas suivants :

- Cas 1 : les deux séquences comparées sont totalement différentes et ne peuvent être alignées.
    - Une seule opération suffit avec le CC pour découvrir que l'intersection est vide et le calcul s'arrête. La complexité est O (1).
    - Le PLSC, dans ce cas, effectue malgré tout, l'ensemble des opérations de comparaison.
    - Ce raisonnement peut être étendu à n'importe quelles sous séquences comparées lors de l'appel quadratique, ce qui est très probable.
- Cas 2 : les deux séquences sont parfaitement semblables, alors
    - La comparaison de $couv(I_1, J_1) + couv(I_2, J_2)$ dépasse la valeur 50, c'est-à-dire le taux maximum pouvant être obtenu par les deux autres arguments de la fonction *max* de l'équation 4. Ce n'est alors plus la peine d'effectuer le calcul de $couv(I_1, J_2)$ et de $couv(I_2, J_2)$. Ainsi, la moitié du calcul à ce niveau peut être économisé. La complexité est de l'ordre de $O(2^l)$ ainsi que celle du PLSC.
    - Là encore, ce raisonnement s'applique à tous les niveaux d'appels récursifs de l'algorithme CC.
- Dans un cas quelconque,
    - Il y a au moins un élément de la séquence qui ne s'apparie pas avec son homologue, c'est alors une combinaison des deux cas précédents, qui mène à une complexité inférieure à PLSC.

2.3.7.3 Défaut de l'algorithme CC

Nous devons mentionner que l'algorithme de calcul de la couverture CC par un moyen dichotomique, comme décrit dans le paragraphe 2.3.4, n'est qu'une estimation inférieure du taux de couverture pouvant être calculé idéalement avec la PLSC. Pour illustrer son défaut principal, nous allons examiner deux exemples. Dans ces exemples, les cadres pointillés sont présents pour montrer l'effet de la récursivité après un découpage dichotomique.

Exemple 1: Prenons les deux séquences dans la figure 7. Sur ce premier exemple, les deux algorithmes produisent le même résultat.

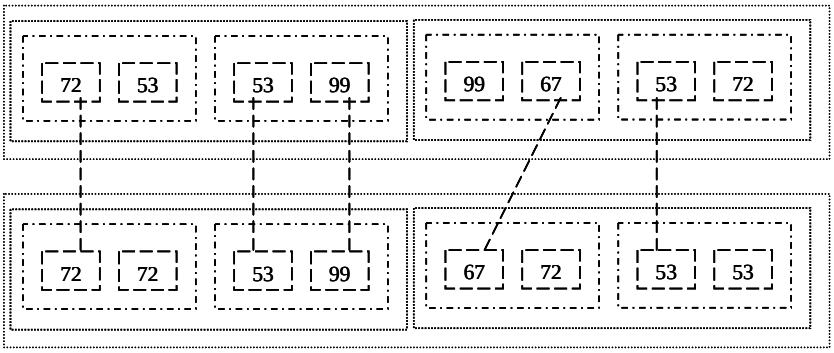


Fig. 7. Appariements identiques effectués par le PLSC et le CC.

Exemple 2 : Si nous prenons l'exemple de la figure 8, le taux de couverture découvert par le CC est nettement inférieur à celui du PLSC.

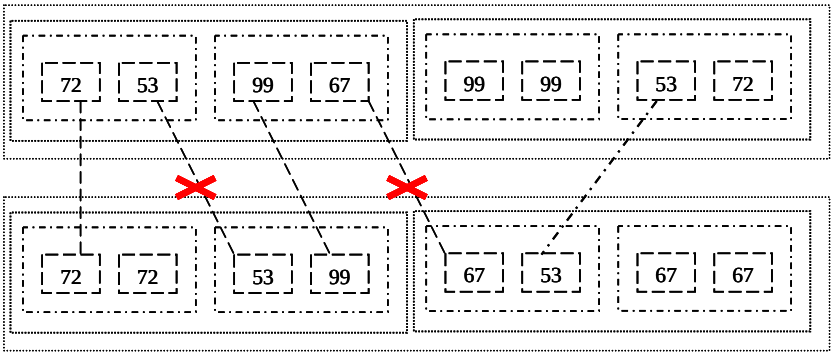


Fig. 8. Appariements différents entre le PLSC et le CC (ce dernier ne parvient pas à effectuer les appariements marqués d'une croix).

2.3.7.4 Comparaison expérimentale de complexité et de performance

La comparaison des deux algorithmes a été réalisée de la manière suivante. Pour un couple de séquences donné, leur taux de couverture calculé avec l'algorithme CC a été comparé à la valeur produite par l'algorithme PLSC.

Théoriquement les deux algorithmes doivent produire la même estimation. En pratique l'algorithme CC est une estimation inférieure de la PLSC qui est un algorithme parfait en terme de précision. L'erreur de précision du CC par rapport à la PLSC est alors calculée sous forme de pourcentage. En parallèle, pour comparer la rapidité des deux algorithmes, le nombre d'opérations nécessaires par chaque algorithme a été évalué.

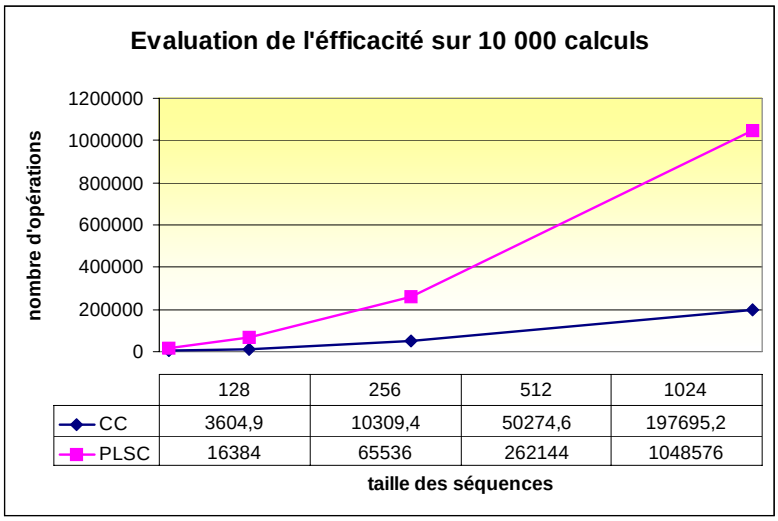


Fig. 9. Comparaison du PLCS et du CC en nombre moyen d'opérations.

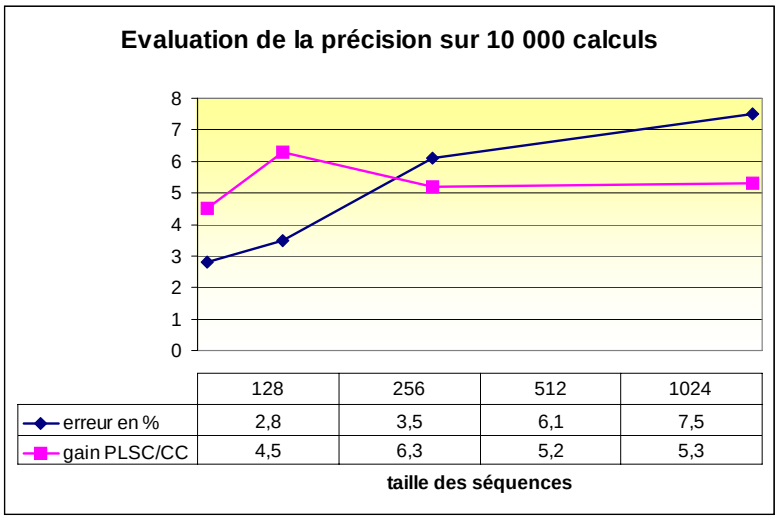


Fig. 10. Estimation de l'erreur et du gain (en nombre d'opérations) du CC par rapport à PLSC.

Cette expérience a été répétée 10 000 fois pour quatre longueurs différentes. Les couples de séquences qui ont servi comme base d'évaluation ont été extraites aléatoirement à partir des vecteurs de caractéristiques audiovisuelles (11 caractéristiques) présents dans notre base de données. Ces caractéristiques de natures variées ont été elles mêmes extraites à partir de différents documents vidéo de genres variés (journaux télévisés, jeux télévisés, films publicitaires…). Cette diversification a pour but de servir de base solide pour l'évaluation de la comparaison. Les moyennes de ces deux paramètres - nombre d'opérations nécessaires et pourcentage d'erreur – pour chacune des quatre longueurs étudiées sont représentées dans les deux figures 9 et 10.

Nous observons par exemple, dans la figure 9 que, pour des couples de séquences de taille 256, le CC effectue en moyenne 10309 opérations pour estimer leur ressemblance, tandis que la PLSC a besoin de 65536 opérations.

En ce qui concerne la précision, nous observons dans la figure 10 que, sur une séquence de 128 valeurs, 128 x 2.79 / 100 = environ 4 valeurs ne sont pas appariées par le CC alors qu'elles le sont par le PLSC.

Le but de cette comparaison étant l'estimation de l'erreur moyenne produite par le CC sur les données que nous avons à traiter pour juger de l'utilité de ce dernier, nous pouvons en conclure que le taux d'erreur est croissant avec la taille, mais à tendance logarithmique. Quant au gain en opérations, il est considérable. L'écart entre le taux de référence calculé par la PLSC et le taux approximé calculé par le CC reste faible face aux variations des valeurs que les mesures des caractéristiques peuvent produire sur un même contenu.

2.3.8 Comparaison des séquences par morceaux

Cette idée, comparer les séquences par morceaux au lieu de les comparer élément par élément, a été adaptée par certains chercheurs, [Horowitz 74, Keogh 97], et a pour arguments essentiellement les deux points suivants :

- dans deux séries chronologiques correspondant à des mesures physiques sur plusieurs occurrences d'un même phénomène, il est rare d'obtenir exactement les mêmes valeurs. On observe plutôt des valeurs évoluant dans un même intervalle.

- dans la plupart des domaines d'application, un élément isolé dans une série n'a pas d'importance effective. Seuls des segments d'une taille minimale donnée auront un intérêt. Dans ce cas, il n'est pas utile de perdre le temps nécessaire pour atteindre une comparaison à l'échelle de l'unité.

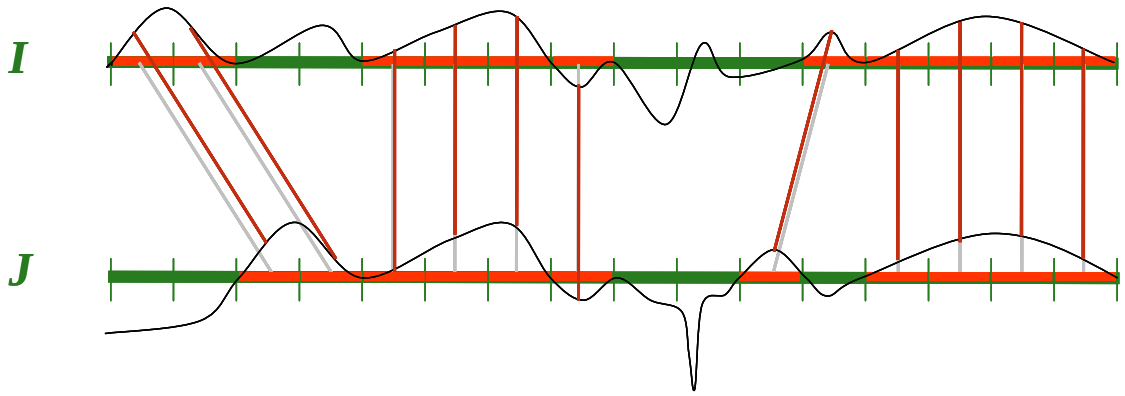


Fig. 11. Exemple de calcul du taux de couverture par morceaux pour deux séquences *I* et *J*.

La figure 11 montre un exemple du calcul du taux de couverture par morceaux. Partant de ces considérations, nous choisissons de comparer les vecteurs de caractéristiques, c'est-à-dire, les séquences, en appariant au mieux leur enveloppe morphologique construite par morceau au lieu de les apparier élément par élément.

2.4 Calcul des enveloppes morphologiques de séries chronologiques

2.4.1 Définitions de morpho mathématiques

Le traitement non linéaire basé sur la morphologie mathématique (MM) se fonde sur un modèle géométrique du signal. Les MM a été présentée et développée en grande partie par George Matheron [Matheron 75] et Jean Serra [Serra 84] à partir de la théorie des ensembles, et a été prolongée plus tard pour le traitement des images en niveau de gris. De nos jours, les MM sont largement répandues dans beaucoup d'applications, fournissant un ensemble d'outils puissants pour le traitement non linéaire des signaux [Serra 88, Haralik 87, Schmitt 94].

Pendant que des opérateurs linéaires sont basés sur le modèle linéaire de superposition, les opérateurs de MM dérivent d'un modèle non linéaire de superposition. La structure mathématique qui donne l'appui au modèle non linéaire de superposition est le treillis complet. Un treillis complet est un ensemble L tel que :

• L est équipé d'une relation d'ordre partielle $\leq$, définie par les propriétés suivantes :

$$\begin{aligned} &\forall x, y, z \in L \\ &(1)\; x \leq x \\ &(2)\; x \leq y \text{ et } y \leq x \Rightarrow x = y \\ &(3)\; x \leq y \text{ et } y \leq z \Rightarrow x \leq z \end{aligned} \tag{7}$$

• Pour chaque ensemble d'éléments {x[n]}, il existe en L un infimum, $\vee\{x[n]\}$, et un supremum, $\wedge\{x[n]\}$.

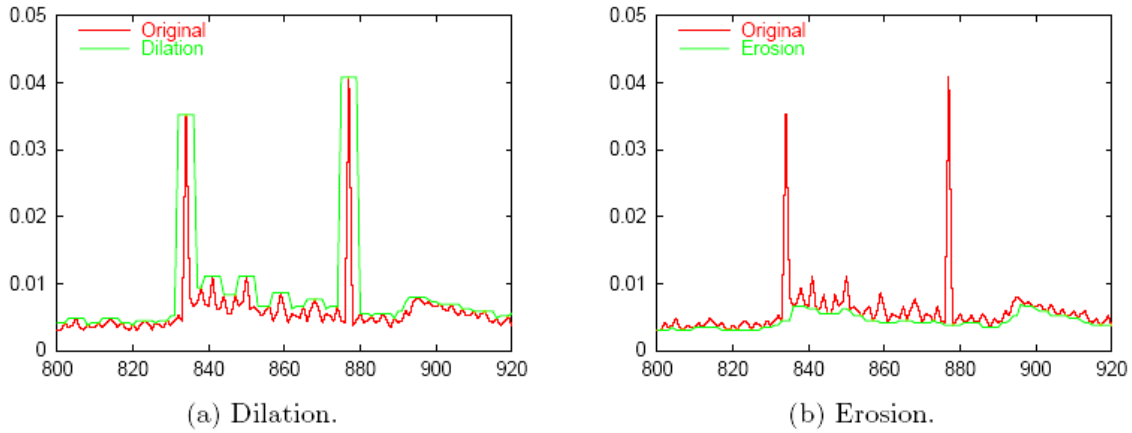
Un exemple de treillis est l'ensemble des réels IR, où l'ordre normal fournit une relation d'ordre complet, indiqué par les opérateurs de supremum et d'infimum.

Formellement, le supremum et l'infimum de deux signaux discrets x[n] et y[n] sont calculés comme suit :

$$\begin{aligned} z = x \vee y \Rightarrow z[n] &= \max\{x[n], y[n]\}, \forall n \\ z = x \wedge y \Rightarrow z[n] &= \min\{x[n], y[n]\}, \forall n \end{aligned} \tag{8}$$

2.4.2 Deux opérateurs morphologiques : la dilatation et l'érosion

La dilatation et l'érosion sont les opérateurs de base de la morphologie mathématique (MM). Comme représenté sur la figure 12, avec des paramètres classiques, la dilatation « étale » les maximums du signal tandis que l'érosion « étale » les minimums.



(a) Dilation. (b) Erosion.

Fig. 12. Exemples de dilatation et d'érosion avec un structurant plat horizontal de taille 2.

Formellement, ces opérateurs peuvent être définis ainsi :

Définition. *Dilatation* : la dilatation de x[n] par l'élément structurant b est définie par :

$$\delta_b\{x[n]\} = x[n] \oplus b[n] = \bigvee_{k=-\infty}^{\infty} x[k] + b[n-k] \tag{9}$$

Définition. *Erosion* : l'érosion de x[n] par l'élément structurant b est définie par :

$$\delta_b\{x[n]\} = x[n] (-) b[n] = \bigwedge_{k=-\infty}^{\infty} x[k] - b[n-k] \tag{10}$$

L'érosion rétrécit les pics et les lignes de crête. Les pics plus étroits que l'élément structurant disparaissent. Parallèlement elle élargit les vallées et les minima. La dilatation produit les effets inverses.

2.4.3 La construction de l'enveloppe morphologique

Avec les deux opérateurs classiques de la MM, l'enveloppe morphologique est construite par glissement du structurant tout au long de la séquence, comme c'est le cas de la figure 13.

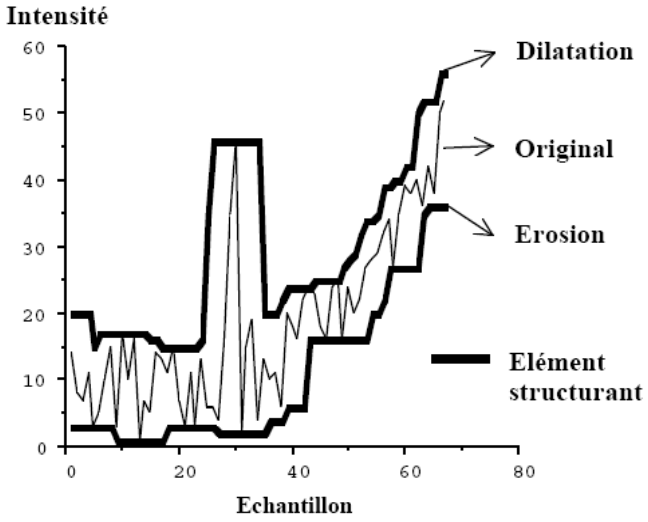


Fig. 13. Exemple d'enveloppe morphologique avec un structurant plan horizontal.

Notons que dans notre algorithme, cette enveloppe est construite par morceaux, et donc le structurant ne glisse pas tout au long de la série. L'érosion et la dilatation se calculent selon des morceaux de la taille du structurant utilisé. Ceci participe au principe d'économie du temps de calcul et de la mise forme récursive de la fonction.

Pour cela, la taille du structurant plan est d'une importance primordiale et influence fortement le calcul du taux de couverture.

2.5 Algorithme de comparaison pour une caractéristique audiovisuelle

Dans cette section, nous adaptons la méthode de comparaison générale que nous avons présenté dans la section 2 dans le but de l'appliquer sur des caractéristiques audiovisuelles. Pour ce faire, il convient d'adapter certains paramètres, les bornes de taille des séquences communes recherchées *tMax* et *tMin*, et la taille du structurant *strt* qui va servir dans la comparaison des enveloppes des séquences lors du calcul de leur couverture.

Tout cela étant en rapport direct avec les caractéristiques audiovisuelles vues comme des séries temporelles, nous commençons par une étude de la nature de ces caractéristiques, et plus précisément de leurs courbes.

2.5.1 Typologie des courbes

Dans le but d'optimiser la comparaison des deux séquences *I* et *J*, du point de vue de la rapidité de calcul, de la qualité des résultats de comparaison et de la gestion du bruit et des décalages, nous avons mené une étude visant à classifier les différentes natures des caracté-

ristiques audiovisuelles. L'étude se base sur le coefficient de variation en raison de l'information qu'il fournit à propos de l'allure de la courbe. Nous définissons le coefficient de variation ainsi :

Définition. *Coefficient de variation.* Le coefficient de variation V d'une séquence X est le rapport entre l'écart type de X et la valeur moyenne de X. Il est parfois multiplié par cent pour exprimer un pourcentage. Si σ_X est l'écart type X et μ sa moyenne, alors $V = \sigma_X / \mu$. Le coefficient de variation indique alors le taux de mouvement absolu de la séquence. [Weisstein 99]

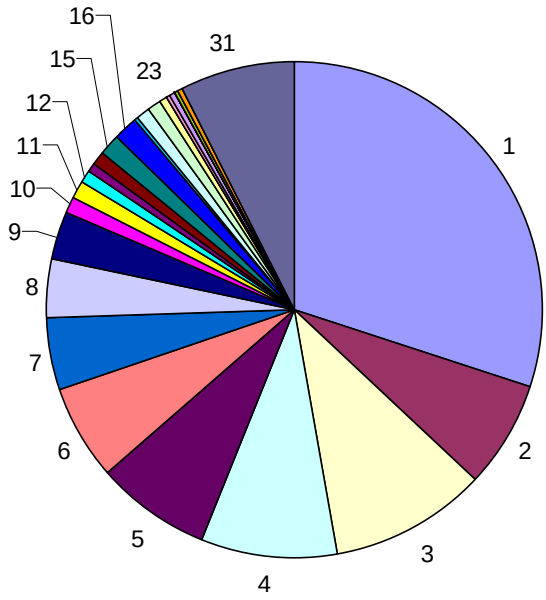


Fig. 14. Distribution des fréquences des coefficients de variation compris entre 0 et 3 (numérotés de 1 à 30) et supérieure à 3 (31).

Une courbe représente un vecteur de valeurs d'une caractéristique. Nous voulons distinguer des courbes peu mouvementées, voire lisses, d'une part, de courbes très perturbées comprenant de fortes oscillations, ou de courbes à mouvement moyen comportant des changement importants à des instants précis, d'autre part.

Le principe de notre méthode ne requiert pas de prétraitement des courbes à analyser, comme par exemple une segmentation en plans ou en unités narratives données (scènes ou autre pouvant garantir une certaine homogénéité des données à comparer.). Donc il s'agit de déterminer automatiquement la nature de la courbe pour adapter les paramètres assurant le bon fonctionnement de l'algorithme de comparaison. L'adaptation doit être instantanée, au moment de l'exécution, puisqu'il n'y a pas d'apprentissage ou d'adaptation aux genres des documents ou aux caractéristiques.

Les caractéristiques que nous utilisons étant toujours positives, une première expérience exécutée sur un ensemble de 1000 séquences de taille et nature aléatoires, nous a permis de déduire que le coefficient de variation prend des valeurs qui dépassent rarement le seuil de 3 (seules 8% des séquences ont des valeurs supérieures à 3). La figure 14 illustre les résultats de cette expérience.

Nous avons pu classer expérimentalement les courbes issues des caractéristiques audiovisuelles selon 3 classes :

- Classe A : $0 \leq V < 0.29$: courbe lisse
- Classe B : $0.29 \leq V < 0.65$: courbe mouvementée
- Classe C : $0.65 \leq V$: courbe avec fortes oscillations

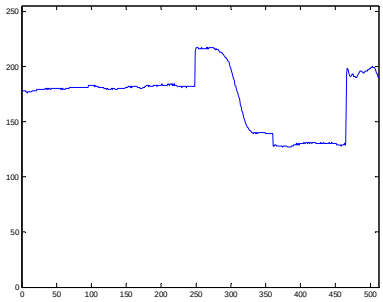
Le nombre de classes n'étant pas important ni les limites exactes. Nous avons juste voulu ne pas mettre toutes les caractéristiques dans le même lot n'étant pas de la même nature. Le but recherché est l'économie du calcul en dépit des opérations inutiles. De plus le structurant devrait couper les séquences en sous séquences aussi de longueur puissance de deux. Une étude plus exhaustive aurait pu être faite sur les typologies de caractéristiques mais nous ne pouvons voir l'intérêt. Un exemple d'une courbe de classe A est donné par la figure 15 (a). Ce sont souvent les caractéristiques concernant une information globale ou moyenne variant très doucement tout au long d'un document. Dans la figure, il s'agit de la luminance moyenne des pixels de l'image.

La classe B est composée en général des caractéristiques plus mouvementées en raison de leur sensibilité au changement du contenu, mais aussi aux bruits. Les séquences issues de telles caractéristiques doivent être comparées en utilisant un structurant plus petit relativement aux séquences issues de la classe A, pour pouvoir suivre de près leur changement. Un exemple est donné dans la figure 15 (b). Une autre caractéristique, figure 15 (c), dont les séquences sont souvent classées dans B, est le contraste couleur de l'image, qui est défini comme suit :

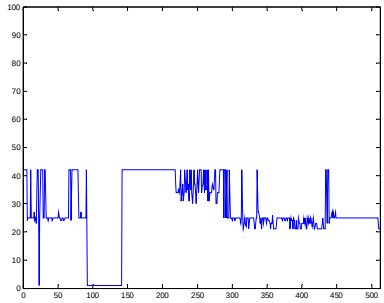
Définition. *Contraste couleur de l'image.* Le contraste couleur de l'image est l'écart entre les deux couleurs les plus fréquentes et les plus différentes de l'image.

$$contraste = |L_1 - L_2| + \text{distcirc}(H_1, H_2) \times \log\left(\frac{S_1 + S_2}{2}\right) \quad (\mathbf{11})$$

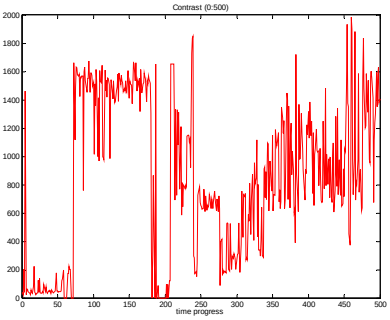
Où L_1, L_2 sont respectivement les luminances des deux couleurs dominantes, H_1,H_2 leurs teintes, et S_1, S_2 leurs saturations, et distcirc(x, y) est la distance circulaire entre x et y.



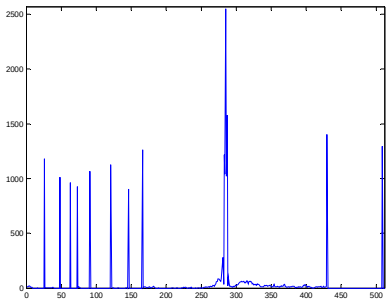
(a) classe A : luminance moyenne



(b) classe B : saturation dominante.



(c) Segment extrait de la caractéristique contraste



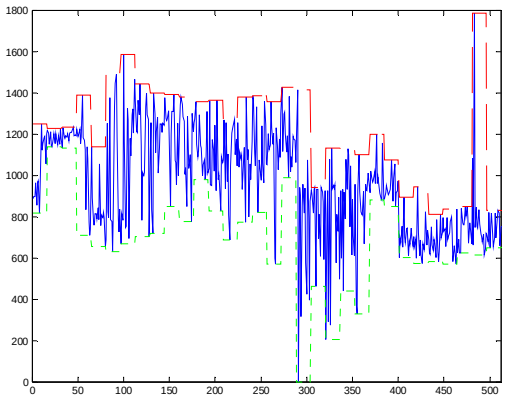
(d) classe C : quantité de mouvement

Fig. 15. Exemples de séquences appartenant aux trois classes A, B et C.

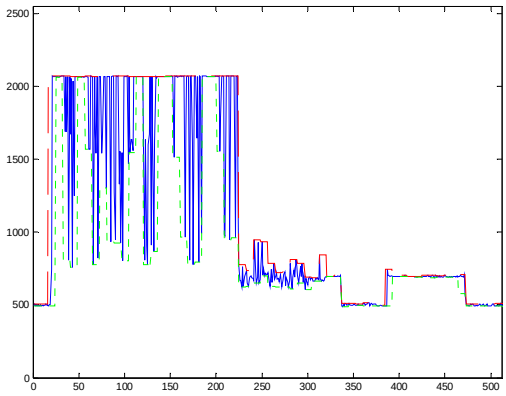
La classe C, celle des séquences ayant un coefficient de variation élevé, est composée essentiellement des caractéristiques comportant des pics relativement importants par rapport au comportement général. Une caractéristique typique de cette classe est la quantité de mouvement, estimée par la variation moyenne des intensités hors changements de plans dans le contenu vidéo (cf. Figure 15 (d)).

Il est important de noter qu'une même série temporelle, extraite pour décrire une caractéristique donnée, peut avoir des comportements différents tout au long de l'axe de temps. Elle peut comporter des séquences pouvant appartenir aux différentes classes. Ceci implique que la détermination de la taille du structurant morphologique doit être faite à chaque calcul de la couverture pour deux séquences données et non pas en fonction de la caractéristique en cours de comparaison.

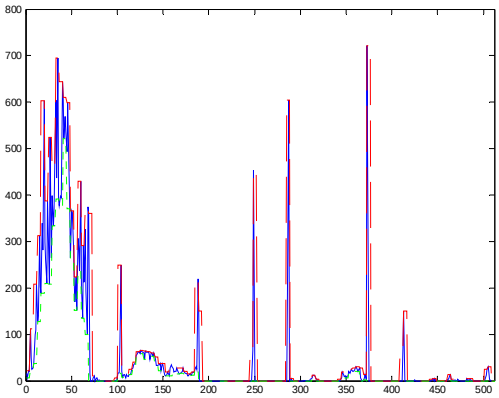
2.5.2 Comparaison par intersection d'enveloppes



classe A



classe B



classe C

Fig. 16. Exemples d'enveloppes morphologiques.

Comme nous l'avons déjà proposé, le but est de comparer la morphologie des courbes représentant les séquences et non pas les éléments isolés de ces séquences. Nous construi-

sons l'enveloppe à l'aide de l'érosion et de la dilatation, et nous vérifions l'appariement de ces enveloppes morceau par morceau. L'algorithme CC qui calcule la couverture, identifie en fait les portions sur lesquelles les deux enveloppes des séries comparées ont une intersection non-nulle. La figure 16 montre les enveloppes morphologiques pour trois séquences de caractéristiques appartenant chacune à une des trois classes : A, B et C.

2.5.3 Adaptation de l'algorithme CC

Nous modifions le calcul de la couverture en y échangeant l'unité de découpage par un segment de taille *strt*. La formule devient alors :

$$couv_{strt}(I,J) = \begin{cases} 0 & \text{si } [e_{strt}(I), d_{strt}(I)] \cap [e_{strt}(J), d_{strt}(J)] = \phi \\ 100 & \text{si } [e_{strt}(I), d_{strt}(I)] \cap [e_{strt}(J), d_{strt}(J)] \neq \phi \text{ et } t = strt \\ \frac{1}{2}\max \begin{pmatrix} couv_{strt}(I_1, J_1) + couv(I_2, J_2), \\ couv_{strt}(I_1, J_2), \\ couv_{strt}(I_2, J_1) \end{pmatrix} & \text{si } [e_{strt}(I), d_{strt}(I)] \cap [e_{strt}(J), d_{strt}(J)] \neq \phi \text{ et } t > strt \end{cases} \quad (\mathbf{12})$$

2.5.4 Le structurant morpho mathématique

Pour chaque couple de séquences *(I,J)* dont le taux de couverture est à estimer, la taille d'un élément structurant *strt* est pré-calculée. Puis $couv_{strt}(I,J)$ est évalué comme étant le pourcentage des sous séquences ordonnées couplées par les opérations morphologiques.

L'exactitude du taux de couverture dépend directement du choix de l'élément structurant. L'élément structurant est plan, il correspond en effet à une durée de temps qui doit être déterminée. Une enveloppe morphologique représentative doit être assez flexible afin de permettre un appariement imprécis, mais pas trop flexible pour éviter d'associer des séquences incomparables. En particulier, pour des séquences lisses, l'élément structurant adapté pourrait être relativement long par rapport à celui qui pourrait convenir au traitement de séquences ayant un coefficient de variation élevé.

Il doit être également proportionnel à la longueur de la séquence. Nous proposons d'initialiser sa taille ainsi:

$$strt(I,J) = t / \delta_{max(V(I),V(J))} \quad (\mathbf{13})$$

Où :

- $0 \leq V < 0.29 \Rightarrow \delta = 32$,
- $0.29 \leq V < 0.65 \Rightarrow \delta = 64$, et
- $0.65 \leq V \Rightarrow \delta = 128$.

2.5.5 Entre l' «efficacité » et la « précision »

Ces deux paramètres sont les paramètres clé de la méthode de comparaison que nous proposons. Ils jouent le rôle de bascules entre le mode ***« efficacité »*** (taille des séquences en dehors de l'intervalle [*tMin*, *tMax*]). et le mode ***« précision »*** (taille des séquences à l'intérieur de l'intervalle [*tMin*, *tMax*]).

2.5.5.1 Le paramètre tMax

Dans le mode *efficacité*, l'algorithme IQR est appliqué. En conséquence, le critère de comparaison est simple et le filtrage est ultra rapide. Arrivant à des séquences de tailles *tMax*, l'algorithme général DiSC bascule du IQR vers l'algorithme plus fin qu'est le ESSV. Le ESSV utilise à son tour un critère de comparaison : le Calcul de Couverture (CC). Or ce critère demande plus de temps pour la comparaison. En échange, il fournit avec précision le degré de ressemblance des séquences comparées.

Donc, ce second mode qui commence au niveau des séquences de taille *tMax* est un mode de précision dit aussi en anglais *effectivity* puisque l'on mesure la ressemblance effective des séquences.

La taille de ce paramètre joue alors le rôle de modérateur entre précision et rapidité d'exécution. Elle dépend de la taille globale des documents comparés, et de la taille approximative des segments recherchés. Bien qu'il limite la borne supérieure des séquences cherchées, *tMax* n'impose pas une taille fixe. Les couples des séquences similaires peuvent avoir des tailles variables comprises entre les deux bornes ; *tMax* maximale et *tMin* minimale.

2.5.5.2 Le paramètre tMin

Le mode *précision* commence avec les séquences de taille *tMax*. Or, au-dessous d'un certain seuil, la comparaison des séquences devient insignifiante :

- pour des séquences de valeurs trop petites, on peut trouver un nombre indéfini de séquences semblables mais cette similarité n'est plus significative.

- en principe, et selon le contexte de l'application, les séquences vidéo semblables nous intéressent jusqu'à une certaine durée. Au dessous de cette durée, même si la ressemblance est significative, elle n'apporte pas ou peu d'informations au processus de comparaison envisagé.

Par cela, afin d'éviter des opérations inutiles et un temps de calcul prolongé le processus de comparaison s'arrête en arrivant à une taille *tMin* et la dimension de la matrice de similarité est définie sur la base de ce paramètre.

Toutes les ressemblances seront identifiées entre des séquences vidéo de taille multiple à ce temps minimal (*tMin)*. Notons aussi que *tMax* n'est pas à son tour le temps maximal des

séquences pouvant être semblables puisque deux séquences semblables successives de n'importe quelle taille peuvent se fusionner pour en former une troisième.

Des expériences faites sur une grande base de données vidéos montrent que les valeurs correspondantes à 0,5 seconde pour *tMin* et 30 secondes pour *tMax* convenaient pour des documents de longueur variant entre 3 minutes et une heure, permettant une détection correcte des invariants de production. Pour de plus longs documents ces paramètres sont réadaptés en fonction des buts de l'analyse (voit chapitre 4).

2.5.6 Comparaison du PLSC et du CC après adaptation des paramètres

Les figures 17 et 18 montre que l'écart entre les deux méthodes baisse après l'utilisation de la comparaison par morceaux et une taille de structurant choisie selon l'équation (13).

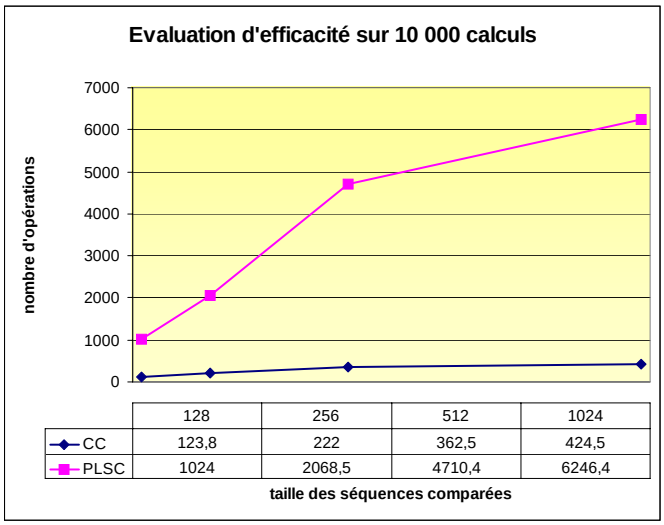


Fig. 17. Comparaison de PLCS et CC en nombres d'opérations après approximation par morceaux.

Par exemple pour les séquences de taille 1024, l'exécution de l'algorithme CC avec une taille de structurant adaptée a baissé l'erreur par rapport à l'algorithme PLSC de 7.5% (figure 10) à 5.3% (figure 18).

De l'autre coté le gain de calcul exprimé en nombre d'opérations par comparaison de deux séquences a augmenté. Par exemple, en comparant les tables 6 et 13, nous remarquons que, pour des séquences de taille 1024 nous passons d'un gain de 5,3 (le CC est en moyenne 5.3 fois plus rapide que le PLCS) à un gain de l'ordre de 14,7 soulignant l'importance de l'application l'algorithme CC dans notre contexte.

Dans le cas de l'analyse des contenus vidéo, il est très intéressant d'identifier des nuages de ressemblances puisque l'évolution des autres caractéristiques comparées en parallèle n'identifiera probablement pas exactement les mêmes limites de segments.

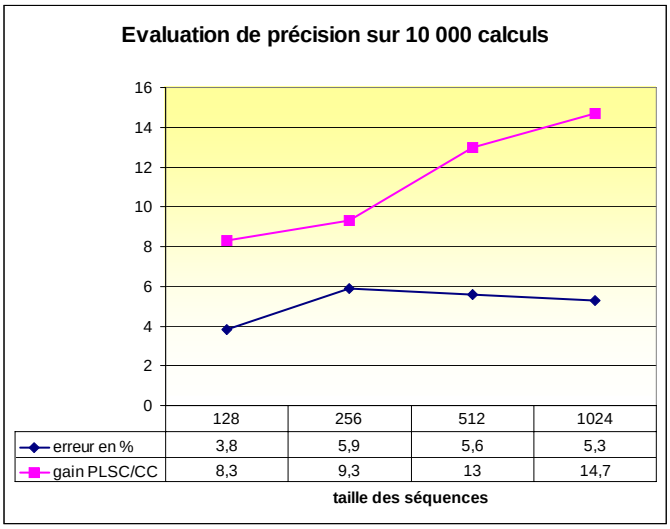
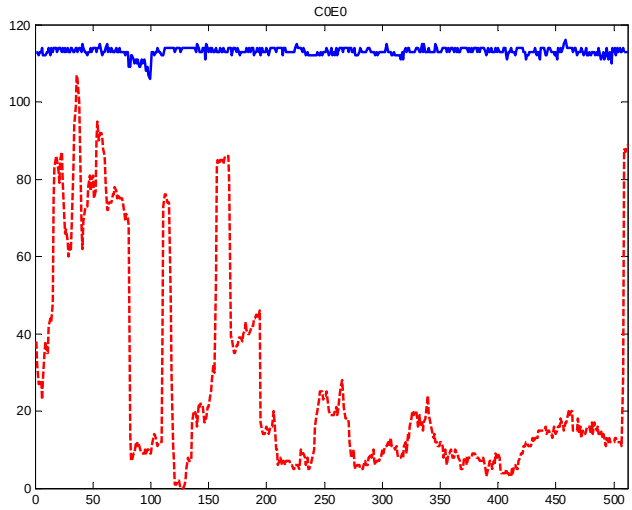


Fig. 18. Estimation de l'erreur et du gain (en nombre d'opérations) du CC par rapport à PLSC après approximation par morceaux.

2.5.7 Exemples et résultats

Dans la figure 19, nous montrons des exemples de couples de séquences : (i) éliminés par l'algorithme IQR, (ii) retenus comme candidats potentiels à la ressemblance mais ensuite éliminé par l'algorithme ESSV à cause d'un faible taux de couverture, et enfin (iii) retenus comme similaires à cause d'un taux de couverture élevé.



Eliminés
(couv = 0%)

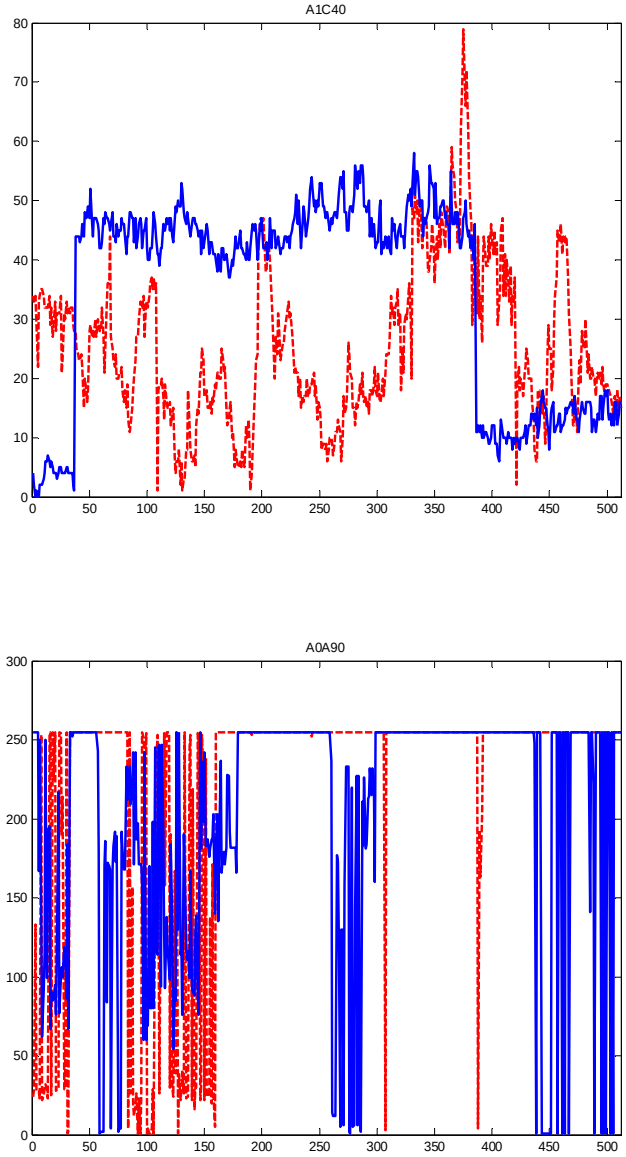


Fig. 19. Exemples d'enveloppes morphologiques.

2.6 Matrice de comparaison

Dans ce paragraphe, nous montrons comment nous mettons en évidence les séquences similaires trouvées et nous présentons un schéma des résultats globaux sous la forme d'une matrice.

2.6.1 Principe

Pour chaque couple de vecteurs (séries temporelles) représentant une caractéristique audio-visuelle, nous enregistrons les résultats de la comparaison dans une matrice. Nous effectuons ensuite une fusion inter matricielle pour extraire les points de ressemblances communs à tous ou pour le moins à une majorité des matrices. Ceci est dans le but d'identifier les éléments communs dans deux documents vidéo.

2.6.2 Construction

Pour une caractéristique donnée F_x, nous extrayons les séries temporelles la décrivant à partir des documents à comparer. Nous obtenons deux séries chronologiques sur lesquelles nous appliquons l'algorithme global DiSC.

Au moment même du déroulement de la comparaison, une matrice carrée est créée (figure 20). Les axes de la matrice sont proportionnels à la dimension temporelle de chaque séquence, ils sont de longueur *dim*, tel que *dim* × *tMin* est la taille de chaque séquence. Le pas de la matrice est alors *tMin*. Dans un premier temps, cette matrice est initialisée à zéro.

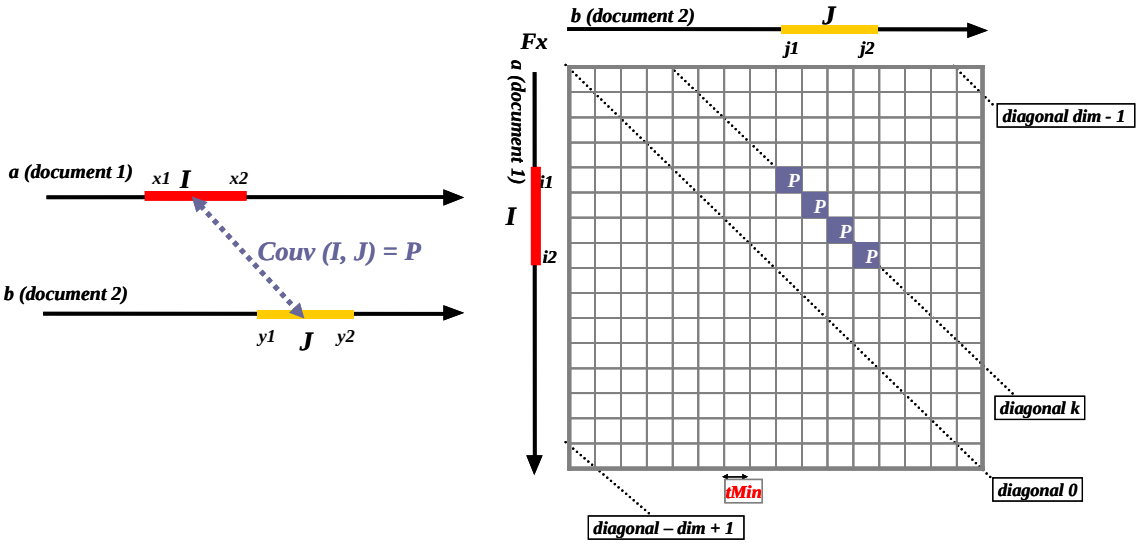


Fig. 20. Matrice de comparaison des séries a et b représentant la caractéristique Fx pour les deux documents (1 et 2).

Pour chaque couple de séquences (*I*, *J*) de taille comprise entre *tMin* et *tMax*, et de *taux de couverture P*, nous procédons comme suit.

Si *x1*, *x2* (resp. *y1*, *y2*) sont les indices des bornes des deux séquence : $I = [I_{x1}, I_{x2}]$, et $J = [J_{y1}, J_{y2}]$ nous avons *x2* - *x1* = *y2* - *y1* = *m* × *tMin*. Les *m* éléments de la diagonale, commençant par les coordonnées (*i1* = *x1* / *tMin*, *j1* = *y1* / *tMin*) et se terminant par (*i2* = x2 / *tMin*, *j2* = *y2* / *tMin*), reçoivent la valeur *P*, ***si et seulement si*** *P* est supérieure à l'ancienne valeur attribuée aux éléments (voir remarque 2 plus bas).

Dans la figure 20, nous donnons un exemple de deux séquences *I* et *J* de taux de couverture *P*. Les éléments ombrés de la matrice reçoivent la valeur *P*. Ils appartiennent à la diagonale *k*, - la diagonale zéro étant la principale -. Nous interprétons le fait que la séquence *J* est décalée par rapport à la séquence *I* de *k*×*tMin* unités de temps. *k* = (*j1* - *i1*). Ce qui veut dire que si on superpose *a* et *b,* le segment *J* sera décalé de (*k* x *tMin*) unités de temps par rapport au segment *I*. Cette interprétation sera discutée en détail dans le chapitre suivant.

2.6.3 Remarques

1. La matrice de comparaison est une représentation de tous les alignements possibles de n'importe quels couples de séquences, de n'importe quelle longueur, issus des deux séries chronologiques comparées, telle que leur longueur soit comprise entre *tMin* et *tMax*.

2. Le choix de garder dans la matrice le taux de couverture le plus élevé à chaque itération est le plus simple. Il respecte au mieux les zones des séquences correspondant à une certaine densité de ressemblance. Ce choix peut être remplacé par d'autres stratégies comme par exemple la moyenne pondérée.

2.6.4 Fusion inter-caractéristiques

Pour comparer deux documents, on procède à la création de la matrice de comparaison pour chacune des caractéristiques mises en jeu. Pour traiter du document en entier et non pas de chaque caractéristique séparément, on fusionne les différentes matrices pour obtenir une matrice résultante.

Une matrice dite d'« auto-similarité » a été utilisée par Foote et Cooper dans plusieurs travaux recents, citons [Foote 01]. Leur matrice résultante ressemble conceptuellement à notre matrice de comparaison, mais les auteurs se sont limités à la comparaison d'un document à lui-même, afin d'accomplir certaines tâches spécifiques, comme par exemple : la segementation, la génération des résumés et la classification. Un autre point qui nous distingue des travaux de Foote et Cooper, est le remplissage optimisé de la matrice, qui nous permet d'envisager le traitement de documents audiovisuels de grande dimension.

En effet, un segment vidéo invariable peut être caractérisé par un sous-ensemble de caractéristiques, qui peut différer selon le type de l'invariant. Pour cette raison, nous devons observer la fusion de toutes les matrices de comparaison calculées indépendamment, chacune pour une caractéristique différente. Des décisions doivent être prises au regard de la discrimination de ces caractéristiques en fonction de leur exactitude, leur efficacité estimée, le genre du document, et leur importance sémantique pour un post-traitement automatique d'interprétation.

Définition. *Fusion de données*. La fusion de données est définie comme étant le processus qui combine des informations provenant de plusieurs sources pour produire une information d'une qualité supérieure plus complète. [Wald 99].

Toujours dans le domaine de la fusion arithmétique, plusieurs choix doivent être effectués dans le but de fusionner les matrices provenant des caractéristiques audiovisuelles pour obtenir un schéma global représentant la comparaison des deux documents vidéo.

Ces choix se repartissent dans les points suivants selon que

- nous voulons ou pas pondérer les caractéristiques en fonction de
    - leur fiabilité,
    - leur influence sur les types précis des documents, ou même
    - leur importance en général.
- nous voulons intégrer
    - systématiquement toutes les caractéristiques en jeux, ou bien
    - seulement un sous-ensemble de caractéristiques que nous jugeons adapté pour comparer un type donné de documents

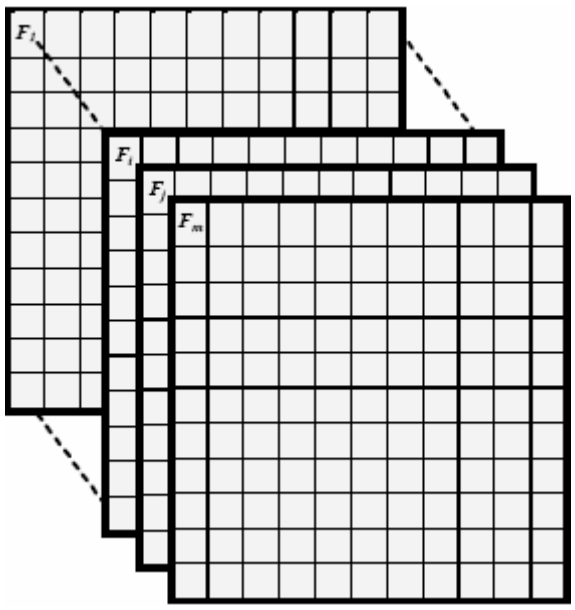


Fig. 21. Fusion inter caractéristique.

Parmi les méthodes de fusion les plus simples, nous pouvons formuler la fusion comme une somme pondérée ou comme un produit pondéré de la façon suivante :

– produit (pondéré ou pas),

$$c_{ij} = \prod_f c_{ij_f}{}^{\alpha_f} \tag{14}$$

– somme (pondérée ou pas),

$$c_{ij} = \sum_f \left(\alpha_f \times c_{ij_f} \right) \tag{15}$$

D'autres propositions de fusion simples existent. Nous citons la fusion par le choix du maximum, le minimum ou bien le médian. Toutes ces fusions, et d'autres de type plus compliquées, comme la fusion possibiliste, la fusion probabiliste, et la fusion par coefficients de confiance, peuvent intervenir selon les critères et les priorités recherchés par la comparaison.

2.6.5 Exemples et résultats

Dans l'exemple illustré par les figures 22 et 23, nous pouvons observer une matrice de comparaison résultante de la comparaison de deux documents de journaux télévisés de la chaîne CNN, la fusion est faite en utilisant une multiplication non pondérée, *tMin*=32, tMax=1024.

Un deuxième exemple, la figure 24 illustre une matrice de comparaison typique résultant de la comparaison de deux plages publicitaires, chacune de durée 3 min. environ (3 × 60sec. × 25 images × 11 carac.=49.500 valeurs à comparer).

Dans cet exemple, les onze caractéristiques utilisées étaient :

– le taux d'activité,
– les deux teintes dominantes,
– les deux saturations,
– les deux luminances dominantes,
– la luminance moyenne,
– la granularités horizontale et verticale de l'image, et
– le contraste.

Dans la figure 24, les diagonales foncées (valeurs élevées dans la matrice) indiquent quatre couples de films publicitaires semblables de différentes longueurs et positions dans le temps des enregistrements. Le premier enregistrement figure verticalement sur la matrice, alors que le deuxième est représenté par la dimension horizontale.

En projetant les blocs contenant ces quatre diagonales, nous trouvons les occurrences correspondantes des éléments semblables dans chaque document. Les blocs foncés identifient les films publicitaires qui sont plutôt semblables tandis que les lignes et les colonnes claires identifient des films publicitaires qui sont très différents des autres.

Il est intéressant de mentionner que les valeurs sur la matrice, identifiées par une ellipse en traits, résultent de la détection de deux films publicitaires différents pour un même produit. Ceci souligne le fait que le producteur de ces films publicitaires a suivi les mêmes ensembles de directives de réalisation et que ces directives sont indirectement capturées par les caractéristiques choisies. La matrice de comparaison carrée a été coupée pour éliminer les valeurs ajoutées afin d'obtenir des séquences de longueur en puissance de 2.

La comparaison, effectuée sur un Pentium 4, 2.6GHz, 512MO sans parallélisation du code, dure moins d'une minute. Après parallélisation exploitant l'indépendance des appels quadratiques récursifs, d'une part et l'indépendance inter caractéristiques d'autre part, la durée de la comparaison est réduite de 10 fois. La parallélisation est détaillée dans le chapitre 4. Des exemples supplementaires de matrices de comparaison sont montrés dans l'annexe A. Ces matrices sont issues de la comparaison de plages publicitaires.

2.7 Conclusion

Nous avons proposé une méthode pour la comparaison des documents vidéo. Cette méthode est basée sur une approche hiérarchique de comparaison en utilisant un filtrage morphologique. Etant donné deux documents vidéo, nous avons employé des techniques rapides de recherche capables d'extraire toutes les séquences semblables, de différentes longueurs, et ceci pour n'importe quelle caractéristique extraite du contenu.

La méthode de comparaison présentée dans ce chapitre a l'avantage de traiter des documents de différents genres et de différentes longueurs puisque nous n'avons pas imposé des hypothèses concernant ces deux caractéristiques. Elle trouve le meilleur alignement temporel entre deux documents afin de maximiser la similarité. La mesure de similarité, qui dérive de cette méthode est présentée dans le chapitre suivant.

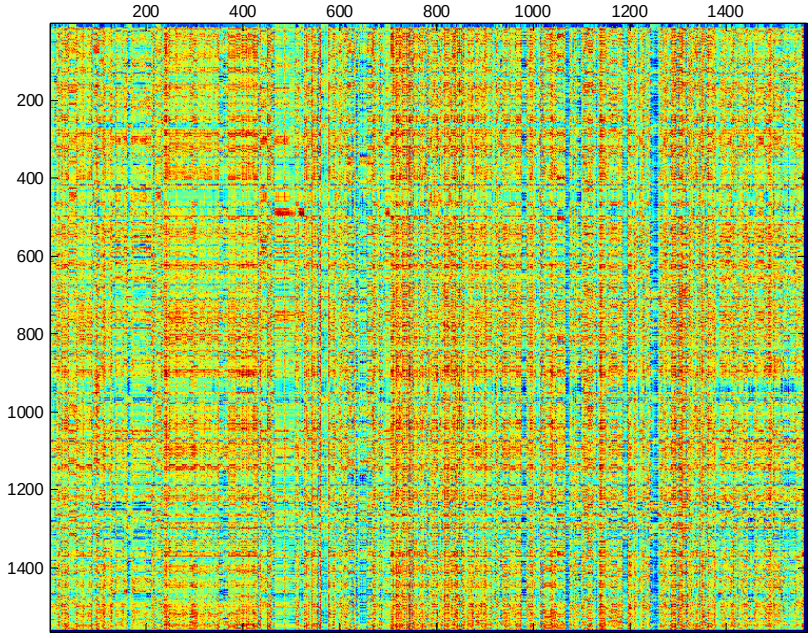


Fig. 22. La matrice de comparaison résultant de la fusion de onze caractéristiques audiovisuelles pour deux documents d'une même collection.

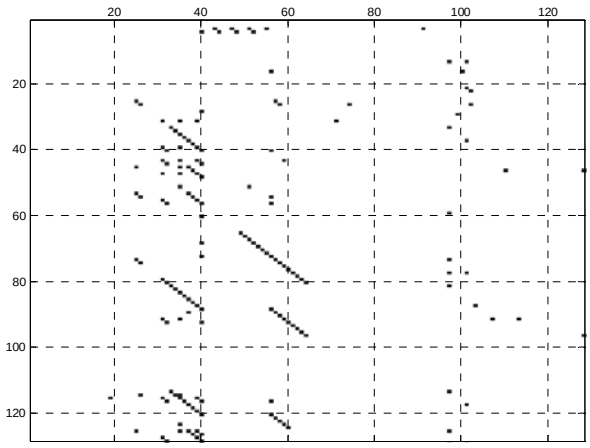


Fig. 23. Un zoom sur la matrice de l'image précédente après un filtrage par un seuil ne retenant que les fortes valeurs. Nous remarquons les segments invariants en direction diagonale.

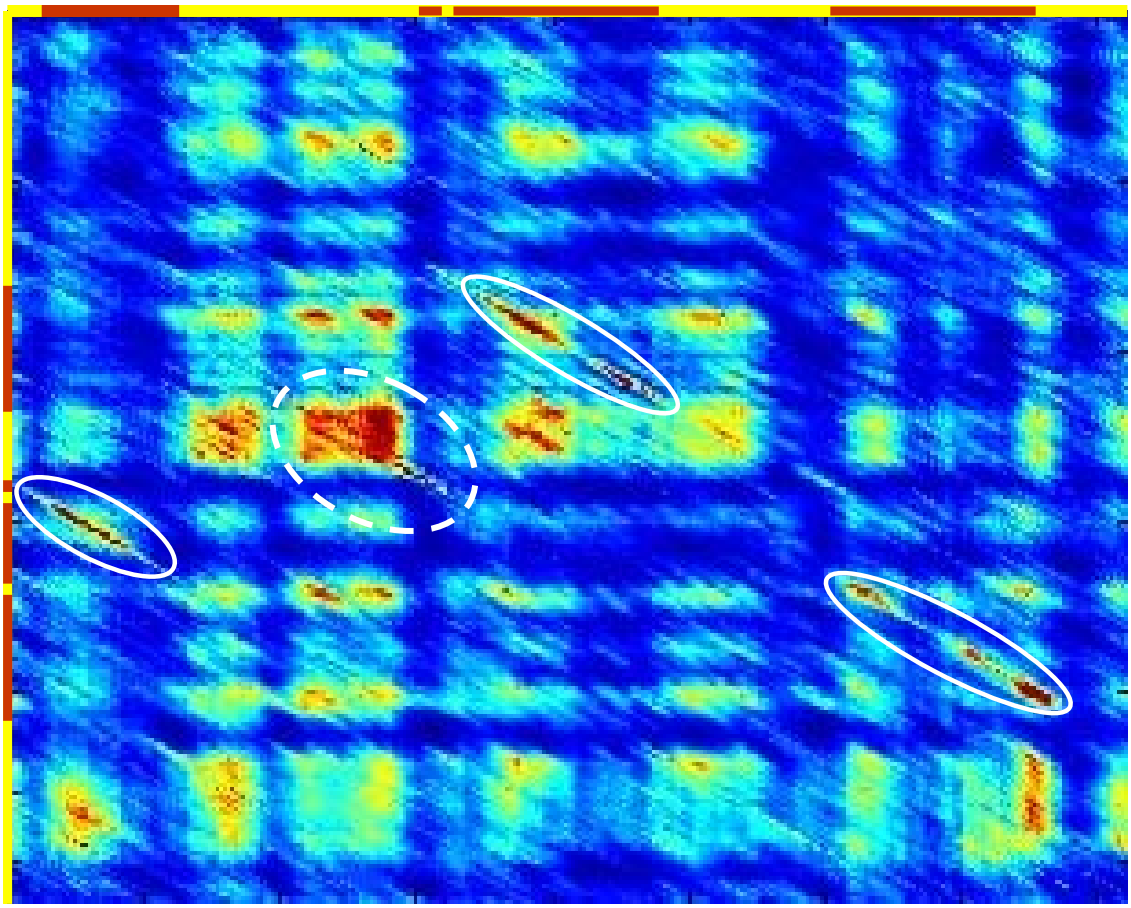


Fig. 24. La matrice de comparaison de deux plages publicitaires.

3 Chapitre 3 : Mesure de similarité

Chapitre 3

MESURE DE SIMILARITE

3.1 Introduction

La mesure de similarité est un outil utile, voire indispensable, pour la gestion automatique ou semi-automatique des documents vidéo numériques. En effet, l'évolution rapide de la technologie a permis au grand public de s'équiper de matériels multimédias sophistiqués. Une grande majorité de foyers disposeront, d'ici quelques années, de lecteurs enregistreurs numériques associés à des disques durs de capacité conséquente, destinés à conserver sur le plus ou moins long terme des programmes télévisés. Différents problèmes seront susceptibles de se présenter :

- Les programmes préférés seront noyés au milieu des flux enregistrés.
- Il y aura plusieurs genres de programmes enregistrés, correspondants aux différents centres d'intérêt d'une ou de plusieurs personnes.

A ces deux problèmes s'ajoute celui, classique, de l'encombrement progressif du disque qui devra conduire au développement de stratégies de nettoyage ou de rangement automatique des enregistrements. Pour aussi classique qu'il soit, ce problème reste insurmonté pour les documents audiovisuels. Comment les indexer et les classer pour les sélectionner en vue de leur suppression ou de leur sélection ultérieure.

Nous ne proposerons pas ici un système complet répondant à ce problème, mais nous allons nous intéresser à un élément clé pour ces outils en devenir : la définition d'une mesure de similarité qui permettra de regrouper, classer ou dissocier les enregistrements.

Être capable de dire que deux documents se ressemblent beaucoup, est une affirmation qui n'est pas uniquement subjective. Nous formulons l'hypothèse que cette affirmation peut être argumentée par la mise en évidence d'éléments communs.

Dans ce chapitre, nous menons une réflexion autour de la ressemblance des documents vidéo entre eux, en formulant plusieurs hypothèses sur la nature même d'une similarité entre deux enregistrements. La mesure que nous proposons est basée sur la quantification des éléments communs détectés grâce aux outils présentés dans le chapitre 2, et sur l'ordre temporel dans lequel ils apparaissent dans chaque document. Nous disposons pour cela d'un schéma de l'évolution temporelle et des éléments communs : la matrice de comparaison.

3.2 La similarité entre deux documents vidéo

3.2.1 Une autre définition pour un document vidéo

Une définition simple d'un document vidéo est « une suite d'images combinée à une succession d'éléments sonores organisés dans le temps. Si nous voulons donner plus d'importance à la connexité du contenu, nous pouvons le définir informellement comme une stricte série d'événements audiovisuels.

3.2.2 Définition de l'événement

Du point de vue sémantique, un événement, comme nous l'entendons dans ce chapitre, n'est pas uniquement un fait qui se déroule sur l'écran, comme par exemple une explosion, le coucher du Soleil, ou une course de chevaux. Il peut être aussi, simplement, un comportement non perceptible d'un sous ensemble de caractéristiques audiovisuelles mises en jeu, comme par exemple une image devenant de plus en plus claire en terme de luminosité accompagnée d'un son dont la fréquence est de plus en plus aiguë.

Un « événement » audiovisuel peut être lié à une suite d'images, une unité sonore, un plan, une scène ou toute autre unité narrative réalisée séparément. Mais ce pourra être un objet plus complexe. Par exemple, pendant une période donnée, on pourra voir associée une même publicité pour des magasins d'électroménager, en préambule du bulletin météo. Ces deux films vidéo (publicité et bulletin météo) n'appartiennent pas à une même unité matérielle – ils sont réalisés séparément - mais ils participent tout de même à un même événement audiovisuel.

3.2.3 Définition de l'espace des événements

Soit $C = \{c_1, c_2, \ldots, c_m\}$ l'ensemble des caractéristiques audiovisuelles dont on dispose pour notre comparaison et Φ la fonction de fusion qui a été appliquée pour obtenir la matrice de comparaison des deux documents vidéo V_1 et V_2.

Soit E, dans ce cas, l'espace des événements correspondant à la matrice obtenue par la fusion Φ. C'est alors l'ensemble des segments dont la taille est multiple d'une taille unitaire *tMin,* formés par la combinaison des caractéristiques de C.

Chaque couple de vidéos est représenté par sa matrice de comparaison avec sa projection dans E. La projection des deux documents dans E est donnée par l'ensemble fini :

$$Proj\,(V_1/E) = Proj\,(V_2/E) = \{e_{11}, e_{12}, \ldots, e_{1dim}\} \subset E. \quad (1)$$

3.2.4 Exemple

Pour illustrer la projection d'un document vidéo dans l'espace des événements, nous allons considérer que les caractéristiques audiovisuelles « virtuelles » étudiées sont :

$$C = \{\text{forme}, \textit{texture}, \textit{emplacement}\}, \quad (2)$$

avec les valeurs qu'elles peuvent prendre, respectivement :

- la forme : carré, rond ou triangle,
- la texture : rayé horizontal (**H**) ou diagonal (**D**), et
- l'emplacement : milieu (**M**), ou coin (**C**).

Soit, par exemple les deux documents vidéo V_1 et V_2 composés chacun des trois suites d'images représentées respectivement comme suit dans la figure 1.

En comparant ces deux documents, et après une fusion par addition, nous obtenons la matrice de comparaison schématisée dans la figure 2. En conséquence la projection de V_1 dans l'espace des événements E serait comme dans la figure 3. Et V_1 sera représenté dans la figure 4.

Parmi les événements dans l'espace E, on trouve toutes les combinaisons possibles des caractéristiques entre elles. Or, il ne nous est pas indispensable de les connaître, car ce qui nous intéresse pour la comparaison de V_1 et V_2 est la projection des documents dans E, donc seuls les événements rencontrés.

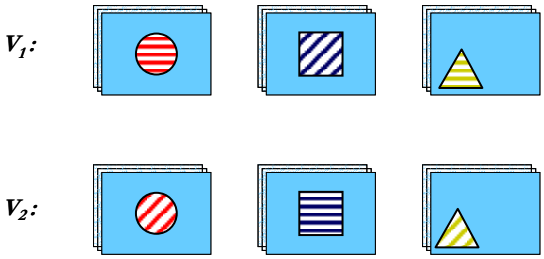


Fig. 1. Deux documents vidéo V_1 et V_2 représentés par les trois caractéristiques : forme, emplacement et texture.

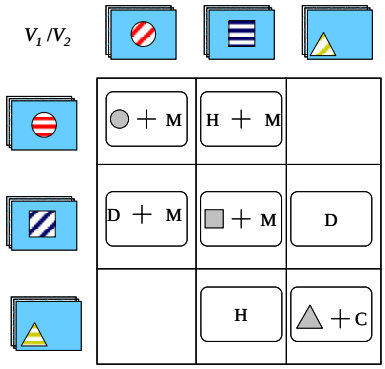


Fig. 2. Matrice de comparaison de V_1 et V_2.

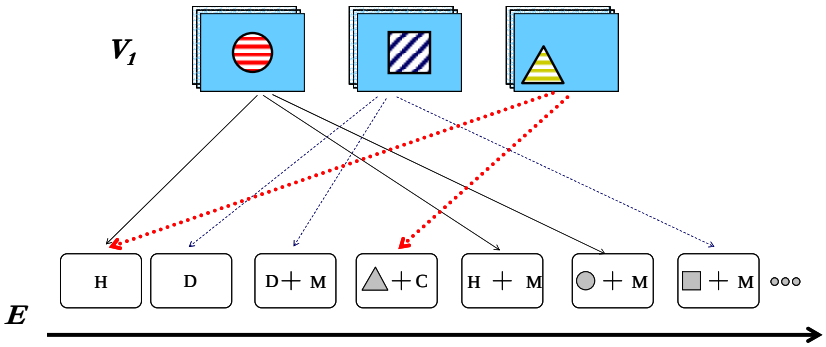


Fig. 3. Projection de V_1 dans E.

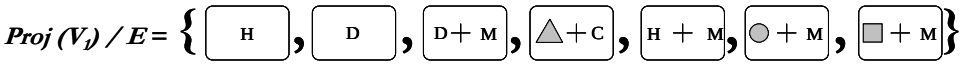


Fig. 4. Représentation de V_1 dans E.

3.2.5 Taille des événements

Tous les événements de base ont la longueur *tMin*, la longueur unité d'un événement. La combinaison de deux ou plusieurs événements est un événement.

Si nous prenons l'exemple précédent, nous savons que [H+M] et [D] sont des événements de taille *tMin*. Or dans la matrice de comparaison ces deux événements apparaissent successivement sur la même diagonale. La successivité de ces événements est un événement à lui seul, de taille 2 × *tMin*. Cet événement est écrit : [[H+M], [D]]. (Figure 5)

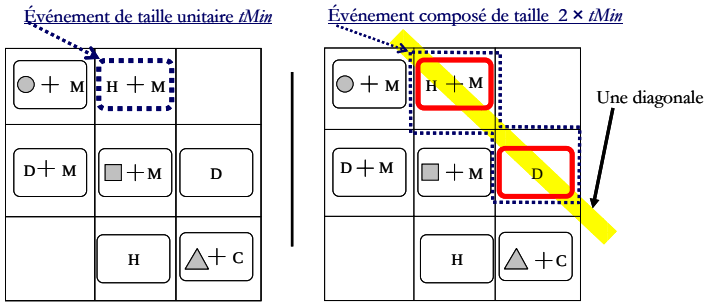


Fig. 5. Evénement unitaire et événement composé.

Ainsi, la publicité pour des magasins d'électroménager, si elle est répétée dans deux documents, devient un événement à elle seule. Il en va de même pour le bulletin météo. Mais le fait que ces deux documents apparaissent systématiquement successivement leur confère le statut d'un unique événement.

3.2.6 Nécessité d'une mesure de similarité

Considérons une matrice de comparaison représentant la comparaison des deux documents vidéo dans l'espace d'événements *E*. Le type de ces événements sera lié à la fusion choisie pour la matrice de comparaison. Que ce soit une matrice issue d'une seule caractéristique ou bien d'une fusion complexe entre plusieurs caractéristiques, il devient utile de définir des mécanismes pour son interprétation et son exploitation. C'est en particulier pour répondre à ces objectifs que nous définissons une mesure.

Avant de procéder à la présentation de cette définition, considérons ces quelques points de réflexion, qui influent sur la définition du concept de similarité, et qui justifieront les choix que nous aurons à faire plus tard dans ce chapitre.

3.2.7 Relativité de la similarité

3.2.7.1 Contexte de la mesure

Le premier point de la relativité de la similarité est le contexte. Nous désignerons chaque événement (correspondant à de courtes séquences vidéo d'une durée d'au moins tMin) par une chaîne de lettres minuscules. Soit :

$$e1 = \mathbf{ab}\,;\ e2 = \mathbf{ac}\,;\ \text{et}\ e3 = \mathbf{yz}\,; \tag{3}$$

La similarité entre *e1* et *e2* peut être discutée. Ces deux événements ne se ressemblent qu'à moitié. Or ils seront plus volontiers considérés comme similaires si on les confronte au troisième événement *e3* composé d'une chaîne totalement différente : ***yz***.

3.2.7.2 Contenu versus composition temporelle

Si nous disposons aussi d'un quatrième événement, *e4*, tel que :

$$e4 = \mathbf{ca}\,; \tag{4}$$

Une décision de la sorte :

Mesure_de_Similarité (*e2*, *e1*) **[> ? = ? < ?]** Mesure_de_Similarité (*e2*, *e4*) (**5**)

Ne peut être prise qu'après une étude du contexte de l'application.

3.2.7.3 La détection de l'invariance

Le dernier point que nous appellerons l'invariance, s'illustre à son tour par l'exemple suivant :

$$e5 = abacadae\,; \tag{6}$$

Comparer l'événement *e5* avec lui-même va nous permettre de découvrir que ***'a'*** est un événement *invariant* à l'événement composé *e5*.

3.2.8 Transitivité de la similarité

D'après tout ce qui précède, nous pouvons déduire qu'un événement, *ei*, peut ressembler à deux autres événements, *ej* et *ek*, sans que ces deux derniers ne soient semblables. Les caractéristiques communes entre *ei* et *ej* différent de celles entre *ei* et *ek* (figure 6).

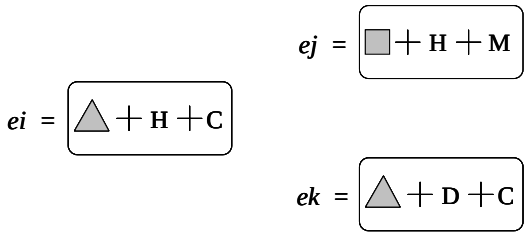


Fig. 6. Illustration de la non transitivité de la similarité.

Nous pouvons déduire que la similarité dont nous parlons est non transitive. Par suite la pseudo distance de similarité que nous allons proposer à la fin de ce chapitre ne vérifie pas l'inégalité triangulaire, et donc n'est pas une distance.

3.3 Mesure de similarité de style

3.3.1 Interprétation de la matrice

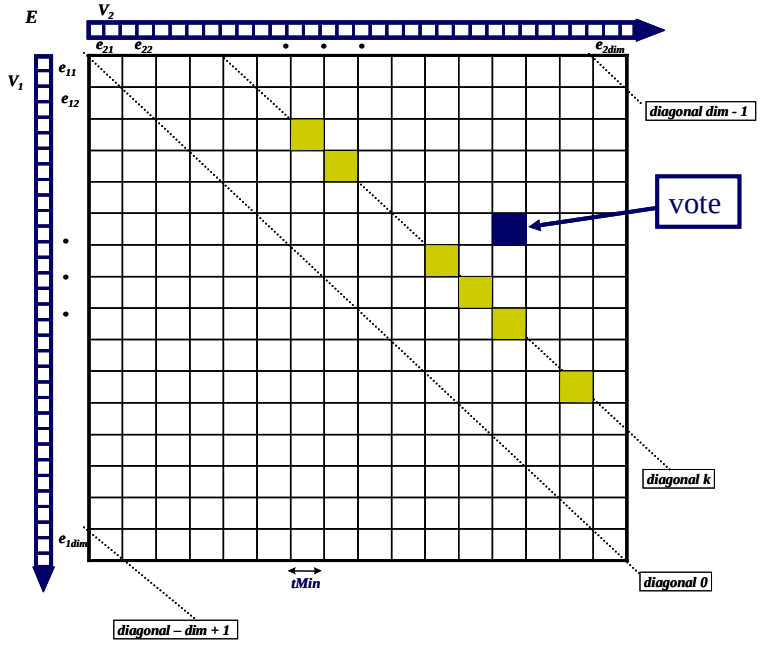


Fig. 7. La matrice de comparaison vue comme une matrice de votes sur la similarité des événements composants chaque document.

Une comparaison est une évaluation des ressemblances et des différences. Donc elle doit être sensible aux deux propriétés.

Etant donné deux documents vidéo à comparer, V_1 et V_2, et une matrice carrée de comparaison *MC*, de dimension *dim*, construite sur ces deux documents. Nous appelons diagonale zéro, la diagonale principale de la matrice. Cette diagonale comporte *dim* éléments. En partant vers la droite les indices des diagonales incrémentent jusqu'à *dim* – 1, tandis que en allant vers la gauche, ils décrémentent pour atteindre 1- *dim* (figure 7).

Considérons maintenant la $k^{ième}$ diagonale de cette matrice. Cette diagonale est la représentation du résultat de la comparaison pour un décalage de temps égal à $k \times tMin$ de V_2 par rapport à V_1. En effet, la matrice *MC* est de la forme suivante :

Par suite, des valeurs élevées sur une diagonale *k* donnée indiquent une forte similarité entre les événements correspondants dans chaque document (figure 8).

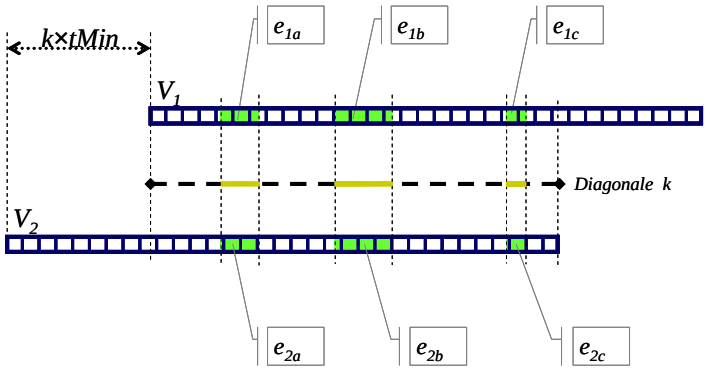


Fig. 8. La superposition des deux documents V_1 et V_2 avec un décalage de $k \times tMin$ unités de temps de V_1 par rapport à V_2. La diagonale *k* de la matrice de comparaison indique une certaine similarité entre des événements des deux documents.

Ainsi la matrice dans son intégralité récapitule toutes les possibilités de décalage, positif et négatif, de V_2 relativement à V_1 et des résultats de comparaison pour chaque alignement.

3.3.2 Densité et répartition des votes

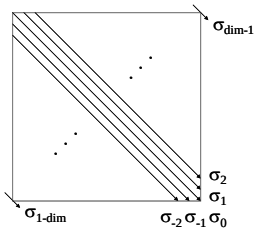


Fig. 9. Calcul de la densité de la matrice diagonale par diagonale.

Les valeurs élevées dans la matrice sont dues à des événements invariants ou semblables. Ce qui implique que la densité des éléments dans la matrice est proportionnelle au degré de

similarité de style entre V_1 et V_2. La figure 9 représente les éléments de chaque diagonale de la matrice.

3.3.2.1 Première mesure intuitive

Une mesure intuitive peut être définie alors, sur la base de cette densité. Définissons M une telle mesure. Lorsque dim^2 est la dimension de la matrice carrée, et σ_i la somme des valeurs sur la diagonale i, nous posons :

$$M = \frac{\sum_{i=-\dim+1}^{\dim-1} \sigma_i}{\dim^2} \quad (7)$$

Bien qu'elle donne une idée de la similarité entre V_1 et V_2, la mesure définie ci-dessus ne tient pas compte de la distribution spatiale des points dans l'espace de la matrice.

	a	*b*	*c*	*d*
a	1	0	0	0
b	0	1	0	0
c	0	0	1	0
d	0	0	0	1

(a)

	a	*b*	*c*	*d*
b	0	1	0	0
b	0	1	0	0
b	0	1	0	0
b	0	1	0	0

(b)

Fig. 10. Importance des poids diagonaux.

Prenons un exemple. Dans un premier cas, figure 10.a, nous avons comparé deux vidéos simples comprenant chacune quatre événements a, b, c et d. La matrice de comparaison correspondante montre des valeurs élevées sur la diagonale 0. Ces deux documents comparés auront une mesure $M1=1/4$.

De l'autre côté, figure 10.b, nous comparons le document comportant le même événement b répété quatre fois, avec le document comportant les quatre événements a, b, c et d successivement. Nous pouvons voir les valeurs élevées en forme verticale indiquant la similitude des événements b. Ces deux documents comparés auront aussi une mesure $M2=1/4$.

Nous nous attendons intuitivement à ce que les deux premiers documents comparés – figure 10.a- produisent une plus grande mesure de similarité que les deux autres –figure 10.b- en raison de la distribution diagonale de leurs points.

Une mesure de similarité plus précise doit pondérer les valeurs de la matrice afin de prendre en considération ce cas.

3.3.2.2 Pondération des votes

Le raisonnement est le suivant. Comme le remplissage de la matrice s'est fait en diagonale, la lecture doit se faire aussi en diagonale. Cette constatation était déjà présupposée dans la définition de la mesure précédente, même si son résultat équivalait à une moyenne globale des valeurs de la matrice. Si nous voulons distinguer les mesures issues des deux matrices de la figure 10, il nous faut pondérer les points de la matrice selon leur distance à une diagonale *k*. Mais par rapport à quelle diagonale faut il pondérer ? En fait, c'est variable.

Prenons un autre exemple. Si un document V_1 est comparable à la deuxième moitié d'un document V_2, la mesure de similarité devrait alors être définie relativement à ce décalage dans le temps, en pondérant le nombre de votes sur la diagonale *k=(dim-1)/2* avec des valeurs élevées.

3.3.3 Identification de scénarios

La diagonale de référence (et par suite de pondération) est celle qui correspond au meilleur alignement possible des deux documents. Dans un premier temps, supposons que nous connaissons, avec une certaine précision, cette diagonale. Soit *k* l'indice de cette diagonale. Pour évaluer la similarité des deux documents comparés, il suffit de calculer la mesure de similarité en pondérant la région où se trouve la diagonale *k* par les poids les plus élevés, comme étant la région de la matrice la plus fiable.

En conséquence, différents scénarios de pondération peuvent être identifiés. Nous détaillons les différents scénarios clés, en donnant des exemples.

3.3.3.1 Décalage constant du temps

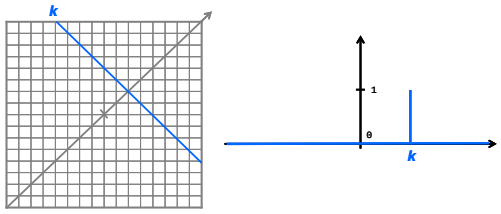


Fig. 11. Pondération pour un décalage constant du temps.

Commençons par le cas très particulier d'un même document vidéo, diffusé deux fois sur une même chaîne de télévision, par exemple le soir et après minuit. Les deux enregistrements de ce document, théoriquement identiques, mais bruités à la diffusion, peuvent aussi présenter un léger décalage de l'un par rapport à l'autre.

Ce décalage résulte de l'imprécision du début de l'enregistrement. Il est, par contre, constant pendant toute la durée des enregistrements. Nous appelons ce scénario le décalage constant du temps.

Dans le cas du décalage constant du temps, nous tenons compte seulement des points sur une diagonale spécifique *k* dans la matrice. Nous pondérons ces points par 1 et les autres par zéro (figure 11).

3.3.3.2 Décalage variable du temps

Un deuxième cas à considérer est celui des documents structurés mais dont le contenu varie fortement, comme par exemple, le tirage du loto, ou les journaux d'actualité de format court. Ces enregistrements sont effectués à un même créneau horaire, sur une même chaîne de télévision, à des jours différents correspondant à la diffusion de documents très structurés. La constance du décalage d'un enregistrement par rapport à l'autre est moins significative, mais elle reste tout de même assez typique et ne dépasse pas un petit intervalle.

En fait, si il existe des éléments semblables dans les deux vidéos, ces éléments sont dans un intervalle $[x_2, x_3]$ autour du point de synchronisation théorique donné par la diagonale de référence *k* (figure 12). Nous appelons ce scénario le décalage variable du temps.

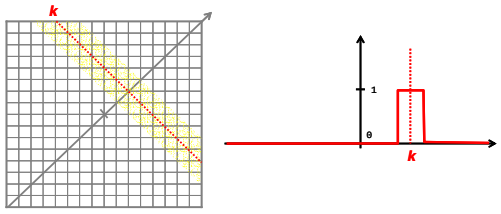


Fig. 12. Pondération pour un décalage variable du temps.

Dans le cas du décalage variable du temps, une bande de diagonales entourant la diagonale *k* est considérée avec la même importance que *k*. Les poids sont fixés à 1 pour des points dans la bande et à zéro pour les autres.

3.3.3.3 Synchronisme symétrique

Le troisième cas que nous considérons est celui des documents produits selon des directives donnant un fil conducteur à respecter. Le fil conducteur comporte alors des événements fixes (Point route, bourse, bulletin météo, etc.) d'une part, et des trous qui peuvent contenir des segments de natures et durées variables (reportages, films publicitaires, interviews). Les événements fixes, d'après leur nom, devraient se dérouler à un instant précis, se trouvant bien sûr sur la diagonale de référence. Ils sont bien entendu aussi les éléments comparables des deux documents, les autres segments étant très diversifiés. Nous pouvons déduire alors que, dans ce genre de scénario, plus on est en retard, c'est-à-dire, plus les éléments comparables s'éloignent de la diagonale de référence *k*, moins la similarité des événements est significative. Nous appelons ce scénario le synchronisme symétrique.

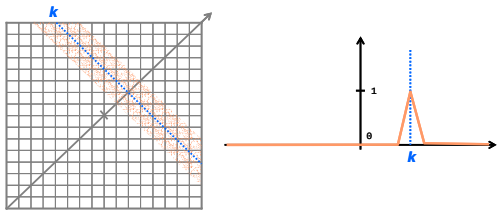


Fig. 13. Pondération pour un synchronisme symétrique.

Le cas du synchronisme symétrique, est le même que le précédent mais l'importance des points diminue en s'éloignant de la diagonale *k* (figure 13).

3.3.3.4 Synchronisme asymétrique

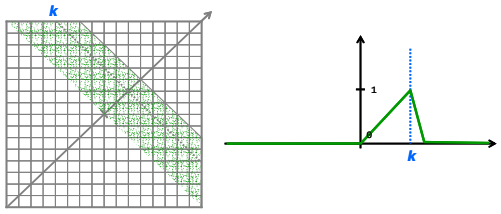


Fig. 14. Pondération pour un synchronisme asymétrique.

Un cas particulier est la restriction de la tolérance du décalage dans l'estimation de la similarité. Par exemple, on peut constater un retard systématique et jamais d'avance. Ceci se manifeste dans le cas où nous disposons d'un document de référence, avec un enchaînement type, et que nous le comparons avec d'autres documents. Il se manifeste, par exemple, dans une grille de programmes comportant des éléments à temps et ordre prédéfinis : loterie, ***puis***, journal, ***puis***, météo, etc. La diagonale *k*, dans ce cas ne sera pas au milieu de l'intervalle de synchronisation. Par suite nous appelons ce scénario, le synchronisme asymétrique (figure 14).

Dans le synchronisme asymétrique, à la différence du synchronisme symétrique, les deux côtés de la diagonale n'ont pas la même importance.

3.3.3.5 Cas général : combinaison de scénarios

En général, les enregistrements comparés, même lorsqu'ils disposent d'un point de synchronisation qu'est la diagonale de référence, présentent un scénario hybride. Ils comportent à la fois des documents rediffusés, des événements fixes et des événements très structurés, à coté de segments variables qui les différencient l'un de l'autre. Ils sont modélisés alors par une fonction plus complexe, comme dans la figure 15.

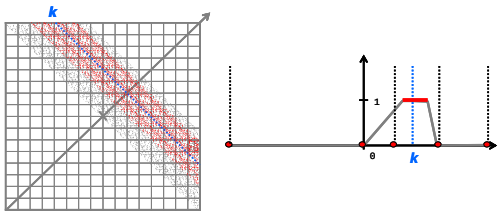


Fig. 15. Pondération générale.

Ces scénarios sont définis, lorsque le contenu des vidéos est structuré et évolue avec le temps. Ils mettent en valeur alors, l'ordonnancement sur l'axe temporel. Le paragraphe suivant abordera la fonction de pondération qui reflètera ces considérations dans le calcul de la mesure de similarité.

3.3.4 La fonction de pondération

Le but est de définir une fonction linéaire appliquée sur l'intervalle fermé [-dim+1, dim-1] pour pondérer les points sur chaque diagonale.

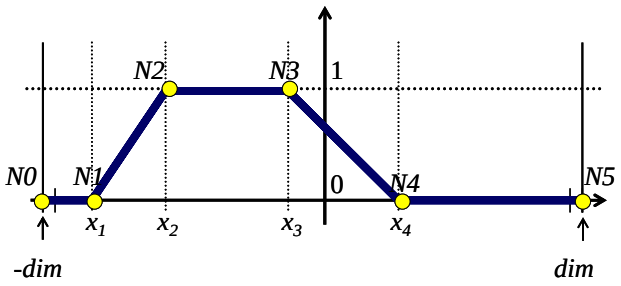


Fig. 16. La fonction de pondération *f* et ses paramètres. L'axe des abscisses représente les diagonales de la matrice de comparaison, l'axe des ordonnées donne les poids correspondants.

Les différents scénarios peuvent être réalisés par une fonction de pondération *f* à quatre paramètres (*x1, x2, x3, x4*). La fonction f_k représentée dans la figure 16 est alors appliquée pour distinguer les éléments de la matrice de comparaison relativement à leur distance à la diagonale *k*.

$$f_k : [1\text{-}dim, dim\text{-}1] \rightarrow]\,0,1] \qquad (8)$$

Dans la figure 16, la fonction f_k associe à chaque abscisse *i*, l'ordonnée du point sur le segment $]\text{-}dim, x_1] \cup]x_1, x_2[\cup [x_2, x_3] \cup]x_3, x_4[\cup [x_4, dim[$, où *k* est le milieu du segment $[x_2, x_3]$. Les coordonnées des cinq points sont fixées en fonction du scénario considéré :

$$N0\ (\text{-}dim, 0),\ N1\ (x_1, 0),\ N2\ (x_2, 1),\ N3\ (x_3, 1),\ N4\ (x_4, 0),\ \text{et}\ N5\ (dim, 0). \qquad (9)$$

Les abscisses de *N1*, *N2*, *N3* et *N4* sont des valeurs ordonnées dans [*N0*+1, *N5*-1] = [1 - *dim*, *dim* - 1]. Elles peuvent être confondues le cas échéant pour omettre un ou plusieurs segments.

Pour définir un scénario de pondération de centre k, nous pouvons tout simplement définir la relation des quatre paramètres avec k.

$$x_\varphi = k \pm t_\varphi,\ \varphi = 1, 2, 3 \text{ et } 4. \tag{10}$$

3.3.5 Normalisation des poids

Les poids de la fonction sont définis par :

$$w_i = f_k(i) = \begin{cases} -\dim < i \le x_1, & w_i = 0 \\ x_1 < i < x_2, & w_i = \dfrac{1}{x_2 - x_1} i - \dfrac{x_1}{x_2 - x_1} \\ x_2 \le i \le x_3, & w_i = 1 \\ x_3 < i < x_4, & w_i = \dfrac{-1}{x_4 - x_3} i + \dfrac{x_4}{x_4 - x_3} \\ x_4 \le i < \dim, & w_i = 0. \end{cases} \tag{11}$$

Pour normaliser ces poids, nous calculons la somme,

$$\sum_{i=1-\dim}^{\dim-1} w_i = \left[(x_1 - 1 + \dim + 1) \times 0\right] + \sum_{i=x_1+1}^{x_2-1} \left(\frac{1}{x_2 - x_1} i - \frac{x_1}{x_2 - x_1} \right) + \left[(x_3 - x_2 + 1) \times 1\right] + \sum_{i=x_3+1}^{x_4-1} \left(\frac{-1}{x_4 - x_3} i + \frac{x_4}{x_4 - x_3} \right) + \left[(\dim - 1 - x_4 + 1) \times 0\right] \tag{12}$$

$$\sum_{i=1-\dim}^{\dim-1} w_i = \tfrac{1}{2}(-x_2 - x_1 + x_3 + x_4) \tag{13}$$

et puisque,

$$-\dim \le x_1 < x_2 \le x_3 < x_4 \le \dim, \tag{14}$$

nous avons :

$$\Sigma = \sum_{i=1-\dim}^{\dim-1} w_i > 0\,. \tag{15}$$

Par suite nous définissons les poids normalisés :

$$W_i = \frac{w_i}{\Sigma}, \tag{16}$$

Nous définissons alors la mesure M_{fk} par :

$$M_{f_k} = \sum_{i=1-\dim}^{\dim-1} (W_i \times \sigma_i), \tag{17}$$

3.3.6 Normalisation des diagonales

Pour que tout alignement des deux documents soit équitablement traité, il faut normaliser les densités des différentes diagonales. En fait, la diagonale *j* peut contenir au plus *(dim-|j|)* éléments. Par suite les tailles des diagonales sont inégales.

Il y a deux possibilités :

- Normaliser M_{fk} et la rendre indépendante de la taille de la matrice, et correcte par rapport aux tailles inégales des diagonales. Il convient pour cela de donner la même importance au contenu de toutes les diagonales pour ensuite, pondérer pour privilégier la bande $[N_2, N_3]$ désirée.

  Or cette normalisation amplifie l'importance donnée aux diagonales sur les bords (pour lesquelles nous avons peu d'informations) par rapport au centre, en prenant le risque d'amplifier du bruit potentiel.

- Ou bien, garder la mesure sans normalisation, et discriminer entre les alignements temporels des deux documents. Par contre, il s'agit ici de définir ici un critère de distance et non pas d'une mesure. Une distance exige une identité parfaite entre les deux documents (y compris un alignement synchrone sur la diagonale 0) pour qu'elle soit nulle.

Nous appliquons le premier choix, tout en reportant le second pour la définition d'une pseudo distance de similarité.

D'où,

$$M_{f_k} = \sum_{i=1-\dim}^{\dim-1} \left(W_i \times \frac{\sigma_i}{\dim - |i|} \right). \tag{18}$$

Nous posons

$$\sigma'_i = \frac{\sigma_i}{\dim - |i|}. \tag{19}$$

Nous avons alors une valeur normalisée σ'_i, quelque soit la taille de la matrice et le numéro de la diagonale dans la matrice.

3.3.7 Définition de la mesure de similarité

Nous définissons la mesure de similarité de style M_{fk} relative à k avec la fonction de pondération f, par :

$$M_{fk} = \sum_{i=1-\dim}^{\dim-1} (W_i \times \sigma'_i) \quad (\mathbf{20})$$

Cette mesure proposée dépend directement de la densité de la matrice et de la distribution des éléments de valeur élevée autour d'une diagonale spécifique désignée sous le nom de la diagonale d'alignement.

En l'absence de précision de la fonction de pondération f, dans la notation de mesure, nous supposons qu'il s'agit d'un scénario de décalage constant de temps; les relations sont données par :

$$x_1=k-1,\ x_2=x_3=k \text{ et } x_4=k+1. \quad (\mathbf{21})$$

Ce cas particulier sera utile pour imposer la prise en considération de l'ordonnancement temporel dans la définition qui sera donnée ultérieurement d'une pseudo distance.

D'une manière générale, étant donné un scénario muni d'une pondération f, la mesure de similarité de style M_f entre deux documents comparés est donnée par la diagonale k qui assure leur meilleur alignement :

$$M_f = \max_{k=-\dim+1}^{\dim-1} M_{fk} \quad (\mathbf{22})$$

3.4 Pseudo distance de similarité

3.4.1 Pourquoi une distance ?

Une distance de similarité entre deux documents vidéo bien définie doit se baser non seulement sur la similarité du contenu et sur l'ordonnancement temporel des éléments comparables mais aussi sur la synchronisation des documents en général. Elle doit vérifier les propriétés d'une distance et notamment doit s'annuler seulement lorsqu'un objet est comparé à lui-même. Donc le décalage par rapport à la diagonale 0 doit être pénalisé et par conséquence la normalisation des diagonales n'est pas appropriée pour cette distance.

3.4.2 Proposition d'une pseudo-distance

Pour une pseudo-distance qui doit être nulle seulement quand un sous-ensemble dans l'espace des événements, E, est comparé à lui-même, il est nécessaire de discriminer les diagonales afin de tenir compte du décalage de temps.

Nous définissons dans un premier temps la mesure de similarité modifiée, M'_{fk}, par :

$$M'_{fk} = \sum_{i=1-\dim}^{\dim-1} (W_i \times \sigma''_i), \tag{23}$$

où σ''_i est la densité de la diagonale i relativement à la taille de la diagonale principale :

$$\sigma''_i = \frac{\sigma_i}{\dim}. \tag{24}$$

Nous définissons la pseudo-distance d_{fk} dans l'espace des événements E, comme suit :

$$d_{fk} = 1 - M'_{fk}. \tag{25}$$

En particulier, $d_k = 1 - M'_k$ est la pseudo-distance de décalage constant de temps relatif à k. La pseudo-distance de similarité entre deux documents V_1 et V_2, pour un scénario donné représenté par une fonction f, est donnée par la pseudo-distance obtenue relativement au meilleur alignement k :

$$d_f(V_1, V_2) = \min_{k=-\dim+1}^{\dim-1} \left(d_{fk}(V_1, V_2)\right) \tag{26}$$

3.4.3 Mesure de similarité versus pseudo-distance de similarité

Du fait qu'elle ne pénalise pas le choix de la diagonale de pondération, la mesure de similarité est parfaitement adaptée pour la mesure du style, surtout quand elle est associée à un scénario non strict (cas du décalage constant du temps). La pseudo-distance, elle, est plus sensible aux différences en structure.

Nous pouvons conclure que la pseudo-distance est plus appropriée pour mesurer la similitude en structure entre les éléments d'une même collection, tandis que la mesure de similarité est plus appropriée pour mesurer la similitude de style entre les documents visuels d'un même genre.

3.5 Conclusion

Dans ce chapitre, nous avons proposé une mesure de similarité de style qui permet d'aborder des problèmes plus complexes que la comparaison de deux versions d'un même document.

Nous avons essayé de nous déconnecter de l'échelle des documents comparés. Ainsi la théorie que nous avons proposée est aussi bien applicable pour la comparaison de deux documents de type jeux télévisés (30 minutes), comme pour la comparaison de deux films publicitaire de durée de quelques secondes, et pour la comparaison de deux journées entières de flux de télévision. Les seuls paramètres à adapter dans ces différents cas, les paramètres de l'ordre technique, *tMax* et *tMin*. Les raisons principales de cette adaptation sont d'un ordre technique : la taille de la matrice ainsi que le temps de calcul estimé deviennent énormes dans le cas où l'on garde le même paramétrage des films publicitaires pour comparer les flux télévisés (24 heures d'affilé)

A part l'échelle, les tailles différentes des documents impliquent des homogénéités du contenu différentes. Dans une émission d'un jeu télévisé le contenu est « homogène », tandis que dans une journée de télévision le contenu est évidemment diversifié. Les mesures que nous avons proposées dans ce chapitre sont parfaitement adaptées dans tous ces cas à travers les scénarios de synchronisation que nous avons proposés.

4 Chapitre 4 : Applications

Chapitre 4

APPLICATIONS

4.1 Introduction

Nous présentons dans ce chapitre des expériences originales dont le but est d'illustrer l'application et de montrer l'utilité des outils proposés dans les chapitres 2 et 3 sur des documents vidéo de tailles différentes.

Notre démarche est guidée par les motivations suivantes :

- Nous cherchons à mettre en évidence la capacité de notre méthode à traiter tout type de contenu sans intervention spécifique sur ces contenus. Nous effectuerons de ce fait les tests sur des documents courants sans adaptation ni apprentissage.

- Une des règles que nous nous sommes fixées au début de nos travaux était la généricité de la méthode. Pour cela nous avons expérimenté nos outils sur un ensemble varié de documents vidéo

    - de genres différents, et de

    - durées différentes.

Enfin nous nous appuierons sur ce dernier chapitre pour introduire de nouveaux axes de recherche en analyse des contenus vidéo basés sur la matrice de comparaison et la mesure de similarité, et soulignant les points d'amélioration possible.

4.2 Méthodologie

4.2.1 Extraction des caractéristiques

La première étape, avant toute application de comparaison des documents vidéo par les méthodes que nous proposons dans cette thèse, est l'extraction d'un ensemble de caractéristiques bas niveau. La construction de la matrice de comparaison, conçue pour être générique pour tout type de caractéristique audiovisuelle de bas niveau pouvant être vue comme une série chronologique, est indépendante du choix des caractéristiques.

Dans nos applications, nous sélectionnés des caractéristiques audio et vidéo bas niveau, issues d'outils développés dans l'équipe SAMOVA à l'IRIT selon des critères de simplicité et d'efficacité. Nous présentons ces caractéristiques et les mécanismes de leur extraction dans la section 3. Ces outils d'extraction sont accessibles via une plateforme dynamique de chaînage et d'indexation Pidot (en construction) [Haidar 05b].

Il a été nécessaire de synchroniser les résultats des traitements sur les deux modes pour la phase de fusion. Les caractéristiques audio sont obtenues sur des intervalles d'une seconde. Celles de la vidéo sont obtenues à raison d'une valeur par image, avec un échantillonnage temporel dépendant alors de la fréquence des images qui est 25 images par seconde pour le SECAM et 30 images par secondes pour le NTSC (nous avons utilisé des enregistrements d'émissions françaises et américaines). De ce fait, nous avons étendus les caractéristiques audio afin d'obtenir des vecteurs synchrones, de même longueur que ceux de la vidéo.

4.2.2 Lecture des matrices de similarités

La génération de la matrice a été présentée dans le chapitre 2. Or la lecture de cette matrice, c'est-à-dire l'extraction automatique de toutes les informations utiles dans la matrice, ne peut être définie par des mécanismes de lecture simples.

En effet, l'exploitation de cette matrice de similarité dépend fortement du contexte de l'application. Nous présentons dans ce qui suit différents contextes. Chaque application impose une méthodologie de lecture qui lui est adaptée. Cette lecture impose généralement une phase de post-traitement de type lissage et/ou filtrage dont les paramètres sont fixés empiriquement ou en fonction d'une analyse de la matrice. De ce fait, avant de procéder à l'extraction d'une information quelconque, la matrice est analysée dans le but d'étudier son homogénéité ainsi que des statistiques sur l'intensité et la répartition des éléments.

4.3 Caractéristiques utilisées

Nous présentons dans ce paragraphe et le paragraphe suivant les caractéristiques audiovisuelles que nous avons utilisées dans nos expérimentations. Toutes les caractéristiques que nous utilisons sont extraites par des outils développés au sein de l'équipe SAMoVA, dont certains ont été utilisés pour participer à diverses campagnes d'évaluation, comme par exemple NIST et TRECVIDEO. D'autres caractéristiques de niveau intermédiaire, sont de même disponibles, par exemple, la détection et l'identification de personnes, la reconnaissance des applaudissements, des rires, de la parole et de la musique, etc. Dans la mesure où ces caractéristiques peuvent être considérées comme des séries chronologiques, même si ce sont des séries binaires parfois, rien n'empêche de les utiliser dans de futures expérimentations.

4.3.1 Caractéristiques vidéo

4.3.1.1 Outil d'extraction baseindexvid

Le projet KLIMT (KnowLedge InterMediation Technology) est un projet européen qui s'inscrit dans le cadre du programme européen ITEA (Information Technology for European Advancement). Ce projet s'est déroulé entre 2002 et 2004. Son but était de construire une technologie qui permette la collaboration entre différents services accessibles sur une plateforme. Parmi ces services, plusieurs outils d'analyse audiovisuelle étaient proposés par des équipes de recherches de l'IRIT et du LIP6 et des entreprises comme Thalès RT, ISOFT, SINEQUA, ou I&IMS. [Conan 03]

Ce projet nous a conduit en particulier à mettre à disposition et à adapter un système d'extraction de descripteurs audiovisuels. Cet outil, baseindexvid, produits des résultats au format XML à partir de documents vidéo MPEG1 et MPEG2.

Ces descripteurs se repartissent essentiellement en quatre types :

- détection de changement de plan et de transition
- extraction des images clés de la vidéo
- description du taux d'activité
- extraction de caractéristiques : texture, couleurs dominantes, contraste

Ces descripteurs sont ensuite transmis aux autres outils de la plateforme. Par exemple, les images clés font l'objet de la vérification de la présence de visage ou de texte.

4.3.1.2 Les deux couleurs dominantes

La première couleur dominante est obtenue à partir de l'analyse des histogrammes en HLS (hue, luminance et saturation) calculés sur l'image. Dans un premier temps, la teinte H1 la

plus représentée est extraite. On cherche alors la saturation S1 la plus fréquente des pixels ayant cette teinte. Sur les pixels correspondants à cette saturation on ne retient que ceux qui ont la luminance L1 la plus souvent représentée.

L'ordonnancement de ces filtres à fait l'objet d'une étude développée pour un projet précédent appelé « ViewTime ».

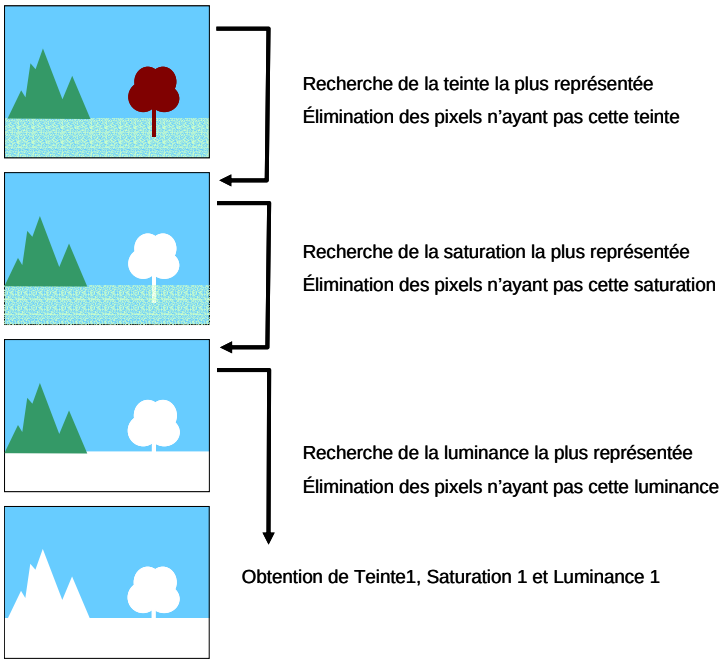


Fig. 1. Extraction de la première couleur dominante.

De cette manière, on extrait dans le cas général, la couleur de la plus grande zone monochrome.

L'objectif de l'extraction de la deuxième couleur dominante est de rendre compte du contraste de l'image. Pour cela, sur l'histogramme en teinte de l'image, on sélectionne la teinte la plus représentée tout en étant la plus « éloignée » de celle sélectionnée lors du choix de la couleur dominante 1. L'éloignement est calculé sur le cercle chromatique à l'aide d'une distance circulaire. La couleur dominante 2 a pour teinte celle dont la teinte maximise :

$$distcirc\ (H2, H1) \times (proportion\ de\ pixels\ ayant\ la\ teinte\ H2). \quad (1)$$

La sélection de L2 et S2 suivent le même procédé que pour la première couleur dominante. De ce fait, une image présentant une couleur unie ou une forte couleur dominante produira en couleur 1 et en couleur 2 sensiblement les mêmes résultats. Par contre une image très structurée dans l'arrangement des couleurs produira deux résultats différents. Notons que dans le cas d'images en N&B, on obtient le plus souvent les nivaux d'intensités les plus contrastées. L'algorithme utilisé est le même que pour la couleur dominante 1.

Voici quelques résultats. La première couleur dominante de chaque image est donnée dans un rectangle en haut à droite. La seconde apparaît juste en dessous.

Fig. 2. Couleurs dominantes extraites.

4.3.1.3 La luminance moyenne

La luminance peut rendre compte de procédés de mise en scène à travers des choix esthétiques – et techniques – portant sur l'éclairage. Par exemple, des actions dans un film peuvent se dérouler dans l'obscurité (par exemple la nuit, sous l'eau, ou dans un souterrain).

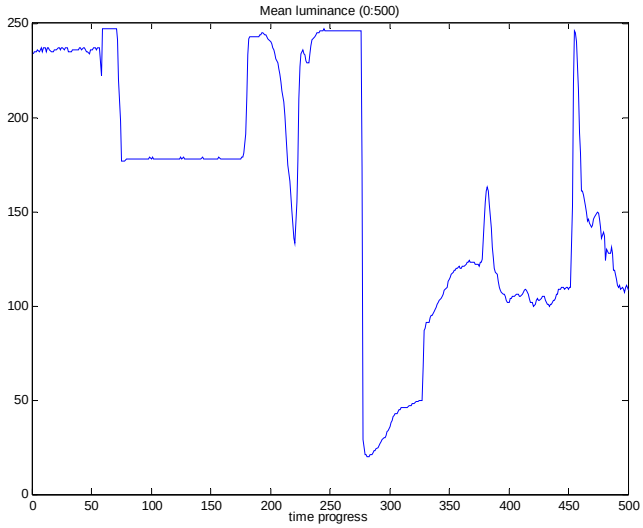


Fig. 3. La luminance moyenne pour une séquence vidéo de 20 secondes.

La luminance, dans notre cas, est calculée sur les coefficients DC des images, c'est-à-dire, elle est exprimée par la moyenne des luminances des pixels par blocs 8x8 :

$$L_p = (R_p + G_p + B_p)/3 \quad (2)$$

$$\text{Lum\_Moy} = \sum L_p / \text{nb\_coef\_DC} \quad (3)$$

Les valeurs de la luminance moyenne varient entre 0 et 255.

Bien entendu, il existe différents modèles de la perception de l'intensité lumineuse dont certains sont spécialisés dans la production d'une mesure globale dite de « luminosité » (lightness). Parmi ces modèles nous pouvons citer :

- le modèle de Priest :

$$\Lambda = (L)^{\frac{1}{2}}, \text{ où } \Lambda \text{ est la mesure et } L \text{ la luminance} \quad (4)$$

- le modèle de Ladd et Piney :

$$\Lambda = 2{,}468 \times (L)^{\frac{1}{3}} - 1{,}636 \quad (5)$$

- le modèle de Foss :

$$\Lambda = 5 \times \log_{10}(L) + 0{,}25 \quad (6)$$

- le modèle de Judd :

$$\Lambda = \frac{0{,}1 \times L \times (L_b + 100)}{L_b + L} \quad (7)$$

où L_b est la luminance moyenne du fond de l'image. Ce dernier modèle ne peut donc s'appliquer qu'à l'analyse d'images dont le contenu est connu a priori.

Quelque soit le modèle, celui-ci dépend directement de la luminance. S'agissant de fonctions de transformation croissantes, la courbe représentant l'évolution de L est respectée à une échelle et une translation en amplitude près. L'observation de la distribution de la luminance moyenne sur de nombreux documents montre qu'il n'y a pas lieu de lui appliquer de telles transformations.

4.3.1.4 Le contraste

Nous définissions le contraste comme étant l'éloignement entre les deux couleurs dominantes selon la formule suivante :

$$contrast = |L_1 - L_2| + \text{distcirc}(H_1, H_2) \times \log\left(\frac{S_1 + S_2}{2}\right), \quad (8)$$

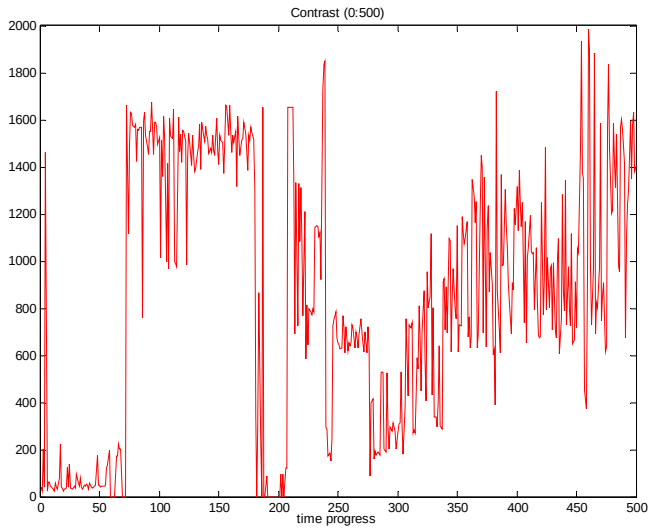


Fig. 4. Le contraste pour une séquence vidéo de 20 secondes.

4.3.1.5 Les orientations et granularités de texture

L'orientation et la granularité des textures sont une autre caractéristique couramment employée pour caractériser des images. Par exemple, dans des scènes de ville avec beaucoup de bâtiments, on peut s'attendre à de nombreux bords verticaux dans l'image. Ce type d'observation est moins probable dans des images de scènes naturelles.

Dans le but d'extraire ces caractéristiques à un moindre coût de calcul, nous n'avons pas utilisé l'approche classique consistant à appliquer une transformée fréquentielle (du type Transformée de Fourier Discrète) puis à calculer l'énergie du spectre à l'aide de bancs de filtres de Gabor. Nous avons simplement compté le nombre de pixels voisins présentant de forts changements d'intensité dans chacune des orientations verticale et horizontale. Ceci nous permet de produire à très faible coût l'estimation de deux caractéristiques : le taux de granularité verticale, et le taux de granularité horizontale.

4.3.1.6 Le taux d'activité

Le taux d'activité est une autre caractéristique vidéo importante. Il reflète les mouvements des objets ou de la caméra et les changements de plans. Il faut au moins 2 images pour calculer le taux d'activité. La décorrélation dans la vidéo provient essentiellement du mouvement et des transitions de plans.

Nous proposons un calcul de cette caractéristique basé sur le nombre de pixels dans l'image ayant changé significativement d'intensité. Une phase préalable à ce calcul, visant à stabili-

ser les valeurs de cette caractéristique, est l'égalisation de l'histogramme des niveaux de gris, dans le cas où il ne s'agit pas d'une image unie.

$$q\_mvt = \sum_{p=1}^{nb\_pixels} \left(1 \times \begin{cases} 1 \text{ si } \text{lum}_{\text{image}}(p) - \text{lum}_{\text{image\_precedente}}(p) > 128 \\ 0 \text{ sinon} \end{cases} \right), \quad (9)$$

4.3.2 Caractéristiques audio

4.3.2.1 Outil d'extraction des caractéristiques audio

En ce qui concerne l'audio, nous avons utilisé les caractéristiques bas niveau extraites par des outils développés dans l'équipe et utilisées essentiellement pour la classification parole/musique/bruit. Le système de fusion d'informations proposé par [Pinquier 03] est fondé sur l'extraction de quatre paramètres :

- la modulation de l'énergie à 4 Hertz,
- la modulation de l'entropie,
- le nombre de segments par seconde (les segments correspondent à des phases stationnaires du signal),
- la durée de ces segments.

Le système développé par J. Pinquier se décompose en deux systèmes de classification correspondant aux deux détections disjointes de la parole et de la musique. Pour la détection de la parole, les deux premières caractéristiques sont utilisées, tandis que pour la détection de la musique ce sont les deux dernières qui entrent en jeu. Ainsi, les passages contenant de la parole, de la musique mais aussi simultanément de la parole et de la musique sont détectés. La décision est prise en comparant des scores de vraisemblance issus de la modélisation de chacun des paramètres considérés.

Les paragraphes ci-dessous décrivent à grands traits les procédés mis en œuvre pour l'extraction de ces caractéristiques. On se réfèrera à [Pinquier 04] pour plus de détail.

4.3.2.2 Modulation de l'énergie à 4 Hertz

Le signal de parole possède un pic caractéristique de modulation en énergie autour de la fréquence syllabique 4 Hertz [Houtgast 85]. En effet, ces modulations correspondent au rythme syllabique. Pour extraire cette information, la procédure suivante est appliquée.

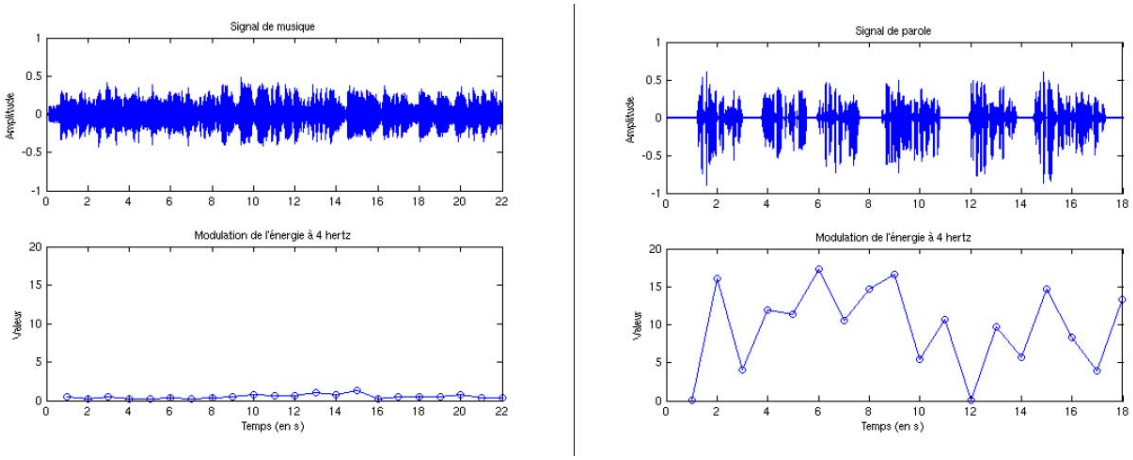


Fig. 5. Modulation de l'énergie à 4 Hertz pour la musique (extrait de Mozart) et la parole (6 phrases de parole lue).

1. Le signal est découpé en trames de 16 ms sans recouvrement.
2. Pour chaque trame, après un fenêtrage de Hamming, 40 coefficients spectraux MFCC et correspondent à l'énergie de 40 bandes de fréquence en accord avec les propriétés de l'oreille humaine.
3. L'énergie de chaque bande est filtrée grâce à un filtre passe-bande de fréquence centrale 4 Hertz.
4. La somme des énergies filtrées est effectuée sur l'ensemble des canaux et est normalisée par l'énergie moyenne.
5. La modulation est obtenue en calculant la variance de l'énergie filtrée en décibels, sur une seconde de signal.

La parole possède une modulation de l'énergie à 4 Hertz plus forte que la musique (exemples de la figure 5).

4.3.2.3 Modulation de l'entropie

Des observations menées sur le signal, ainsi que sur le spectrogramme, font apparaître une structure plus « ordonnée » du signal de musique que de parole. Pour mesurer ce «désordre», un paramètre fondé sur l'entropie du signal est calculé [Moddemeijer 89] :

$$H = \sum_{i=1}^{k} -p_i log_2 p_i \qquad (\mathbf{10})$$

avec p_i = probabilité de l'événement i et k = nombre d'événements.

Une procédure similaire à celle employée pour le paramètre de modulation de l'énergie à

4 Hertz est appliquée.

1. Le signal est découpé en trames de 16 ms sans recouvrement.
2. L'entropie est estimée pour chaque trame grâce à un estimateur non biaisé.
3. La modulation est obtenue en calculant la variance de l'entropie sur une seconde de signal. On obtient 62 valeurs de l'entropie par seconde.

Compte tenu du « désordre » relatif de la parole par rapport à la musique, la modulation de l'entropie est généralement plus élevée pour la parole que pour la musique.

4.3.2.4 Paramètres de segmentation

La longueur des segments quasi stationnaires est différente pour la parole et la musique. En utilisant une segmentation du signal en zones quasi stationnaires, on cherche à mettre en évidence cette information. Elle permet d'atteindre, pour la parole, une segmentation sub-phonétique.

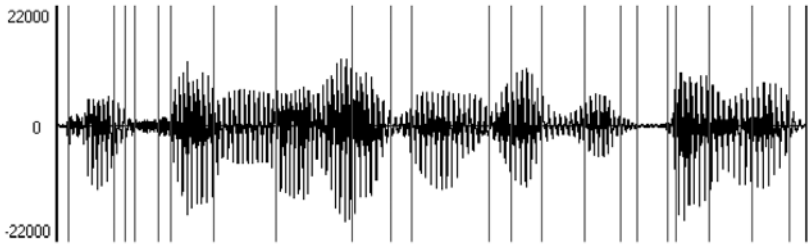


Fig. 6. Résultat de la segmentation sur environ 1 seconde de parole. La phrase prononcée est : «Confirmez le rendez-vous par écrit ».

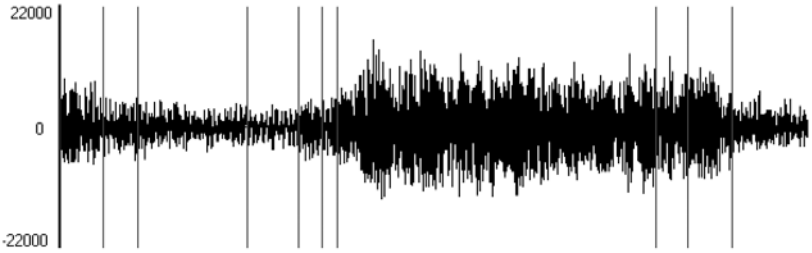


Fig. 7. Résultat de la segmentation sur environ 1 seconde de musique d'un extrait de l'enregistrement d'une oeuvre de Mozart.

La longueur d'un segment varie entre 20 et 100 ms pour la parole (figure 6). Pour la musique, un segment correspond à la tenue d'une note ; il peut être beaucoup plus long (figure 7).

Les deux paramètres suivants sont calculés après application de l'algorithme DFB (Divergence Forward-Backward) [André-Obrecht 88].

– Nombre de segments

Le nombre de segments présents durant chaque seconde de signal est calculé. Ce nombre est plus important pour la parole (23 segments dans notre exemple, figure 6) que pour la musique (10 segments dans notre exemple, figure 7).

– Durée des segments

La durée des segments est fortement corrélée au nombre de segments par seconde. Afin de limiter la corrélation de ces deux paramètres de segmentation, la durée moyenne des segments sur une seconde est calculée sur les 7 segments les plus longs de la seconde. Ce nombre de segments caractéristiques a été fixé expérimentalement. Les segments sont généralement plus longs pour la musique (180 ms dans notre exemple, figure 6) que pour la parole (80 ms dans notre exemple, figure 5).

4.4 Conception et mise en œuvre technique

4.4.1 Décodage vidéo

Le décodage de documents vidéo pour l'extraction des caractéristiques a été effectué avec la bibliothèque Libmpeg2 sous licence GPL [LibMpeg2]. Cette bibliothèque est l'une des plus rapides, avec une vitesse de décodage supérieure à celle la lecture du document sur les machines actuelles. La majeure partie du code est écrite en C.

4.4.2 Outils d'extraction et de comparaison

Après une phase de test développée sous Matlab, nous avons développé les outils de comparaison en C++ et les outils d'extraction des mesures de similarité en C pour des raisons gains en temps de calcul.

4.4.3 Parallélisation

Les algorithmes de comparaison définis dans le chapitre 2 sont des outils très rapides. Ils peuvent être facilement parallélisés pour réduire la durée de la comparaison en raison des appels quadratiques indépendants mis en oeuvre, d'une part, et en raison de l'exploitation séparée des caractéristiques audiovisuelles à comparer. Nous avons profité de ces propriétés pour générer une version effectuant des appels parallèles de notre outil de comparaison.

Le partage du travail consiste essentiellement à :

- exécuter une boucle par répartition des itérations entre les tâches ;
- exécuter plusieurs sections de code mais une seule par tâche ;
- exécuter plusieurs occurrences d'une même procédure par différentes tâches (orphaning).

Nous avons utilisé les deux premiers types de parallélisation :

- le premier (parallélisation de boucle) dans :
  - l'appel de comparaison des caractéristiques chacune à part, et dans
  - les calculs effectués sur les matrices de comparaison dans le processus de fusion inter caractéristiques,
- le deuxième (parallélisation des sections) dans :
  - les appels quadratiques de l'algorithme IQR, et dans
  - les appels doubles ou quadratiques (selon le critère, voir chapitre 2 algorithme CC) pour les calculs des couvertures.

L'annexe B donne plus de détails concernant les éléments de mise en œuvre de la parallélisation sur le super calculateur de l'IRIT : CALIF.

4.5 Expérience 1 : Etude du style d'un film de cinéma

4.5.1 Description et but

Deux motivations essentielles nous ont poussé à appliquer la méthode d'auto-comparaison sur le film Matrix Reloaded.

1. Il s'agit d'un long-métrage présentant une structure narrative classique. Nous souhaitons étudier la cohérence et l'homogénéité des différentes parties ou scènes composant le film. Les changements de rythme, de décor et des procédés de mise en scène sont des événements que l'ont peut supposer voir se refléter au niveau des caractéristiques bas niveau.
2. D'un autre coté, ce film a fait l'objet d'un travail artistique particulièrement soigné qui se transcrit directement sur certaines caractéristiques de bas-niveau. En particulier, l'étalonnage bien spécifique, visant à donner un ton « verdâtre » aux images doit pouvoir transparaître à travers l'analyse des couleurs dominantes. Les très nombreux effets spéciaux liés à la production de ralentis ou d'accélérés doivent également laisser des traces caractéristiques après analyse de la quantité de mouvement.

L'analyse que nous effectuerons de la matrice de comparaison pourra être rapprochée d'une critique du film que le lecteur pourra trouver en annexe C.

4.5.1.1 Conditions de l'expérience

La durée du document est de 2 heures et quelques minutes. Il est codé en format mpeg2, en 25 images par secondes. En ce qui concerne le paramétrage des deux seuils de la comparaison :

- le paramètre *tMin* a été initialisé à 2^6 unités ce qui équivaut à 2.5 secondes, tandis que,
- le paramètre de filtrage *tMax* a été initialisé à 2^{10} unités équivalentes à 41 secondes.

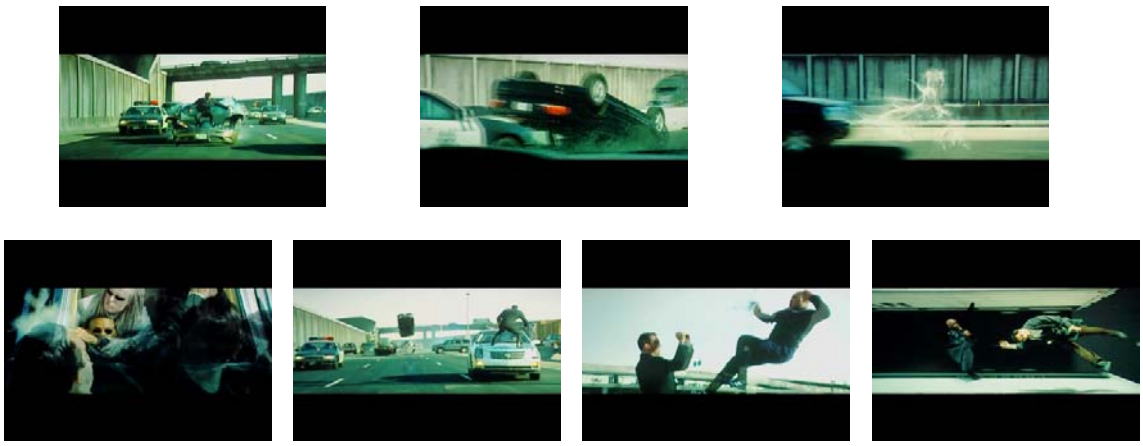


Fig. 8. La poursuite des voitures.

Ces initialisations, proches de celles utilisées dans la comparaison des journaux télévisés, et des jeux télévisés présentée plus loin, ont été légèrement augmentées dans le seul but de réduire le temps de calcul.

Les 15 caractéristiques utilisées dans cette expérience proviennent de la vidéo (11 caractéristiques) et de l'audio (4).

4.5.2 La matrice de Matrix

Cette matrice est présentée dans la figure 9.

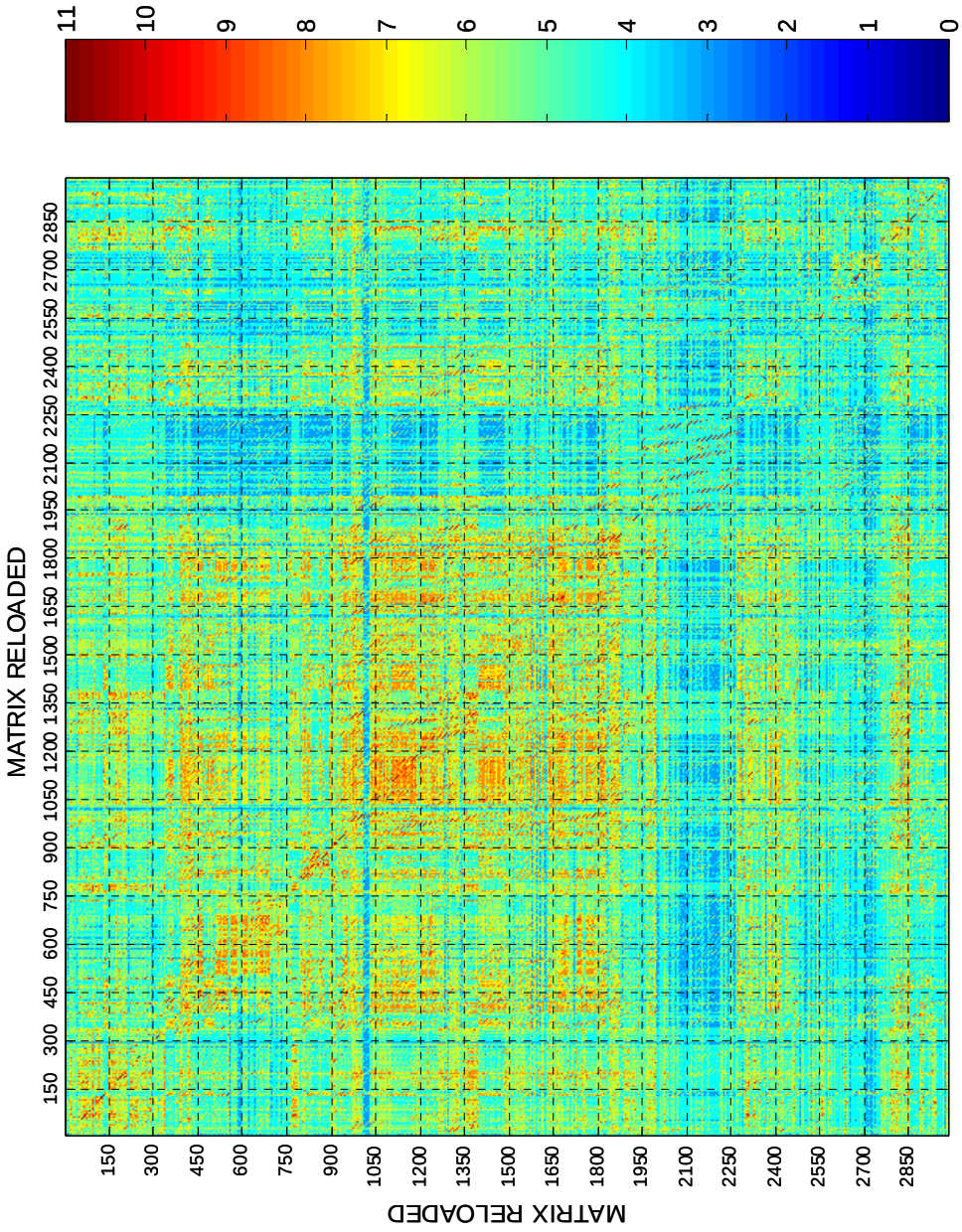


Fig. 9. La matrice de comparaison du film « Matrix Reloaded » avec lui-même. Une couleur chaude (resp. froide) indique un fort (resp. faible) taux de similarité.

4.5.3 Analyse diagonale de la matrice

Nous pouvons distinguer dans la matrice de comparaison trois phases principales : le début, le milieu et la fin. Le milieu est caractérisé par le bloc de fortes valeurs se situant approximativement entre les positions (300 : 300) et (1800 : 1800) de la grille de la matrice.

Ce rectangle contient en fait des scènes très homogènes dans lesquelles les caractéristiques évoluent doucement. Même dans le cas de la bataille avec les clones, qui se déroule entre les coordonnées 1050 et 1200, ces caractéristiques évoluent peu. L'interpolation entre images synthétisées, appuyée par un fort étalonnage visant à assurer la dominance des mêmes couleurs tout au long de la scène, donnent à cette séquence une très grande homogénéité visuelle, en induisant un effet de lissage sur les caractéristiques.

A part le combat, ce sont des scènes se déroulant soit dans le monde réel (associé visuellement à des images plutôt sombres), soit dans le monde virtuel (associé à des tonalités de couleur vertes). Le basculement entre les deux mondes est peu important dans cette partie.

C'est après la première heure du déroulement du film, au-delà de l'abscisse 1800, que ce basculement commence à être très rapide induisant ainsi de forts changements au niveau des caractéristiques.

A la seizième minute après la première heure, une déconnexion remarquable entre les images successives du film peut être remarquée à travers les faibles valeurs de la matrice entre les abscisses 1950 et 2250. C'est là que commence une poursuite de voitures avec un taux d'action très élevé et des mouvements rapides dans un endroit ouvert qu'est la route, à la lumière du jour, chose qui est rare dans le reste du film. Ces conditions induisent ainsi des évolutions perturbées au niveau des caractéristiques.

Fig. 10. Le combat avec les clones.

Le passage (1950 → 2250) dans le document se distingue de la plupart des autres passages pas les deux adjectifs suivants :

- tout d'abord, il ne ressemble pas aux autres. Cette information est directement lisible grâce à la distribution des valeurs faibles en ligne et en colonne,

- ensuite, ce passage n'est pas homogène. Cette information est à son tour visualisable via les valeurs fortes très clairsemées à l'intérieur des blocs diagonaux (1950 :1950) et (2250 :2250).

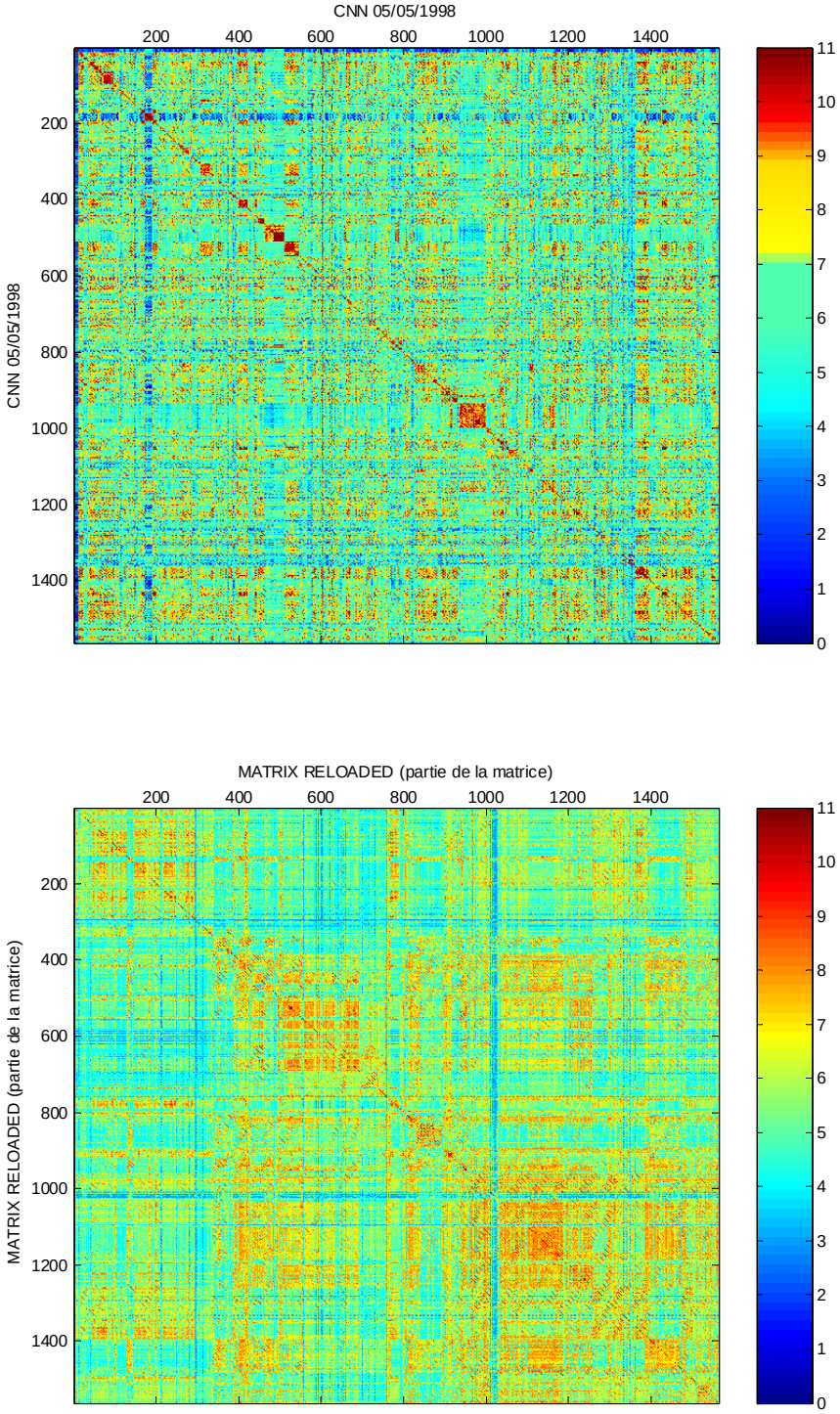


Fig. 11. Matrices d'auto comparaisons. En haut : un document vidéo avec des transitions « standard ». En bas : Partie du film MATRIX RELOADED avec des changements plus lisses en raison des procédés de postproduction.

Un retour au calme est remarquable entre les abscisses 2250 et 2500. Nous pouvons distinguer des scènes plus homogènes. Ce passage donne au spectateur l'occasion de « reprendre son souffle » avant de lui présenter la fin. Naturellement, cette fin est active, le changement de scènes est rapide, ce qui peut être à nouveau observé essentiellement entre les abscisses 2500 et 2750 de la matrice.

4.5.4 Analyse de l'effet « PostProduction »

La postproduction est l'étape qui structure les parties audio et vidéo du document, leur insuffle un rythme, en gérant en particulier l'étalonnage qui vise à assurer, ou non, le suivi visuel des plans montés, et à égaliser la structure fréquentielle de la bande son.

Cette étape est souvent planifiée dés l'écriture du document, et requiert des moyens importants. Ils ne peuvent pas être mis en œuvre pour n'importe quel type de production – en particulier pour des émissions de télévision en directe et/ou quotidienne.

La figure 11 rend compte du côté beaucoup plus « lissé » du film Matrix, qui bénéficie d'une importante postproduction, comparé à un journal télévisé dont le contenu est relativement plus chaotique.

4.5.5 Evaluation Technique

D'après le compte-rendu fourni par CALIF et présenté dans la figure 12, nous pouvons voir que cette expérimentation tournée en parallèle sur 8 processeurs a nécessité 7 heures 22 minutes de temps d'exécution. Le temps total d'exécution utilisé par tous les processus étant de 53 heures et 12 minutes. Nous pouvons en déduire, que la plus grande majorité du temps ces 8 processus travaillaient en parallèle.

```
PBS Job Id: 261.calif, Job Name: matrix reloaded
resources_used.cpupercent=51896
resources_used.cput=53:12:12
resources_used.mem=629072kb
resources_used.ncpus=8
resources_used.vmem=738704kb
resources_used.walltime=07:22:05
```

Fig. 12**.** Récapitulation de l'expérience Matrix Reloaded renvoyée par CALIF à la fin du job.

4.6 Expérience 2 : Structuration des flux de télévision

4.6.1 Macro structuration

La structuration automatique des documents a longtemps été limitée au seul travail de segmentation en plans. Basé sur les techniques de segmentation en plans, et en proposant des techniques de regroupement dans une démarche ascendante, certains travaux se sont concentrés sur la macrosegmentation ou la segmentation en séquences, dites aussi unités sémantiques.

Parmi les premiers travaux dans ce domaine, ceux de Aigrain et Joly [Aigrain 96] étaient basés sur des règles de logique exprimant des règles d'écriture cinématographique. Ils regroupent ainsi les plans pour former des macroblocs sémantiques. Après ces travaux, différentes activités de recherches ont contribué à l'évolution dans ce domaine en continuant de s'appuyer sur démarche ascendante, c'est-à-dire en se basant toujours sur le regroupement de petites unités temporelles.

Nous proposons ici d'étudier les moyens offerts par la matrice de comparaison pour appréhender des unités structurelles de grande échelle (de l'ordre du document ou de la plage horaire dans un flux de télévision). Il s'agit donc de bien de s'intéresser à une macrostructure – mais ici, la notion de macrosegment sera associée à une échelle plus grande que celle de la scène.

4.6.2 Description et but

Nous avons pu disposer d'un corpus de 7 jours continus de la télévision de la chaîne France2. Ce corpus se compose de 7 fichiers, encodés en format mpeg2, enregistrés chacun à partir de 15h28 pile sur une durée de 24heures. Ces enregistrements ont été effectués entre les 9 et 17 mai 2005. Ils comportent divers types de programme : jeux télévisés, pièces de théâtre, cérémonies, journaux d'informations, débats, télé achat, film, documentaires, séries, etc.

4.6.3 Paramétrisation de tMin et tMax

Une première question s'est posée avant tout sur l'applicabilité de la méthode. Conçue avec deux paramètres contrôlant la complexité et la précision, *tMax* et *tMin*, il convenait d'en établir une valeur acceptable pour aborder ce volume de données.

L'adaptation de *tMin* devait être importante. Les tailles des documents étaient trop grandes pour autoriser une précision très fine de la comparaison. En fait la comparaison à l'échelle de la seconde est utile pour la comparaison des plages publicitaires, des émissions du même type, voire même les programmes d'une durée de deux heures. Or, quand nous parlons de journées de télévision, la comparaison à la seconde près devient plus un obstacle qu'à atout.

De l'autre côté, en ce qui concerne le paramètre *tMax*, notre méthode de comparaison des séries chronologiques reprend le principe de l'algorithme de la plus longue sous séquence commune. L'approximation que nous avons proposée dans ce travail de recherche, *Calcul de*

Couverture, est une comparaison dichotomique basée sur les opérateurs de la morphologie. La taille du structurant est adaptée en fonction de la longueur des caractéristiques. Pour des longueurs importantes, le rapport de la taille du structurant sur la taille du segment ne dépasse pas une constante fixée (cf. chapitre 2). De ce fait, la possibilité de trouver des sous segments dont les enveloppes morphologiques se recouvrent est très élevée. Donc une longueur de segments à comparer trop grande peut générer une taille de structurant inadaptée et ainsi à une approximation de la PLSC de piètre qualité.

Un compromis entre les deux contraintes citées ci-dessus nous a ramené à adopter les deux paramètres suivants :

– $tMax = 2^{12}$ équivalent à 2min 43 secondes

– $tMin = 2^{9}$ équivalent à 20 secondes de vidéo

D'après le raisonnement ci-dessus et les premiers essais, ce paramétrage a été jugé suffisant pour détecter même l'occurrence de la météo, tout en ne sacrifiant pas la précision de notre algorithme.

4.6.4 Analyse des résultats

4.6.4.1 Analyse diagonale de la matrice

Puisque c'est une comparaison de deux journées entières, il est intéressant de comparer la composition temporelle de leurs grilles de programme (Tableau 1). Le début des enregistrements étant fixe, il peut servir de point de référence. Pour cela une analyse de l'entourage de la diagonale principale (diagonale zéro) de la matrice (figure 13) peut être menée.

Sur cette diagonale, défilent des blocs de couleurs plus ou moins homogènes. Ces blocs constituent en fait les programmes successifs de la chaîne. Aux débuts des enregistrements, c'est-à-dire vers 15H28 (00h00), ces blocs sont bien marqués. Si nous nous référons au contenu du tableau 1, nous voyons qu'il s'agit des mêmes programmes qui se répètent. Un report des informations de la grille de programme apparaît sur la matrice de comparaison. Nous pouvons vérifier que les limites des blocs coïncident avec les bornes de la grille. La majorité de ces programmes gardent leur place toute la semaine en principe, mais pas le week-end.

L'intensité des couleurs (et par suite les valeurs) des blocs est liée au degré d'homogénéité du programme. Par exemple, le bloc correspondant à un documentaire n'aura pas de fortes valeurs si le sujet traité est différent chaque jour.

En analysant la matrice, nous trouvons que les programmes qui génèrent des blocs à forte valeur sont ceux qui correspondent à des tournages en studio. Citons par exemple : Les Amours, Des chiffres et des Lettres, Tout vu tout lu, On a tout essayé.

Les séries de télévision produisent des blocs moins intenses mais toujours identifiables. Par exemple : Rex, Urgences, Amour gloire et beauté, Des jours et des vies, et Derrick.

de lundi 9 mai 05, 15H28 à mardi 10 mai 05, 15H27 Offset		Programme	de mardi 10 mai 05, 15H28 à mercredi 11 mai 05, 15H27 Offset	
00h00m-00h18	1	Derrick	1	00h00-00h15
00h24-01h10	2	Rex (série)	2	00h21-01h10
01h17-01h44	3	Des chiffres et des lettres	3	01h15-01h42
01h50h-02h25	4	Tout vu tout lu	4	01h47-02h23
02h29-03h12	5	Urgence	5	02h27-03h10
03h21-04h12	6	On a tout essayé	6	03h21-04h11
04h21-04h28	7	Houf	7	04h21-04h28
04h28-04h30	8	Météo	8	04h28-04h30
04h31-05h15	9	Le journal de 20h	9	04h30-05h12
05h30-08h25	10	Les Molières	10	xx-xx
xx-xx	11	Big Mama film	11	05h31-07h02
xx-xx	12	Comme au cinéma l'hebdo	12	07h11-07h17
xx-xx	13	Une virée à l'enfer	13	07h20-08h54
08h38-10h00	14	Huis clos, Robert Hossein	14	xx-xx
xx-xx	15	Le journal de la nuit	15	09h01-09h24
10h05-10h08	16	Météo	16	09h25-09h29
xx-xx	17	Histoires courtes	17	09h37-10h34
10h15-12h13	18	Le nouveau testament film	18	xx-xx
xx-xx	19	Pascal Sevran (musique)	19	10h35-11h25
xx-xx	20	30 millions d'amis	20	11h27-11h56
xx-xx	21	Les arts en liberté	21	11h57-12h30
12h13-12h38	22	Parole de danses	22	xx-xx
xx-xx	23	Le journal de minuit	23	12h31-12h57
12h39-12h42	24	Météo (répétition)	24	12h57-12h59
12h43-12h52	25	Sauvez angor (doc.)	25	xx-xx
xx-xx	26	Faites entrer l'accusé (doc.)	26	12h52-14h24
12h53-14h24	27	Double jeu (émis. sp. San Paolo)	27	xx-xx
14h26-14h56	28	Les amours	28	14h26-14h57
14h58-14h59	29	Point Route (6H25)	29	14h58-14h59
15h02-17h05	30	Télé matin (6H31)	30	15h02-17h00
17h12-17h34	31	Des jours et des vies	31	17h06-17h27
17h39-18h00	32	Amour, gloire et beauté	32	17h31-17h52
18h00-19h20	33	C'est au programme	33	xx-xx
xx-xx	34	Top pops	34	17h58-18h24
xx-xx	35	KD2A	35	18h30-19h20
19h30-20h00	36	Motus	36	19h30-20h00
20h06-20h38	37	Les amours	37	20h06-20h34
20h46-21h19	38	La cible	38	20h36-21h19
21h28-21h30	39	Météo	39	21h28-21h30
21h30-22h12	40	Le journal de 13H00	40	21h31-22h13
22h22-23h59	41	Derrick	41	22h24-23h59

Table 1. Grille des programmes des deux journées comparées. Tableau de référence.

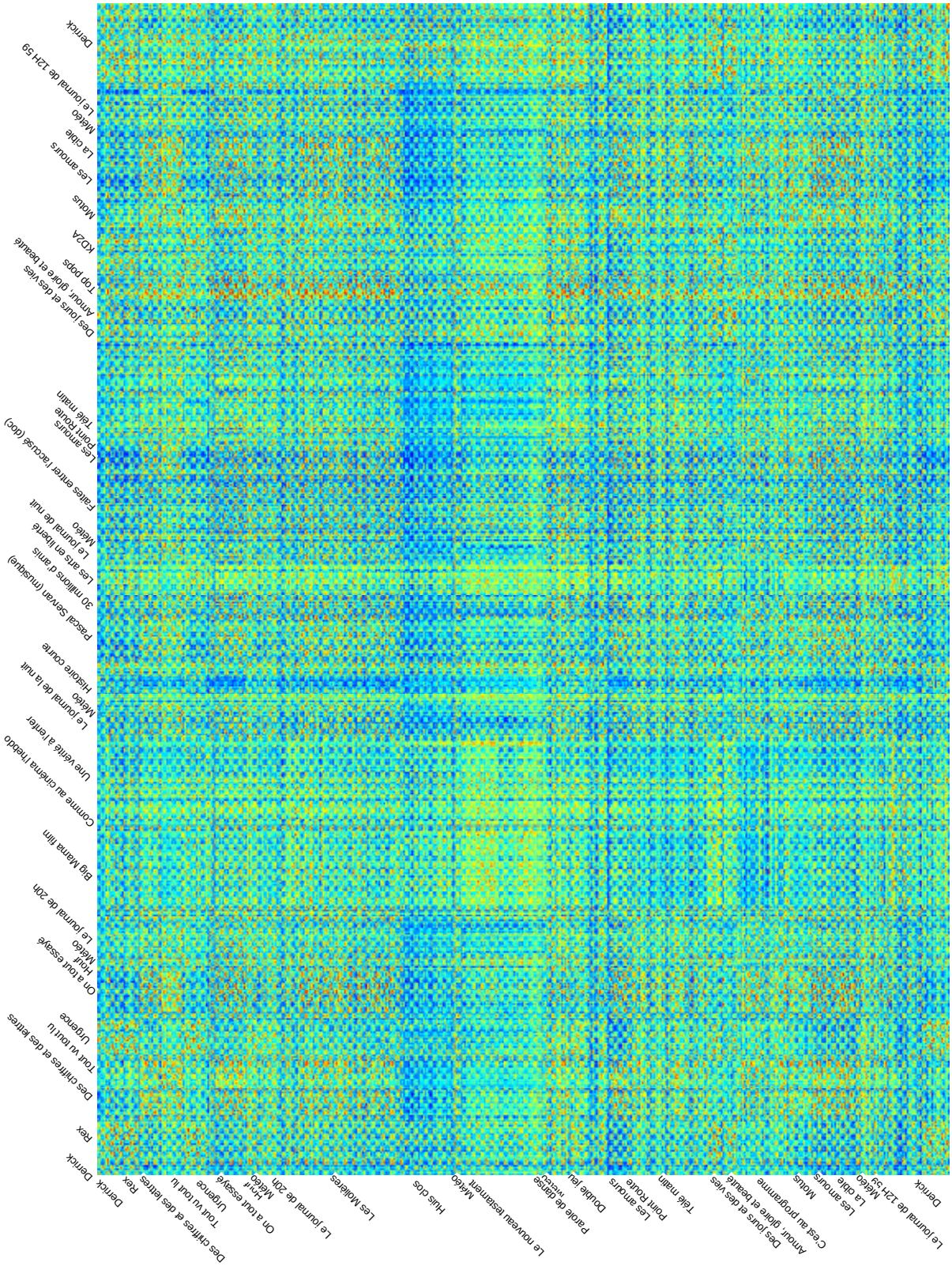


Fig. 13. Matrice de comparaison de deux journées entières de la chaîne France2.

Quant aux journaux télévisés, ils sont caractérisés par des blocs moins homogènes, présentant de petites zones clairsemées. Ce fait est dû à l'intercalage régulier entre des plans en studio où le présentateur parle et les reportages filmés sur les différents événements. Les premiers sont des passages très similaires entre eux (mouvements, couleurs dominantes, granularité, luminosité et contraste). Ils produisent alors les valeurs intenses des blocs. Alors que les seconds sont très diversifiés et génèrent la plupart du temps des valeurs basses.

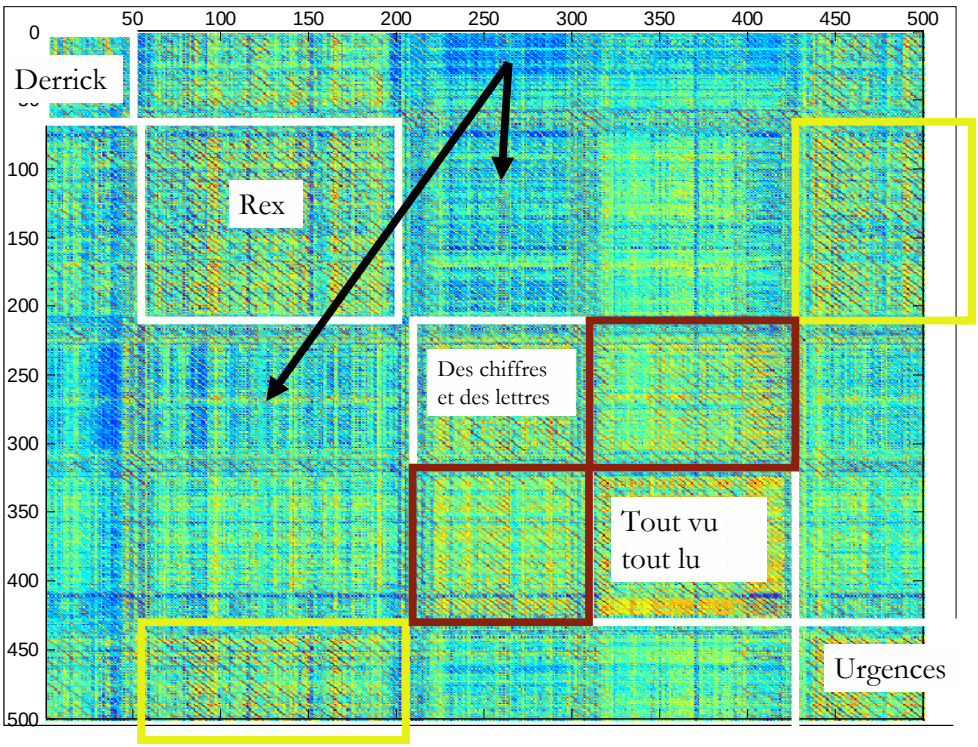


Fig. 14. Zoom avant sur une sous partie de la matrice.

Au moment du programme de la soirée, juste après le journal du soir, l'identification des blocs devient moins facile. La raison principale vient de la diversification de cette plage

horaire entre les deux jours comparés (un film d'un côté, une cérémonie de remise de prix de l'autre).

Lors de l'identification des blocs, nous avons constaté que certains programmes commencent exactement à des heures bien précises. C'est le cas, entre autres, de Point Route, certaines diffusion de la Météo, Télé Matin, le Journal, sauf le journal de nuit et du soir, qui n'apparaissent pas sur une des deux soirées, les Amours et Derrick. Par analogie, on peut voir que certains programmes ont une taille fixe. Parmi ces programmes, citons Houf, Météo, Point Route, Les Amours, et certaines séries télévisées.

4.6.4.2 Analyse globale

Les blocs de similarité ne se trouvent pas seulement sur la diagonale. La matrice comporte l'information de comparaison de chaque couple de programmes dans les deux journées comparées. Pour étudier la similarité de deux programmes en particulier, il suffit d'extraire le bloc correspondant. Ce bloc est la matrice de similarité des programmes comparés. Ainsi dans cette matrice, la rediffusion des Amours est repérable.

Nous remarquons aussi que les programmes fortement typés, en particuliers ceux filmés en studio, présentent une forte similarité les uns envers les autres (voir par exemple, le zoom avant dans la figure 14). Nous pensons, après vérification, que c'est en relation avec le fait que ces programmes sont diffusésss à l'intention d'un même public, donc en utilisant plus ou moins le même type d'habillage et des procédés de réalisation et de production identiques. Ce type de programme se caractérise également par un usage important des plans rapprochés montrant les animateurs ou les participants. Sur cette même figure, nous pouvons constater des zones de dissimilarité, entre des programmes de genres différents, ici des séries télévisées et des jeux télévisés. Ces zones sont désignées par des flèches.

4.6.5 Evaluation Technique

```
PBS Job Id: 275.calif Job Name: FRANCE2
Execution terminated Exit_status=0
resources_used.cpupercent=368585
resources_used.cput=382:03:31
resources_used.mem=2548432kb
resources_used.ncpus=8
resources_used.vmem=2676048kb
resources_used.walltime=62:08:31
```

Fig. 15**.** Récapitulation de l'expérience FRANCE2 envoyée par CALIF à la fin du job.

Cette expérience, la plus importante de toutes celles que nous avons mené en matière de taille de document vidéo traité, et de temps de calcul, démontre avant tout l'applicabilité de notre méthode sur de petits comme sur de grands documents.

4.7 Expérience 3 : Classification en genre et étude de la composition temporelle des journaux télévisés

4.7.1 But et description

Nous fixons comme but pour cette expérience d'étudier la capacité de la mesure de similarité définie dans le chapitre 3 à discriminer l'appartenance de tous ces documents à un ou plusieurs genres et à une ou plusieurs collections (ou sous genres). Un deuxième but sera d'illustrer la différence des scénarios de pondération pour définir la mesure de similarité. Nous voulons enfin souligner la différence entre la mesure et la distance de similarité définies dans le chapitre 3.

Nous disposons d'une collection de journaux télévisés, issue du corpus de TRECVIDEO 2003, enregistrés à partir des deux chaînes d'informations ABC et CNN pendant le mois de mai 1998, chacun d'une durée allant de 28 à 32 minutes.

Un sous-ensemble de journaux servant de corpus d'expérimentation a été défini. Cet ensemble se compose de :

- 9 journaux CNN numérotés dans cette section de **1** à **9**,
- 1 journal ABC dont le numéro le désignant serait **11**, jouant le rôle de l'intrus dans la collection CNN,
- 1 document de jeu télévisé, « Pyramide », hors du corpus TRECVIDEO, jouant le rôle de l'intrus vis-à-vis du genre des documents précédents (**journaux télévisés**) portant le numéro 10.

Pour commencer cette expérience, nous supposons que nous ne disposons pas des connaissances *a priori* citées ci-dessus, c'est à dire que nous ne savons pas qu'il y a une notion de genre, de collection et d'intrus.

L'ensemble choisi pour l'expérience est donc :

$$ENS = \{\text{CNN1 (1), CNN2 (2), CNN3 (3), CNN4 (4), CNN5 (5), CNN6 (6), CNN7 (7), CNN8 (8), CNN9 (9), Pyramide (10), et ABC (11)}\}. \quad (11)$$

Nous avons mesuré la similarité de chaque membre de l'ensemble des documents choisis pour cette expérience à tous les autres et à lui-même. Soit :

```
Début

   Etant donné un scénario : S ;
   Etant donnés deux sous ensembles construits à partir d'ENS : ENS1 et
ENS2 ;

   Pour chaque document D1 de l'ensemble ENS1 faire

            Mesure_cumulée = 0 ;
```

```
            Distance_cumulée = 0 ;

            Pour chaque document D2 de l'ensemble ENS2 faire

                Mesure_cumulée + = Mesure (D1, D2, S);
                Distance_cumulée + = Distance (D1, D2, S);

            Fin pour

    Fin pour
```

Fin

4.7.2 Définition de l'expérience

Dans le chapitre 3, nous avons parlé de différents scénarios pour mesurer la ressemblance entre documents vidéo. Ces scénarios ont pour but de pondérer les points sur (et autour de) une diagonale choisie pour mettre en évidence l'ordre temporel des éléments communs entre les deux documents. Nous avons ainsi défini quatre scénarios. Pour chacun, nous exécutons l'algorithme cité plus haut.

Nous avons proposé dans le chapitre précédent d'associer à un scénario de pondération l'équation :

$$x_\varphi = k \pm t_\varphi,\ \varphi = 1, 2, 3 \text{ et } 4. \qquad (\mathbf{12})$$

Il faut donc définir k, x_1, x_2, x_3 et x_4 pour pouvoir définir un scénario quelconque S.

4.7.2.1 La recherche de la diagonale de référence

Tous les enregistrements commencent au début des journaux CNN et ABC ou du programme de jeu Pyramide. Il implique que la diagonale k de pondération n'est pas trop loin de la diagonale zéro des matrices de comparaison. Or, à la différence de l'expérience précédente, les débuts des enregistrements ne sont pas synchronisés exactement, d'où la nécessité de rechercher la diagonale de référence. Bien entendu, c'est la diagonale qui est supposée aligner le plus parfaitement possible deux documents lorsqu'ils sont comparables et dans leur éléments et dans l'ordre temporel de l'occurrence de ces éléments.

Supposons que le scénario S soit défini par x_1, x_2, x_3 et x_4 relativement à une diagonale k à déterminer. Afin de trouver le meilleur alignement entre les documents, nous avons calculé pour chaque matrice, toutes les mesures de similarité correspondant à chaque diagonale de la matrice (de dimension $dim \times dim$) d'abscisse comprise dans l'intervalle Diag.

$$Diag = [(1\text{-}dim)/20, (dim\text{-}1)/20]. \qquad (\mathbf{13})$$

Le meilleur alignement est obtenu lorsque la mesure de similarité est la plus grande. La recherche de k autour de la première diagonale se légitime par le fait que les enregistrements commencent tous par le début.

4.7.2.2 La définition des scénarios

Nous définissons maintenant différentes valeurs de x_1, x_2, x_3 et x_4 illustrant différents scénarios, qui permettront de déterminer k, et par suite de produire les valeurs des mesures de similarité.

4.7.2.2.1 Scénario 1

Le scénario 1 représente le paradigme du décalage de temps constant.

$$x_1 = x_2 = x_3 = x_4 = k. \quad (\mathbf{14})$$

4.7.2.2.2 Scénario 2

Le scénario 2 correspond au synchronisme symétrique, où l'intervalle $[x_1, x_4]$ couvre 50% des diagonales, autour de k. Soit :

$$[x_1, x_4] = 0.5 \times (2 \times dim - 1), \quad (\mathbf{15})$$

Et toujours :

$$x_2 = x_3 = k. \quad (\mathbf{16})$$

4.7.2.2.3 Scénario 3

Le scénario 3 est également un synchronisme symétrique, où $[x_1, x_4]$ couvre 60% des diagonales dont 20% appartiennent à l'intervalle $[x_2, x_3]$.

$$[x_1, x_4] = 0.5 \times (2 \times dim - 1) ; \quad (\mathbf{17})$$

$$[x_2, x_3] = 0.2 \times (2 \times dim - 1) ; \quad (\mathbf{18})$$

4.7.2.2.4 Scénario 4

Enfin, le dernier scénario défini pour cette expérimentation, représente le cas spécial du décalage variable de temps où toutes les diagonales ont la même importance, c'est à dire lorsque $[x2, x3]$ couvre 100% de la matrice.

$$x_1 = - dim, x2 = - dim + 1, x3 = dim - 1, \text{ et } x4 = dim. \quad (\mathbf{19})$$

4.7.3 Résultats

4.7.3.1 Illustrations graphiques

Nous observons les résultats de l'expérience 3 reportés sur les graphiques de la figure 19. Dans cette figure,

- les graphiques A et B représentent les résultats, pour les scénario 1 à 4, pour la mesure de similarité cumulée décrite plus haut et appliqué à l'ensemble des documents *ENS*.

$$ENS_1 = ENS_2 = ENS. \quad (\mathbf{20})$$

- Les graphiques C et D, représentent la mesure de similarité cumulée, pour chacun des quatre scénarios, sur les ensembles débarrassés des documents identifiés comme intrus d'après les graphiques A et B.

$$ENS_1 = ENS_2 = \{\text{CNN1 (1), CNN2 (2), CNN3 (3), CNN4 (4), CNN5 (5), CNN6 (6), CNN7 (7), CNN8 (8), et CNN9 (9)}\}. \quad (\mathbf{21})$$

- Les graphiques E et F donnent les résultats de la distance cumulée. Avec :

$$ENS_1 = \{\text{CNN1 (1), CNN2 (2), CNN3 (3), CNN4 (4), CNN5 (5), CNN6 (6), CNN7 (7), CNN8 (8), et CNN9 (9)}\}, \quad (\mathbf{22})$$

$$ENS_2 = ENS. \quad (\mathbf{23})$$

- Les graphiques G et H ne sont en fait qu'un zoom sur E et F après élimination des données concernant Pyramide (10) et ABC (11), et ceci pour pouvoir mieux visualiser les différences entre les documents CNN (1 à 9).

- Notons que, les valeurs sur l'axe des coordonnées des graphiques n'ont aucune signification en elles-mêmes. Nous avons déjà évoqué la relativité de la notion de similarité dans les chapitres 1 et 3. De ce fait, nous nous intéressons plus à la comparaison des scores obtenus par chaque document de l'ensemble ENS1, entre eux, qu'aux scores eux-mêmes.

- Finalement rappelons que dans le cas du scénario 1, où seuls les éléments de la diagonale de référence sont invoqués, cette diagonale est la diagonale zéro dans le cas d'une auto comparaison. Il s'en suit que les mesures sont alors saturées.

$$Mesure\ (D_i, D_i, \text{Scénario 1}) = 1, \forall\ i \in ENS_1, \quad (\mathbf{24})$$

et la même manière, les distances sont nulles :

$$Distance\ (D_i, D_i, \text{Scénario 1}) = 0, \ \forall\ i \in ENS_1, \quad (\mathbf{25})$$

Pour la visualisation des graphiques, nous avons donc choisi de séparer la représentation des résultats issue du scénario 1 (graphiques A, C, E et G), de celles issues des scénarios 2,3 et 4 (graphiques B, D, F et H).

4.7.3.2 Interprétations des graphiques

Interprétation du graphique ***A*** :

1. Le scénario 1 ne concernant que les éléments de la diagonale de pondération, la composition temporelle est fortement prise en compte dans les mesures de similarité calculées.

2. Comme c'est une somme cumulée des mesures de similarités comportant forcément une mesure ayant une valeur de 1 (grâce à l'auto comparaison), nous devons comparer les comportements des barres correspondant aux documents au dessus de la valeur 1.

3. Ainsi, nous voyons bien que le document Pyramide (10), est le premier intrus aux ensembles ENS_1 et ENS_2.

4. De la même manière, le document ABC (11) se distingue. Mais cette distinction n'est pas aussi claire que celle de Pyramide. Il apparaît que ces documents du même genre (journal ABC avec la collection) comportent des éléments comparables.

Interprétation du graphique ***B*** :

5. Les scénarios 2, 3, et 4 prennent en compte les éléments communs dans une bande autour de la diagonale de pondération *k*. Dans le cas du scénario 4, cette bande couvre toutes les diagonales de la matrice. Donc on peut dire que les éléments semblables dans les documents comparés sont comptés même s'ils n'apparaissent pas dans le même ordre temporel.

6. Là aussi, nous pouvons voir que les deux documents intrus, Pyramide (10) et ABC (11) sont repérables. Nous en concluons que le style n'est pas le même.

7. Bien entendu, la taille de l'ensemble étudié ne permet pas de conclure avec certitude l'appartenance des documents CNN (1 à 9) à une même collection. Mais une expérience sur un ensemble de plus grande taille ne pourrait que le confirmer. En fait, la présence d'un document « spécial » qu'est le document 8, pénalise cette conclusion. Avec une grande cardinalité de ENS_2 cette exception est lissée par les autres documents.

Interprétation du graphique ***C*** et ***D*** :

8. Comme nous l'avons précisé, les graphiques C et D concernent les mesures de similarité entre les documents CNN seulement. Autrement dit, il s'agit de voir selon les scénarios, lequel, parmi les documents représentés, pourrait être le plus proche de la collection CNN (des 9 documents) en entier.

9. Ces graphiques sont générés pour illustrer la différence entre les différents scénarios.

10. Considérons par exemple les documents CNN1 (1) et CNN7 (7) dans le graphique D.

    - Pour le scénario 2 : le document CNN7 a un score plus haut que CNN1, il est de ce fait plus proche de la collection,

    - Tandis qu'après considération du scénario 4, nous trouvons que le document CNN1 a un score plus élevé que le document CNN7, et cette fois-ci c'est CNN1 qu'est plus proche de la collection.

    - Donc tout dépend des préférences prises en compte par la mesure de comparaison. Comme le scénario 2 privilégie la zone autour de la diagonale *k* seulement, alors que le scénario 4 prend toute la matrice sans distinction, il apparaît que le document CNN1 respecte mieux la composition temporelle de la collection. A l'inverse, le document CNN7, comporte plus d'éléments similaires, mais placés un peu dans le désordre. Ce désordre n'est pas pénalisant dans le cas du scénario 4. De ce fait, on décidera ici que CNN7 est plus conforme au style de production de CNN, que CNN1.

11. Selon le même raisonnement, nous pouvons déterminer que CNN5 (5) est le plus proche à la collection d'un point de vue global, c'est-à-dire, en style et en composition temporelle. Il est élu alors représentant de la collection.

12. Quand nous avons examiné le document 8, pour connaître les raisons de son éloignement relatif à la collection, nous avons remarqué qu'il contenait un reportage sur un défilé de mode représentant une grande partie des informations, le rendant légèrement différent du reste de la collection.

Tous les graphiques discutés ci-dessus concernent des mesures de similarité selon différents scénarios. Jusqu'à présent, un décalage dans l'alignement des deux documents comparés n'engendrait pas une pénalisation pour la mesure. La distance que nous avons définie prend en compte ce décalage et le pénalise. C'est en fait la différence essentielle qui doit être observée dans les graphiques E, F G et H par rapport aux graphiques A, B, C et D. Nous remarquons également que la mesure de similarité opère une meilleure distinction entre les membres de la collection CNN et ceux ne lui appartenant pas.

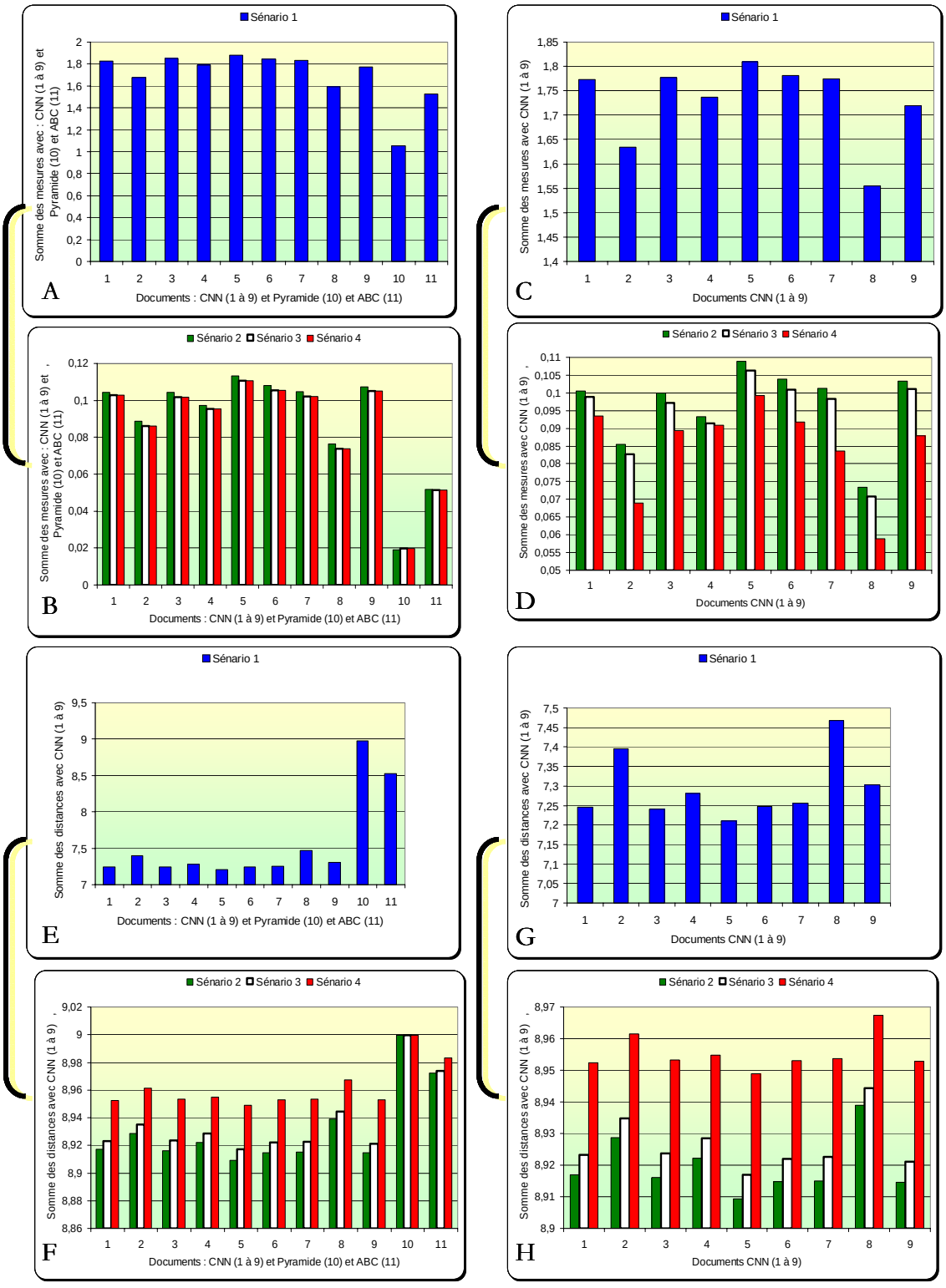


Fig. 16. Représentation des différentes mesures et distances cumulées.

4.7.4 Extraction automatique de l'organisation structurelle des enregistrements

Nous avons employé les distances de similarité, calculées pour déterminer le document le plus représentatif de sa propre collection. Sur la collection de journaux CNN que nous avons utilisée, il s'agit du document CNN5. En utilisant ce document, nous calculons l' « histogramme des invariants de production », produit par comparaison du document choisi avec chaque document de la collection.

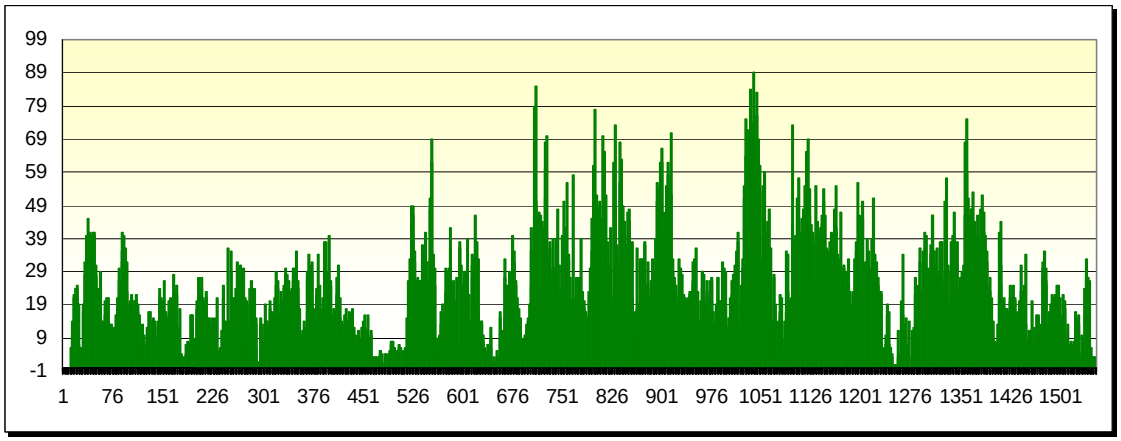


Fig. 17. L'histogramme des invariants de production basé sur le document CNN5.

Pour le construire, nous additionnons les matrices de similarités produites, avec CNN5 comme premier document comparé et CNN1 à CNN9 comme second document successivement. Ensuite nous projetons toute la matrice sur l'axe du document CNN5. L'histogramme obtenu est donné dans la figure 17.

Afin d'étudier la structure de la collection, c.-à-d. l'ordre temporel de ces invariants de production, nous avons seulement tenu compte des points sur les bandes centrales des diagonales, d'une largeur de 10% de toutes les diagonales. Un seuil, dépendant de la taille de la collection et de son caractère plus ou moins structuré, appliqué sur cet histogramme, permet d'identifier les extraits les plus révélateurs du contenu récurrent de cette collection. Le choix de ce seuil sera discuté dans la conclusion. L'histogramme est ici seuillé à 40, et les segments qui dépassent cette valeur sont considérés comme représentant de l'organisation structurelle des documents de la collection.

Les invariants de production principaux de cette collection de journaux télévisés identifient des segments classiquement repérés par les outils d'analyse dédiés : le présentateur, les informations boursières, la météo, et quelques publicités partagées par l'ensemble des journaux. Les films publicitaires apparaissent comme invariants à la collection probablement parce que tous les documents de CNN ont été enregistrés dans la même période et présentent de ce fait les mêmes annonces (figure 18).

L'ensemble de ces invariants de production et leur ordre temporel peuvent alors être employés pour établir automatiquement un prototype de la structure d'un programme de journal de CNN TV.

Fig. 18. Les images clés extraites du document CNN5 représentatif de sa collection.

4.7.5 Evaluation Technique

En ce qui concerne le paramétrage utilisé dans cette expérience. Nous avons initialisé :

– *tMax* à 2^{10} ou 41 secondes, et

– *tMin* à 2^5 ou 1,28 secondes

Il faut noter que tous ces graphiques sont calculés à partir du même ensemble de matrices de comparaison. Une fois la matrice de comparaison calculée, les traitements suivants prennent peu de temps (changement de scénario, d'ensemble choisi, etc). Différents scénarios, alignements et mesures peuvent être évalués dans un temps très réduit (de l'ordre de la seconde).

Cette expérience, repose sur une fusion des caractéristiques très stricte. La matrice de comparaison associe toutes les mesures extraites dans la définition des invariants de production.

La mise en œuvre du calcul parallèle permet de réduire les 30 minutes de calcul requises pour comparer deux des documents audiovisuels cités au-dessus sur un Pentium 4, 2.6GHz avec 512Mo de Ram, à moins de 3 minutes sur CALIF.

4.8 Conclusion

Dans ce chapitre d'applications nous avons défini des mécanismes d'évaluation qualitative des outils de comparaison que nous proposons. Les résultats présentés ne peuvent pas être évalués quantitativement. Ce sont plutôt des propositions de voies pour des utilisations potentielles des outils.

Comme perspectives d'investissements des résultats présentés, nous pourrons appliquer la mesure de similarité de style la plus élémentaire aux différents sous blocs constituant la similarité entre les programmes pour identifier davantage leur forte similarité.

Dans la perspective d'une analyse automatique de l'organisation structurelle d'un document d'une collection, les directives visant à identifier les éléments de structure doivent encore être formalisées.

Cette considération mène à tenir compte de deux points :

- les collections produites selon des contraintes fortes d'ordonnancement temporel (loterie nationale, quelques jeux de TV) produisent des histogrammes d'invariants denses présentant des valeurs élevées. Ces histogrammes ne peuvent pas être analysés de la même manière que ceux obtenus avec des programmes de journaux télévisés par exemple. Le choix du seuil doit être encore étudié.

- les documents ayant une structure qui ne peut pas être détectée sur la diagonale principale (la majeure partie des programmes de sport par exemple) exigeront une analyse différente de la matrice de comparaison pour détecter automatiquement des éléments de structure.

5 Chapitre 5 : Conclusion générale

Chapitre 5

CONCLUSION GENERALE

Ce travail de recherche nous a conduit à élargir notre champ de connaissances. A côté des méthodes et des travaux spécialisés sur l'analyse vidéo, nous avons abordé certains des problèmes de la communauté des séries chronologiques. Cette communauté s'intéresse à son tour au domaine du multimédia qui représente une source importante de séries numériques et de problèmes corollaires.

Nous avons formulés des propositions au sujet de la comparaison des documents audiovisuels, et indirectement de la comparaison de séries chronologiques. Ces contributions sont résumées dans le paragraphe suivant. Nous présentons à la suite de ce paragraphe les conclusions majeures que nous pouvons tirer de notre expérience sur ce sujet et nous concluons ce manuscrit par la présentation de différentes perspectives pour de futures activités de recherche.

5.1 Contributions

Nous avons mené des réflexions portant aussi bien sur la signification de la similarité et de la dissimilarité entre les documents que sur la procédure à suivre pour la comparaison des contenus audiovisuels.

En ce qui concerne la méthode de comparaison, les points majeurs que nous avons évoqués sont :

- La conception et l'implémentation d'un algorithme général qui effectue la recherche, la vérification et l'extraction de toutes les séquences similaires de tailles variables à partir de deux séries chronologiques.

- La conception, l'implémentation et l'évaluation d'un algorithme de calcul de ressemblance de deux séquences numériques. Cet algorithme a été comparé avec l'algorithme de la «Plus longue Sous Séquence Commune ».

- La conception d'une procédure de comparaison globale, adaptée à différentes échelles de documents vidéo (allant de une minute jusqu'à vingt quatre heures d'enregistrement), à tout type de caractéristiques audiovisuelles, et à n'importe quel genre de documents comparés.

- Un schéma original représentant le résultat de la comparaison : la matrice de similarité. Ce schéma permet de produire une estimation de la ressemblance.

En ce qui concerne la mesure de similarité, nous avons introduit les notions suivantes :

- la notion de similarité de style, qui gère aussi bien la similarité totale (identification des copies d'un même document), que la similarité partielle (quantitative, qualitative ou temporelle).

- les scénarios d'alignement par pondération des décalages/synchronisations des segments semblables entre deux documents, prenant en compte :

    - le degré de ressemblance, et

    - l'organisation temporelle des éléments semblables.

En ce qui concerne les domaines d'application potentiels de la comparaison des documents audiovisuels, nous nous sommes essentiellement intéressés à :

- l'étude du style d'un film de cinéma

- l'étude de collections de documents vidéo, notamment de journaux télévisés

- l'étude de flux télévisés

5.2 Perspectives

Pour conclure nous proposons un ensemble de pistes inspirées par ce travail qui pourraient faire l'objet d'approfondissements.

En ce qui concerne la fusion des caractéristiques. Nous pouvons étudier la sélection automatique de l'ensemble de caractéristiques participant à l'évaluation finale de la similarité de sorte à élever l'influence des caractéristiques qui fournissent des informations qui se recoupent et atténuer l'effet de celles qui pénalisent l'extraction et l'identification des invariants de production. Nous pouvons imaginer attribuer des indices de confiances à chaque caractéristique selon des critères automatiques à définir.

En ce qui concerne la matrice de comparaison générée et vu la richesse de l'information qu'elle peut contenir et sa dimension parfois immense, il nous sera indispensable dans nos travaux proches de lui consacrer une étude élargie. Nous étudions la possibilité de proposer des mécanismes de lecture automatique de cette matrice, en fonction d'une application donnée et de définir des seuils de filtrage adaptés. Cette étude permettra d'évaluer la comparaison en vue de l'identification de programmes répétés sur une base de données de documents indexés manuellement pour obtenir des résultats concernant sont exploitation possible pour la recherche d'information.

Une fois les éléments similaires extraits, un des axes de recherche à considérer serait proposer des mécanismes d'indexation de ces segments et de classification selon des ontologies ou des metadonnées dépendant du domaine d'application.

Parallèlement, nous pouvons explorer les concepts proposés dans cette thèse, notamment l'algorithme générique IQR afin d'observer le genre de résultats qu'il peut fournir lorsqu'il est appliqué sur la dimension spatiale de l'image en l'association avec la dimension temporelle et ceci pour l'étude des comportements des objets dans la vidéo.

Nous sommes enfin intéressés par l'adaptation de cet algorithme afin de traiter des flux en continu en temps réel parallèlement à la diffusion.

BIBLIOGRAPHIE

[Adjeroh 98] D. Adjeroh, I. King, M.C. Lee. A distance measure for video sequence similarity matching. Proceedings International Workshop on Multi-Media Database Management Systems, Dayton, OH, USA, pp. 72-9, Aug. 1998.

[Agarwal 00] P. K. Agarwal, L. Arge, J. Erickson. Indexing moving points. Proc. of the 19th ACM Symp. on Principles of Database Systems (PODS), pages 175–186, 2000.

[Agrawal 93] R. Agrawal, C. Faloutsos, A. Swami. Efficient similarity search in sequence databases. Proc. of the Fourth International Conference on Foundations of Data Organization and Algorithms, Chicago, October 1993. Lecture Notes in Computer Science 730, Springer Verlag, 1993, 69-84.

[Agrawal 93b] R. Agrawal, T. Imielinski, A. Swami. Database mining: A performance perspective. IEEE Transactions on Knowledge and Data Engineering, 5(6):914-925, December 1993. Special Issue on Learning and Discovery in Knowledge-Based Databases.

[Agrawal 95] R. Agrawal, K. Lin, H. S. Sawhney, K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. Proceedings of the 21st VLDB Conference, pages 490-501, Zurich, Switzerland, 1995.

[Aigrain 95] Ph Aigrain, Ph Joly, V. Longueville. Medium Knowledge-Based Macro-Segmentation of Video into Sequences. In M. Maybury (Ed.) (pp. 5-16), IJCAl 95 - Workshop on Intelligent Multimedia Information Retrieval. Montreal, August 19, 1995.

[André-Obrecht 88] R. André-Obrecht. A New Statistical Approach for Automatic Speech Segmentation. IEEE Transactions on Audio, Speech, and Signal Processing, 36(1): 29–40, janvier 1988.

[Antani 02] S. Antani, R. Kasturi, R. Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. Pattern Recognition 35(4), pp. 945–965, 2002.

[Beckmann 90] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. Proc. of ACM SIGMOD, pages 322-331, Atlantic City, NJ, May 1990.

[Berchtold 98] S. Berchtold, C. Böhm, H.-P. Kriegel. The Pyramid-Technique: Towards Breaking the Curse of Dimensionality. Proc. Int. Conf. on Management of Data, ACM SIGMOD, Seattle, Washington, 1998.

[Berndt 94] D. J. Berndt, J. Clifford. Using dynamic time warping to find patterns in time series. KDD-94: AAAI Workshop on Knowledge Discovery in Databases, pages 359-370, Seattle, Washington, July 1994.

[Bhat 98] D. Bhat, S. Nayar. Ordinal measures for image correspondence. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 Issue: 4, pp. 415V423, April 1998.

[Bollobás 97] B. Bollobás, G. Das, D. Gunopulos, H. Mannila. Time-Series Similarity Problems and Well-Separated Geometric Sets. Proceedings of the 13th Annual Symposium on Computational Geometry (Nice, France, June 4–6, 1997), J.-D. Boissonnat, Ed. ACM Press, New York, NY, 454–456.

[Bozkaya 97] T. Bozkaya, N. Yazdani, M. Ozsoyoglu. Matching and Indexing Sequences of Different Lengths. Proc.of the CIKM, Las Vegas, 1997.

[Brin 95] S. Brin. Near Neighbor Search in Large Metric Spaces. VLDB Conf., 1995.

[Broder 97] A. Broder, S. Glassman, M. Manasse, G. Zweig. Syntactic clustering of the web, in Proc. 6th Int. World Wide Web Conf., vol. 29, no. 8–13, Computer Networks and ISDN Systems, Sept. 1997, pp. 1157–1166.

[Bruno 03] E. Bruno, S. Marchand Maillet. Prédiction Temporelle de Descripteurs Visuels pour la Mesure de Similarité entre Vidéos. Proceedings of the GRETSI'03, Paris, France, September 2003.

[CALIF] http://intranet.irit.fr/Systeme/Serveurs/calif.shtml

[Califano 93] A. Califano, I. Rigoutsos. FLASH: A fast look-up algorithm for string homology. Proc. of the 1st International Conference on Intelligent Systems for Molecular Biology, pages 353-359, Bethesda, MD, July 1993.

[Califano 94] A. Califano, R. Mohan. Multidimensional indexing for recognizing visual shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(4):373-392, 1994.

[Calliope 89] Calliope. La parole et son traitement automatique. Masson, Paris, France, 1989.

[calliope] calliope.gs.washington.edu/software/bonsaiWebDocs/Glossary.html

[Castelli 01] V. Castelli, L. D. Bergman. Image Databases: Search and Retrieval of Digital Imagery. John Wiley & Sons, 2001.

[Chan 99] K.-P. Chan, A. W.-C. Fu. Efficient time series matching by wavelets. ICDE, 1999.

[Chandra 00] R. Chandra & al. Parallel Programming in OpenMP. éd. Morgan Kaufmann Publishers, oct. 2000. Premier livre sur OpenMP.

[Chang 99] H. Chang, S. Sull, S. Lee. Efficient video indexing scheme for content-based retrieval, IEEE Trans. Circuits Syst. Video Technol., vol. 9, pp. 1269–1279, Dec. 1999.

[Chen 04] M. Chen, A. Hauptmann. Multi-modal classification in digital news libraries. ICASSP 2004. Montreal, Canada, May 17-21. 2004.

[Chergui 04] http://www.idris.fr/data/cours/parallel/openmp/OpenMP_cours.pdf

[Cheung 00] S. Cheung, A. Zakhor. Estimation of Web Video Multiplicity. Proceedings of the SPIE - Internet Imaging, vol. 3964, pp. 34-46. San Jose, California. January 22-28, 2000.

[Cheung 02] S.-C. Cheung, A. Zakhor. Efficient video similarity measurement with video signature. Proceedings of the 9th IEEE International Conference on Image Processing, vol. 1, pp. 621-624, Sept. 2002.

[Chu 99] K. Chu, M. Wong. Fast Time-Series Searching with Scaling and Shifting. ACM Principles of Database Systems, Philedelphia, PA, pages 237–248, June 1999.

[Ciaccia 97] P. Ciaccia, M. Patella, P. Zezula. M-tree: An efficient access method for similarity search in metric spaces, in VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, Athens, Greece. 1997, pp. 426-435.

[Conan 03] V. Conan, I. Ferrané, P. Joly, C. Vasserot. KLIMT: Intermediations Technologies and Multimedia Indexing. Third International Workshop on Content-Based Multimedia Indexing (CBMI'03), Rennes, France, September 2003. INRIA, 11-18.

[Cormen 90] T.H. Cormen, C.E. Leiserson, R.L. Rivest. Introduction to Algorithms. MIT Press, 1990.

[Courtney 97] J. D. Courtney. Automatic video indexing via object motion analysis. Pattern Recognition, vol. 30, no. 4, pp. 607-626, April 1997.

[cps] web.cps.msu.edu/rlr/terms.html

[Crochemore 94] M. Crochemore, W. Rytter. Text Algorithms. Oxford Univ. Press, 1994.

[Dagtas 00] S. Dagtas, W. Al-Khatip, A. Ghafoor, R. L. Kashyap. odels for motion-based video indexing and retrieval. IEEE Trans. Image Proc., vol. 9, no. 1, pp. 88-101, Jan. 2000.

[Das 97] G. Das, D. Gunopulos, H. Mannila, Finding Similar Time Series, In proceedings of Principles of Data Mining and Knowledge Discovery, 1st European Symposium. Trondheim, Norway, Jun 24-27. 1977. pp 88-100.

[Deng 98] Y. Deng B. S. Manjunath. NeTra-V: Toward and object-based video representation. IEEE Trans. Circ. Syst. for Video Tech., vol. 8, no. 5, pp. 616-627, 1998.

[Dimitrova 02] N. Dimitrova, H. Zhang, B. Shahraray, I. Sezan, T, Huang, A. Zakhor, Applications of video content analysis and retrieval, IEEE Multimedia, vol. 9, pp. 42-55, July-Sept. 2002.

[Duong] Source : http://nduong.free.fr/

[Ekin 03] A. Ekin, A. M. Tekalp, R. Mehrotra. Automatic soccer video analysis and summarization. IEEE Trans. on Image Proc., vol. 12, no. 7, pp. 796-807, July 2003.

[Erickson 83] B. W. Erickson, P. H. Sellers. Recognition of patterns in genetic sequences. D. Sanko and J. B. Kruskal, editors, Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Addison Wesley, MA, 1983.

[Faloutsos 94] C. Faloutsos, M. Ranganathan, Y. Manolopoulos. Fast subsequence matching in time-series databases. ACM SIGMOD, pages 419–429, Minneapolis MN, May 1994.

[Faloutsos 95] C. Faloutsos, K.-I. Lin. FastMap: A fast algorithm for indexing, datamining and visualization of traditional and multimedia datasets. Proc. ACM SIGMOD, pages 163–174, May 1995.

[Faloutsos 97] C. Faloutsos, H. Jagadish, A. Mendelzon, T. Milo. Signature technique for similarity-based queries. In SEQUENCES97, 1997.

[Flickner 95] M. Flickner et al. Query by Image and Video Content: The QBIC System, IEEE Computer, Vol. 28, No. 9, pp. 23-32, 1995.

[Foote 01] J. Foote, M. Cooper. Visualizing Musical Structure and Rhythm via Self-

Similarity. Proceedings of the 2001 International Computer Music Conference, (Havana, Cuba, 2001), International Computer Music Association, 419-422.

[Foresti 02] G. L. Foresti, L. Marcenaro, C. S. Regazzoni. Automatic detection and indexing of video-event shots for surveillance applications. IEEE Trans. Multimedia, vol. 4, no. 4, pp. 459-471, Dec. 2002.

[Fukunaga 72] K. Fukunaga. Introduction to Statistical Pattern Recognition. Academic Press, New York, 1972.

[Gaffney 99] S. Gaffney, P. Smyth. Trajectory Clustering with Mixtures of Regression Models. Proc. of the 5th ACM SIGKDD, San Diego, CA, pages 63–72, Aug. 1999.

[Gallliano 05] S. Gallliano, E. Geoffrois, D. Mostefa, K. Choukrii, J.-F. Bonastre, G. Gravier. The ESTER Phase II Evaluation Campaign for the Rich Transcription of French Broadcasts News. Interspeech'2005 - Eurospeech — 9th European Conference on Speech Communication and Technology, Lisboa, 4-8 septembre 2005.

[Gargi 00] U. Gargi, R. Kasturi, S. H. Strayer. Performance characterization of video-shot change detection methods. IEEE Trans. Circ. Syst. for Video Tech., vol. 10, pp. 1-13, Feb. 2000.

[Ge 00] X. Ge, P. Smyth. Deformable markov model templates for time-series pattern matching. Proc ACM SIGKDD, 2000.

[Gionis 99] A. Gionis, P. Indyk, R. Motwani. Similarity search in high dimensions via hashing. Proceedings of the 25th International Conference on Very-Large Databases (VLDB'99), (Edinburgh, cotland), 1999.

[Gionis 99] A. Gionis, P. Indyk, R. Motwani. Similarity search in high dimensions via hashing. Proc. of 25th VLDB, pages 518–529, 1999.

[Goldin 95] D. Goldin, P. Kanellakis. On Similarity Queries for Time-Series Data. Proceedings of CP '95, Cassis, France, Sept. 1995.

[Greenspan 02] H. Greenspan, J. Goldberger, A. Mayer, A probabilistic framework for spatio–temporal video representation and indexing. Proc. 7th European Conf. Computer Vision, Part IV, Berlin, Germany, 2002, pp. 461–475.

[Grimson 90] W. E. L. Grimson, D. P. Huttenlocher. On the sensitivity of geometric hashing. Proc. 3rd Intl. Conf. on Computer Vision, pages 334-338, 1990.

[Guttman 84] A Guttman. R-trees: A dynamic index structure for spatial searching. Proc. ACM SIGMOD Conf., pp 47-57, 1984.

[Haidar 04] S. Haidar, Ph. Joly, B. Chebaro. Audiovisual production invariant searching. 1ère Conférence en Recherche d'Information et Applications (CORIA'04), Toulouse, 10 mars 12 mars 2004. IRIT, ISBN 2-9520326-2-9, p. 333-346.
ftp://ftp.irit.fr/pub/IRIT/IHMPT/ART.ps/Articles/Siba.Haidar/coria04.pdf

[Haidar 04b] S. Haidar, Ph. Joly, B. Chebaro. Detection Algorithm of Audiovisual Production Invariant. 2nd Int. Workshop on Adaptative Multimedia Retrieval (AMR2004), Worksop 13 of 16th European Conference on Articial Intelligence (ECAI2004), Valencia, Spain, 23 août 27 août 2004. ftp://ftp.irit.fr/pub/IRIT/IHMPT/ART.ps/Articles/Siba.Haidar/AVPI_AMR04.pdf

[Haidar 05] B. Haidar, Ph. Joly, S. Haidar. A Graph Based Approach to Automatically Chain Distributed Multimedia Indexing Services. Ninth IASTED Int. Conf. on Internet and Multimedia Systems and Applications, Grindelwald, Switzerland, 21-23 février 2005.

[Haidar 05b] B. Haidar, Ph. Joly, J.-P. Bahsoun. Distributed Opened Cross-Media Indexing Platform. 11th Annual Scientific Conf. on Web Technology, New Media, Communications and Telematics Theory, Methods, Tools and Applications (EUROMEDIA'2005), Toulouse, 11 avril 13 avril 2005.

[Haidar 05c] S. Haidar, Ph. Joly, B. Chebaro. Mining for Video Production Invariants to Measure Style Similarity. Special issue on "Intelligent Multimedia Retrieval" of the Int. Journal of Intelligent Systems (IJIS), Wiley, V. -, p. à paraitre, 2005.

[Haidar 05d] S. Haidar, Ph. Joly, B. Chebaro. Style Similarity Measure for Video Documents Comparison. 4th Int. Conf. on Image and Video Retrieval (CIVR2005), Singapore, 20 juillet 22 juillet 2005.

[Haidar 05e] S. Haidar. Mesure de la similarité de style pour la comparaison de documents vidéo. Colloque annuel des Doctorants EDIT'05, UPS, Toulouse, 11 avril 12 avril 2005. Service Repro-UPS, p. 96-100.

[Hampapur 00] A. Hampapur, R. M. Bolle. Feature based Indexing for Media Tracking. Proc. of Int. Conf. On Multimedia and Expo, Aug. 2000, pp. 67-70.

[Hampapur 01] A. Hampapur, R. M. Bolle. Comparison of distance measures for video copy detection. Proc. of Int. Conf. on Multimedia and Expo, Aug. 2001.

[Hampapur 02] A. Hampapur, K.-H. Hyun, R. Bolle. Comparison of sequence matching techniques for video copy detection. Proceedings of SPIE - Storage and Retrieval for Media Databases 2002, San Jose, CA, January 2002, vol. 4676,

pp. 194-201.

[Hampapur 02b] A. Hampapur, R. Jain, T. E. Weymouth. Production model based digital video segmentation, Multimedia Tools Appl., vol. 1, pp. 9-46, 1995.

[Hanjalic 02] A. Hanjalic. Shot-boundary detection: Unraveled and resolved? IEEE Trans. Circ. Syst. for Video Tech., vol. 12, pp. 90-105, Feb. 2002.

[Haralik 87] R.M. Haralik, S.R. Sternberg, X. Zhuang. Image analysis using mathematical morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(4):532–550, July 1987.

[Hauptmann 98] A. Hauptmann, M. Witbrock. Story segmentation and detection of commercials in broadcast news video. Advances in digital librairies conference, ADL-98, Santa Barbara, CA., Apr. 22-24, 1998.

[Hebrard 84] J. J. Hebrard. Distances sur les mots. Application à la recherche de motifs. Thèse de 3e cycle, Université de Haute-Normandie, 1984.

[Herley 03] C. Herley. Argos: Automatically extracting repeating objects from multimedia streams. IEEE Trans. on Multimedia, 2003. Submitted. Available as MSR-TR-2004-02.

[Herley 04] C Herley. Extracting Repeats from Media Streams - .Acoustics, Speech, and Signal Processing, 2004.

[Hirschberg 77] D. S. Hirschberg. Algorithms for the longest common subsequence problem. J. ACM, 24:664-675, 1977.

[Hoi 03] C. Hoi, W. Wang, M. R. Lyu. A Novel Scheme for Video Similarity Detection. CIVR 2003: 373-382

[Horowitz 74] S. L. Horowitz, T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. Proc. 2nd Int. Joint Conf Pattern Recognition., 1974, pp, 424433.

[Houtgast 85] T. Houtgast et J. M. Steeneken. A Review of the MTF Concept in Room Acoustics and its Use for Estimating Speech Intelligibility in Auditoria. Journal of the Acoustical Society of America, 77(3) :1069–1077, 1985.

[Hunt 77] J. W. Hunt, T. G. Szymanski. A fast algorithm for computing longest common subsequences. Commun. ACM, 20:350-353, 1977.

[Indyk 99] P. Indyk, G. Iyengar, N. Shivakumar. Finding pirated video sequences on the Internet. Stanford Infolab, Stanford, CA, Tech. Rep., Feb. 1999.

[Iyengar 98] G. Iyengar, A. Lippman. Distributional clustering for efficient content-based retrieval of images and video. Proc. 1998 Int. Conf. Image Proces-

sing, vol. III, Vancouver, BC, Canada, 2000, pp. 81–84.

[Jaffré 04] G. Jaffré, Ph Joly, S. Haidar. The SAMOVA Shot Boundary Detection for TRECVID Evaluation 2004. Proceedings of TRECVIDEO 2004 Workshop, Gaithersburg, Maryland, USA, 2004

[Jagadish 95] H. V. Jagadish, A. O. Mendelzon, T. Milo. Similarity-based queries. Proc. of the 14th ACM PODS, pages 36–45, May 1995.

[Jaimes 03] A. Jaimes, J. R. Smith. Semi-automatic, data-driven construction of multimedia ontologies. Proc. IEEE ICME, 2003.

[Joly 96] Ph. Joly, H.-K. Kim. Efficient Automatic Analysis of Camera Work and Microsegmentation of Video Using Spatio-Temporal Images. Signal Processing : Image Communication, Elsevier, Eurasip, Amsterdam. 1996.

[Kahveci 01] T. Kahveci and A. K. Singh. Variable length queries for time series data. Proc. of IEEE ICDE, Proceedings 17th International Conference on Data Engineering, Heidelberg, Germany, pages 273–282, 2001.

[Keogh 00] E. Keogh, M. Pazzani. Scaling up Dynamic Time Warping for Datamining Applications. Proc. 6th Int. Conf. On Knowledge Discovery and Data Mining, Boston, MA, 2000.

[Keogh 01] E. Keogh, K. Chakrabarti, S. Mehrotra, M. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. Proc. of ACM SIGMOD, pages 151– 162, 2001.

[Keogh 97] E. Keogh, P. Smyth. A probabilistic approach to fast pattern matching in time series databases. Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining. pp 24-20. 1997.

[Keogh 99] E. Keogh, M. Pazzani. An indexing scheme for similarity search in large time series databases. SSDBM, Cleveland, Ohio, 1999.

[Kim 01] S. H. Kim, R.-H. Park. An efficient algorithm for video sequence matching using the Hausdorff distance and the directed divergence. Proc. SPIE Visual Communications and Image Processing 2001, vol. 4310, pp. 754-761, San Jose, CA, Jan. 2001.

[Kim 02] S. H. Kim, R.-H. Park. An efficient video sequence matching using the Cauchy function and the modified Hausdorff distance. Proc. SPIE Storage and Retrieval for Media Databases 2002, pp. 232-239, San Jose, CA, USA, Jan. 2002.

[Kobla 96] V. Kobla, D.S. Doermann, K-I. Lin, C. Faloutsos. Compressed domain video indexing techniques using DCT and motion vector information in

MPEG video. In Proceedings of the SPIE conference on Storage and Retrieval for Image and Video Databases V, Volume 3022, pages 200-211, 1996.

[Kollios 99] G. Kollios, D. Gunopulos, V. Tsotras. On Indexing Mobile Objects. Proc. of the 18th ACM Symp. on Principles of Database Systems (PODS), pages 261–272, June 1999.

[Kraaij 04] W. Kraaij, A. Smeaton, P. Over, J. Arlandis. TRECVIDEO 2004 – an Introduction. Proceedings of TRECVIDEO 2004 Workshop, Gaithersburg, Maryland, USA, 2004

[Kulesh 02] V. Kulesh et al. Video clip recognition using joint audio-visual processing model. In Proc. of ICPR'02, vol. 1, pp. 500-503, 2002

[Kushilevitz 98] E. Kushilevitz, R. Ostrovsky, Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, pp. 61423, May 1998.

[Lamdan 88] Y. Lamdan, H. J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. Proc. 2nd Intl. Conf. on Computer Vision, pages 238-249, 1988.

[Lee 00] S.-L. Lee, S.-J. Chun, D.-H. Kim, J.-H. Lee, C.-W. Chung. Similarity search for multidimensional data sequences. ICDE, 599-608, San Diego, CA, 2000.

[LibMpeg2] http://libmpeg2.sourceforge.net/

[Lienhart 01] R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide, Int. J. Image Graph., vol. 1, pp. 469-486, Aug. 2001.

[Lienhart 97] R. Lienhart, C. Kuhmünch, W. Effelsberg. On the Detection and Recognition of Television Commercials. Proceedings of the International Conference on Multimedia Computing and Systems, Ottawa, Ontario, Canada, pp. 509-516, June 1997.

[Lienhart 97b] R. Lienhart, W. Eelsberg, R. Jain, VisualGREP: a systematic method to compare and retrieve video sequences. Proceedings of IS&T=SPIE Conference on Storage and Retrieval for Image and Video Database VI, Vol. SPIE 3312, 1997, pp. 271–282.

[Lin 01] Tong Lin and Hong-Jiang Zhang. Integrating Color And Spatial Features for Content-Based Video Retrieval. Invited Paper, Proceedings of 2001 International Conference on Image Processing, Thessaloniki, Greece. October 7-10, 2001.

[Lopresti 94] D. Lopresti, A. Tomkins. On the Searchability of Electronic Ink. IWFHR 94.

[Ma 02] Yu-Fei Ma, Hong-Jiang Zhang. Motion Texture: A New Motion based Video Representation. Proceeding of 2002 International Conference on Pattern Recognition, ICPR, August, 2002.

[MacQueen 67] J. MacQueen. Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. Mathematical Statistics, vol. 1, 1967, pp. 281–297.

[Manjunath 02] B. S. Manjunath, P. Salembier, T. Sikora (Eds), Introduction to MPEG-7: Multimedia Content Description Interface, Wiley, 2002.

[Manolopoulos 03] Y. Manolopoulos, A. Nanopoulos, A. N. Papadopoulos, Y. Theodoridis. R-trees have grown everywhere. 2003. Technical Report available at http://www.rtreeportal.org/

[Matheron 75] G. Matheron. Random Sets and Integral Geometry. Wiley, New York City, NY, USA, 1975.

[McConnell 91] R. McConnell et al. Correlation and dynamic time warping: Two methods for tracking ice in SAR images. IEEE Transactions on Geoscience and Remote Sensing, 29(6):1004-1012, 1991.

[Moddemeijer 89] R. Moddemeijer. On Estimation of Entropy and Mutual Information of Continuous Distributions. Signal Processing, 16(3) :233–246, 1989.

[Mohan 98] R. Mohan. Video Sequence Matching. Proc ICASSP '98, IEEE, May 1998, Seattle.

[Mukherjee 97] S. Mukherjee, E. Osuna, F. Girosi. Nonlinear prediction of chaotic time series using support vector machines. Proceeding of IEEE Neural Networks for Signal Processing, NNSP'97, pages 24–26, September 1997.

[Nakatsu 82] N. Nakatsu, Y. Kambayashi, S. Yajima. A longest common subsequence algorithm suitable for similar text strings. Acta Inf., 18:171-179, 1982.

[Naphade 01] M. Naphade, R. Wang, T. Huang, Multimodal pattern matching for audio-visual query and retrieval. Proc. SPIE, Storage and Retrieval for Media databases, M. Naphade et al, Volume 4315, pages 188-195, Jan 2001, San Jose, CA.

[Needleman 70] SB Needleman, CD Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology 48(1): 443-453. 1970.

[Park 00] S. Park, J. Y. Wesley W. Chu, C. Hsu. Fast retrieval of similar subsequences of different lengths in sequence databases. ICDE, San Diego, CA, February 2000.

[Park 00b] S. Park, W. Chu, J. Yoon, C. Hsu. Efficient Searches for Similar Subsequences of Different Lengths in Sequence Databases. Proceedings of ICDE, pages 23–32, 2000.

[Pass 96] G. Pass, R. Zabih, J. Miller. Comparing images using color coherence vectors. Proc. ACM Multimedia, pages 65–73, Nov. 1996.

[Paterson 94] M. Paterson, V. Dancik. Longest common subsequences. Mathematical Foundations of Computer Science, 19th International Symposium (MFCS), volume 841 of LNCS, pages 127–142, 1994.

[Perng 00] C.-S. Perng, H. Wang, S. R. Zhang, D. S. Parker. Landmarks: a new model for similarity-based pattern querying in time series databases. ICDE, pages 33–42, San Diego, USA, February 2000.

[pestmanagement] www.pestmanagement.co.uk/library/gloss_d2.html

[Pfoser 00] D. Pfoser, C. Jensen, Y. Theodoridis. Novel Approaches in Query Processing for Moving Objects. Proceedings of VLDB, Cairo Egypt, Sept. 2000.

[Pinquier 03] J. Pinquier, J.-L. Rouas, R. André-Obrecht. Fusion de paramètres pour une classification automatique parole/musique robuste. Technique et Science Informatiques (TSI), 22(7-8) :831–852, 2003.

[Pinquier 04] J. Pinquier. Indexation sonore : recherche de composantes primaires pour une structuration audiovisuelle. Thèse de doctorat, Université Paul Sabatier, Toulouse,décembre 2004.

[Pinsach 03] J. Llach Pinsach. Analysis of video sequences for content descryption. Table of content and index creation and scene classification. Thèse doctoral. Departament théorie du signal et communications. Univerité Polytechnique de Catalunya. Mai 2003.

[Qu 98] Y. Qu, C. Wang, X. Wang. Supporting Fast Search in Time Series for Movement Patterns in Multiple Scales. Proc of the ACM CIKM, pages 251–258, 1998.

[Rafiei 00] D. Rafiei, A. Mendelzon. Querying Time Series Data Based on Similarity. IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No 5., pages 675–693, 2000.

[Rafiei 97] D. Rafiei, A. O. Mendelzon. Similarity-based queries for time series data. In ACM SIGMOD, pages 13–25, Tucson, AZ, 1997.

[Rafiei 98] D. Rafiei, A. O. Mendelzon. Efficient retrieval of similar time sequences using dft. In FODO, Kobe, Japan, 1998.

[Roytberg 92] M. Roytberg. A search for common patterns in many sequences. Computer Applications in the Biosciences, 8(1):57-64, 1992.

[Sakoe 78] H. Sakoe, S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-26(1):43–49, Feb. 1978.

[Saltenis 00] S. Saltenis, C. Jensen, S. Leutenegger, M. A. Lopez. Indexing the Positions of Continuously Moving Objects. Proceedings of the ACM SIGMOD, pages 331–342, May 2000.

[Santini 99] S. Santini, R. Jain. Similarity measures. IEEE Trans. Pattern Anal. Machine Intell., vol. 21, pp. 871–883, Sept. 1999.

[Satoh 01] S. Satoh, Y. Nakamura, T. Kanade. Name-It: Naming and detecting faces in news videos. IEEE Multimedia, vol. 6, no. 1, pp. 22-35, Jan.-Mar. 2001.

[Schmitt 94] M. Schmitt, J. Mattioli. Morphologie Mathématique. Masson, Paris, 1994.

[Sellers 74] Sellers P. 1974. On the theory and computation of evolutionary distances. SIAM J Appl Math 26:787–793.

[Serra 82] J. Serra. Image Analysis and Mathematical Morphology. New York: Academic Press, 1982, pp. 390-423.

[Serra 84] J. Serra. Image Analysis and Mathematical Morphologie. Academic Press, London, UK, 1984.

[Serra 88] J. Serra. Image Analysis and Mathematical Morphologie: Theoretical Advances, volume 2. Academic Press, London, UK, 1988.

[Shahabi 00] C. Shahabi, X. Tian, W. Zhao. TSA-tree: A waveletbased approach to improve the efficieny of multi-level surprise and trend queries. SSDBM, 2000.

[Shan 98] M.-K. Shan, S.-Y. Lee. Content-based Video Retrieval based on Similarity of Frame Sequence. Proc. IEEE Conf. on Multimedia Computing and Systems, pp.90-97, 1998

[Shivakumar 98] N. Shivakumar, H. Garcia-Molina. Finding nearreplicas of documents on the web, in World Wide Web and Databases. International Workshop WebDB'98, Valencia, Spain, pp. 204-12, Mar 1998.

[Simon 88] Imre Simon. Sequence Comparison: Some Theory and Some Practice. Instituto de Matemática e Estatística, Universidade de São Paulo 05508 São Paulo, SP, Brasil. April 2, 1988.

http://www.ime.usp.br/~is/papir/sctp/node1.html

[Singh 98] A. Singh, D. Agrawal, K. V. R. Kanth. Dimensionalityreduction for similarity searching in dynamic databases. ACM SIGMOD, Seattle, WA, June 1998.

[Smith 81] TF Smith, MS Waterman. Comparison of biosequences- Adv. Appl. Math, 2, 482-489, 1981

[SourceForge] http://sourceforge.net

[Stuart 77] S. Dreyfus, A. Law. The Art and Theory of Dynamic Programming, Academic Press, Inc., 1977.

[Sundaram 02] H. Sundaram, S.-F. Chang. Computable scenes and structures in films. IEEE Trans. Multimedia, vol. 4, pp. 482-491, Dec. 2002.

[Tan 00] Y. P. Tan, S. R. Kulkarni, P. J. Ramadge. Rapid estimation of camera motion from compressed video with application to video annotation. IEEE Trans. Circ. Syst. Video Tech., v. 10, pp. 133-146, 2000.

[Tan 99] Y. Tan, S. Kulkarni, P. Ramadge. A framework for measuring video similarity and its application to video query by example. Proc. of ICIP, Kobe, Japan. 1999.

[Tekalp 04] A. M. Tekalp, Automatic video segmentation and indexing – where are we ? 5th international workshop on image analysis for multimedia interactive services. April 21-23, Lisboa, Portuogal, 2004.

[Uhlmann 91] J.K. Uhlmann. Satisfying General Proximity/Similarity Queries with Metric Trees. Information Processing Letters, v40, p175-179,1991.

[Vasconcelos 00] N. Vasconcelos, A. Lippman. Statistical models of video structure for content analysis and characterization. IEEE Trans. Image Process. 9 (1) (2000) 3–19.

[Vasconcelos 01] N. Vasconcelos. On the complexity of probabilistic image retrieval. Proc. 8th IEEE Int. Conf. Computer Vision, vol. 2, Vancouver, BC, Canada, 2001, pp. 400–407.

[Vingron 89] M. Vingron, P. Argos. A fast and sensitive multiple sequence alignment algorithm. Computer Applications in the Biosciences, 5:115-122, 1989.

[Vlachos 02] M. Vlachos, G. Kollios, G. Gunopulos. Discovering similar multidimensional trajectories. In proceedings 18th International Conference on Data Engineering. pp 673-684. 2002.

[Vlachos 03] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, E. Keogh. Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures. 9th ACM SIGKDD. August 24 - 27, 2003. Washington, DC, USA. pp 216-225.

[Wald 99] L. Wald. Some Terms of Reference in Data Fusion. IEEE Transactions on Geoscience and Remote Sensing, 1999.

http://ieeexplore.ieee.org/iel5/36/16544/00763269.pdf

[Wang 03] W. Wang, M.R. Lyu. Automatic Generation of Dubbing Video Slides for Mobile Wireless Environment. Proc. of IEEE International Conf. on Multimedia and Expo, Orlando, Florida, July 27-30, 2003.

[Wang 94] J. T.-L. Wang, G.-W. Chirn, T. G. Marr, B. Shapiro, D. Shasha, K. Zhang. Combinatorial pattern discovery for scienti_c data: Some preliminary results. Proc. of the ACM SIGMOD Conference on Management of Data, Minneapolis, May 1994.

[WebActu] Extrait d'un critique sur Webactua.net, par Fëanor - Publié le mercredi 28 mai 2003. Par Fëanor - Publié le mercredi 28 mai 2003. http://www.ados.fr/cinema/salles-dvd/matrix-reloaded-la-critique_article448.html

[Weber 98] R. Weber, H.-J. Schek, S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. Proceedings of the 24th International Conference on Very-Large Databases (VLDB'98), pp. 194205, (New York, NY, USA), August 1998.

[Weisstein 99] E.W. Weisstein. Variation Coefficient. From MathWorld. A Wolfram Web Resource. 1999.

http://mathworld.wolfram.com/VariationCoefficient.html

[Wu 00] Y. Wu, Y. Zhuang, Y. Pan. Content-Based Video Similarity Model. Proc. of the 8th ACM International Multimedia Conf. on Multimedia, Marina del Rey, CA, USA, pp.465-467, October 30-November 04, 2000.

[Wu 92] S. Wu, U. Manber. Fast text searching allowing errors. Communications of the ACM, 35(10):83-91, October 1992.

[Yazdani 96] N. Yazdani, M. Ozsoyoglu. Sequence Matching of Images. Proceeding of

8'th Int. Conf. on Statistical and Scientific Database, 1996.

[Yeung 95] M. M. Yeung, B. Liu, Efficient Matching and Clustering of Video shots. Proceedings of International Conference on Image processing'95, Washington, DC, pp. 338-341, 1995.

[Yi 00] B.-K. Yi, C. Faloutsos. Fast Time Sequence Indexing for Arbitrary Lp Norms. Proceedings of VLDB, Cairo Egypt, Sept. 2000.

[Yi 98] B.-K. Yi, H. Jagadish, C. Faloutsos. Efficient retrieval of similar time sequences under time warping. ICDE 98, pages 23 – 27, Orlando, Florida, February 1998.

[Zhang 97] H. J. Zhang, D. Zhong, S. W. Smoliar. An Integrated System for Content-Based Video Retrieval and Browsing. Pattern Recognition, Vol. 30, No, 4, pp. 643-658, 1997.

[Zhong 95] D. Zhong, H. J. Zhang, S. F. Chang. Clustering Methods for Video Browsing and Annotation. Proceedings of IS&T/SPIE Storage and Retrieval for Image and Video Databases IV, San Jose, CA, pp. 239-246, 1995.

[Zhong 99] D. Zhong, S. F. Chang. An integrated approach for content-based video object segmentation and retrieval. IEEE Trans. Circ. Syst. for Video Tech., vol. 9, no. 8, pp. 1259-1268, Dec. 1999.

ANNEXE A : QUELQUES MATRICES DE COMPARAISON

Dans cet annexe, nous présentons quelques matrices de comparaison. Il s'agit de la comparaison de plages publicitaires extraites de deux journées de télévision - 6 plages de chaque journée.

Tout d'abord les résumés de ces plages sont présentés sous forme de vignettes. Chaque vignette représente un film publicitaire. Les 62 films publicitaires sont ensuite identifiés par des numéros, voir le tableau plus loin. Un graphe de correspondance est dressé manuellement pour indiquer les films communs entre chaque couple de plages.

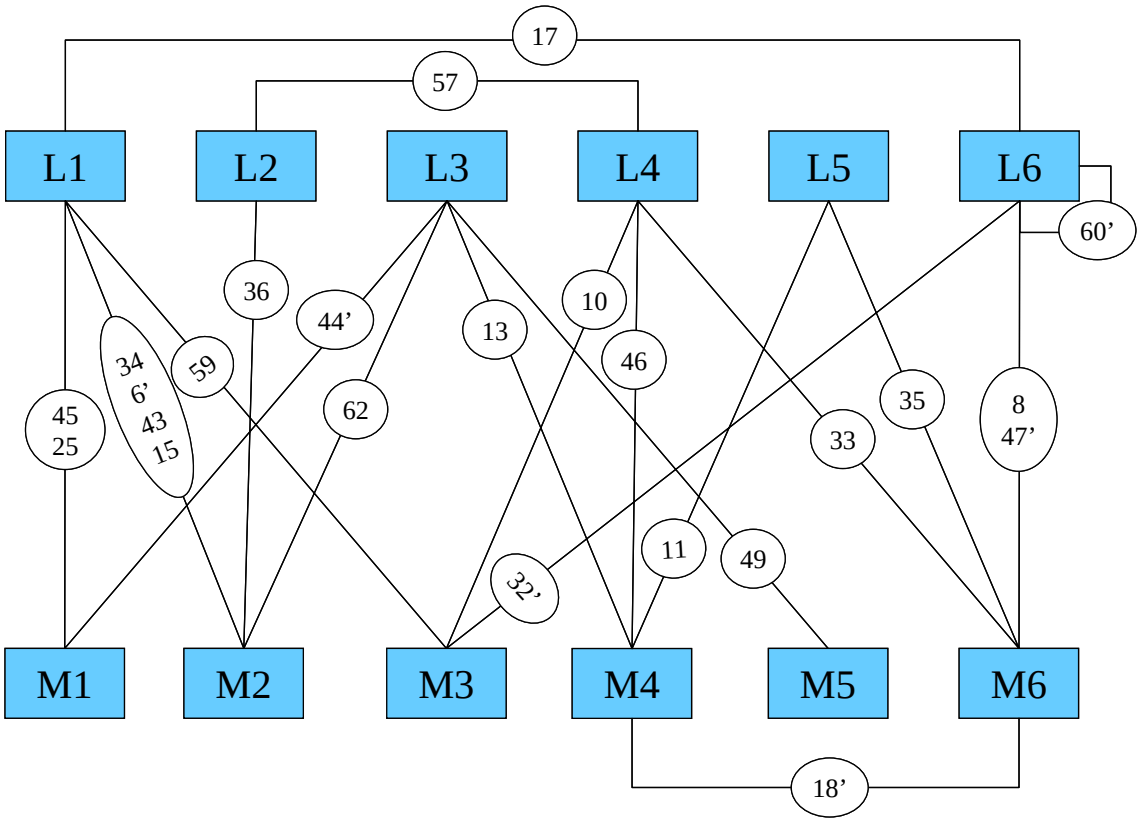
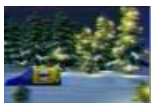


Fig. A.1. Graphe de correspondance entre les plages publicitaires. (L = Lundi / M = Mardi)

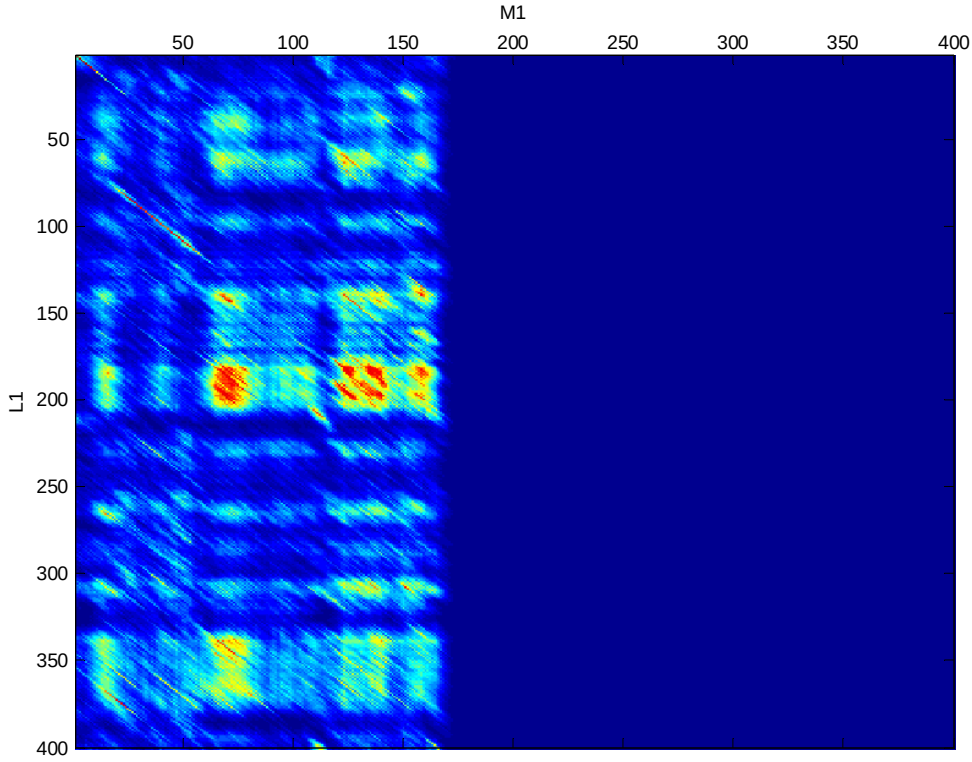
Lundi	**L1** **03:33**	 1 moripion 0001-0007	 2 F2PUB 0008-0011	 3 centerparcs 0011-0040	 4 FleuryMichon 0042-0101	 5 JamesBond 0103-0132
 6 Antibiotiques 0133-0153	 7 Masques 0154-0214	 8 Calgon 0215-0245	 9 HenriDès 0245-0255	 10 Wattwiller 0256-0326	 11 F2PUB 0326-0330	 12 Logo2 0330-0333
L2 **02 :30**	 1 F2PUB 0000-0004	 2 KnorEpinarts 0005-0025	 3 Tele2 0025-0055	 4 Volvic 0056-0112	 5 cdNocturne 0113-0128	 6 KubOr 0128-0143
 7 Signal 0144-0204	 8 TailleFine 0205-0225	 9 F2PUB 0226-0230	L3 02 :35	 1 F2PUB 0000-0004	 2 Bridelight 0005-0025	 3 Roc 0027-0046
 4 Senoble 0047-0107	 5 Mediatis 0108-0138	 6 StYorre 0140-0153	 7 Chess 0155-0204	 8 Wok 0206-0230	 9 F2PUB 0231-0235	**L4** **02 :25**
 1 F2PUB 0000-0004	 2 LorealPrefe- rance 0004-0034	 3 Barilla 0035-0105	 4 PetroleHahn 0106-0126	 5 HolidayOnIce 0127-0146	 6 CeriseDePro- vence 0147-0150	 7 Tele2 0151-0220
 8 F2PUB 0220-0224	L5 01 :13	 1 F2PUB 0000-0004	 2 3213gagner 0005-0012	 3 Bjorg 0013-0021	 4 Anadvil 0022-0030	 5 Lactel 0031-0101
 6 3213gagner2 0102-0107	 7 F2PUB 0108-0112	L6 04 :21	 1 F2PUB 0000-0004	 2 FruitDOr 0005-0016	 3 Diadermine 0017-0037	 4 CenterParks 0039-0058
 5 Fuca 0059-0107	 6 Leerdammer 0108-0123	 7 Audika 0124-0154	 8 Wcnet 0155-0211	 9 Synthol 0212-0227	 10 PFGPre- voyances 0228-0258	 11 Gourmet 0259-0314
 12 Hepatoum 0315-0326	 13 wcnet 2 0327-0337	 14 3213gagner 0339-0344	 15 Amora 0345-0415	 16 F2PUB 0416-0420		

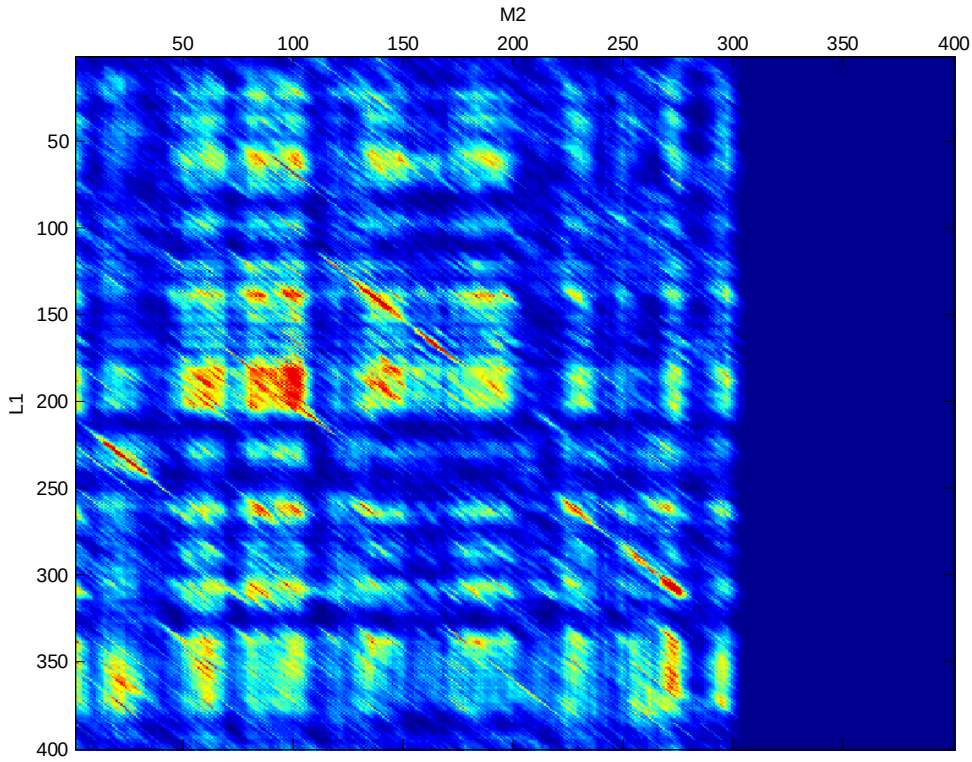
Mardi	M1 01 :30	 1 Moripion 0000-0007	 2 F2PUB 0007-0010	 3 FleuryMichon 0011-0031	 4 Mediatis2 0032-0102	 5 Boursin 0103-0123
 6 F2PUB 0124-0128	M2 02 :39	 1 F2PUB 0000-0004	 2 Masques 0005-0025	 3 KubOr 0026-0041	 4 Antibiotiques2 0042-0102	 5 JamesBond 0103-0130
 6 Wok 0131-0154	 7 Calgon 0156-0226	 8 3213gagner2 0227-0232	 9 F2PUB 0233-0236	M3 02 :34	 1 F2PUB 0000-0004	 2 3213gagner 0005-0012
 3 Brossard 0013- 0029	 4 LOrealRevita-lift 0029-0049	 5 BANature 0050-0108	 6 Wattwiller 0110-0140	 7 Hepatoum2 0141-0150	 8 Barilla 0151-0221	 9 3213gagner2 0222-0228
 10 F2PUB 0228-0232	M4 02 :43	 1 F2PUB 0000-0004	 2 WeightWat-chers 0005-0019	 3 3213gagner 0020-0028	 4 EauEclerante 0029-0046	 5 Bjorg 0048-0055
 6 PetroleHahn 0056-0116	 7 Bridelight 0117-0138	 8 CeriseDePro-vince2 0039-0142	 9 GarnierBelle-Color 0143-0213	 10 DrPierreRi-caud 0214-0234	 11 3113gagner2 0235-0240	 12 F2PUB 0241-0243
M5 01 :16	 1 Darty 0001-0007	 2 F2PUB 0008-0012	 3 Roc 0013-0033	 4 AGagner 0033-0053	 5 ProPlan 0055-0114	 6 F2PUB 0114-0116
M6 04 :18	 1 F2PUB 0000-0004	 2 Taft 0005-0025	 3 HolidayOnIce 0026-0045	 4 AssuranceMa-ladie 0046-0106	 5 Audika 0107-0137	 6 CeriseDePro-vince 138-0141
 7 PFGPre-voyance2 0142-0112	 8 Codotussyl 0213-0219	 9 Florette 0220-0240	 10 Libra 0241-0311	 11 Rothelec 0312-0342	 12 Lactel 0343-0412	 13 F2PUB 0413-0417

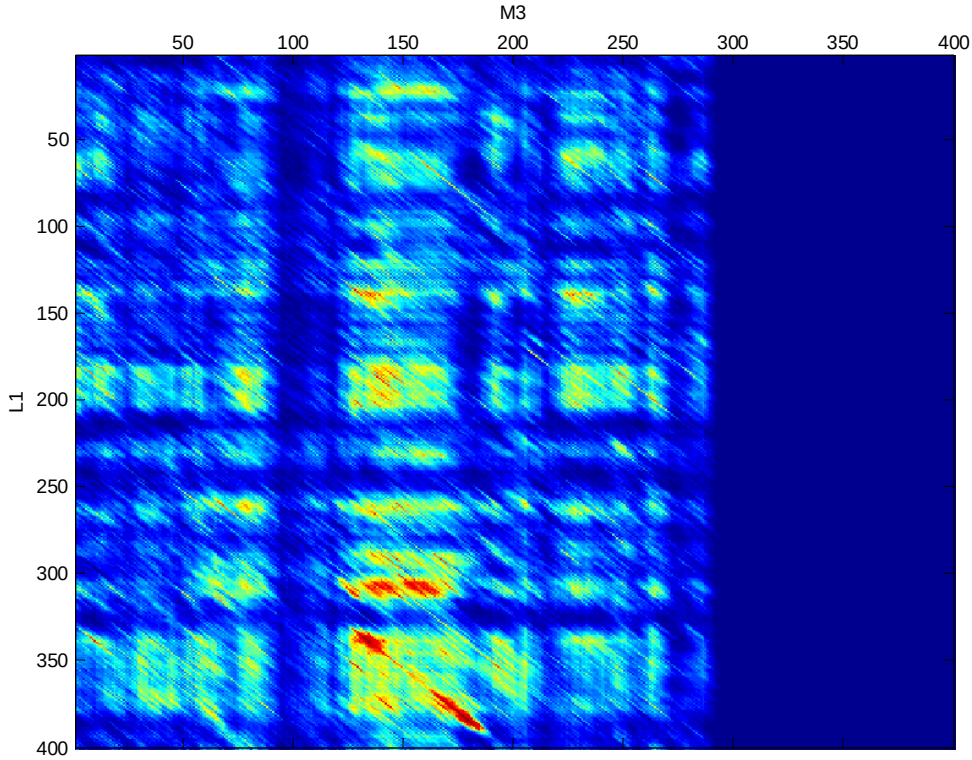
Nous désignons chaque film publicitaire par un identifiant :

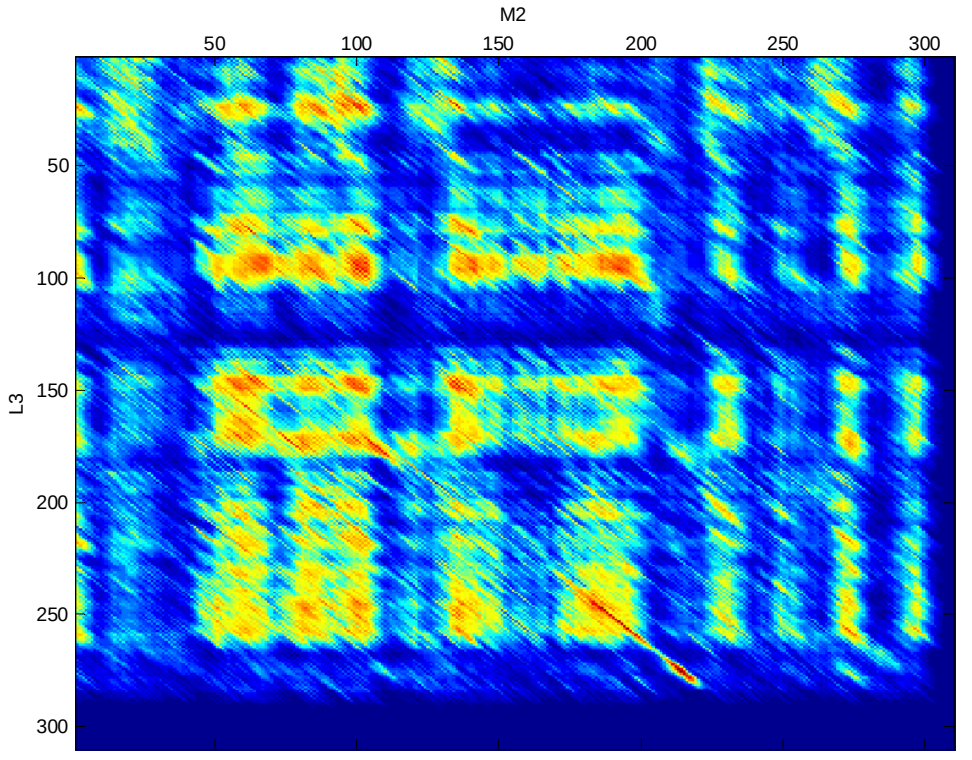
1	F2 PUB	22	Diadermine	43	Masques
2	3213 gagner	23	Dr Pierre Ricaud	44	Mediatis
3	AGagner	24	Eau Ecalrate	45	Moripion
4	Amora	25	Fleury Michon	46	Petrole Hahn
5	Anadvil	26	Florette	47	PFG Prevoyances
6	Antibiotiques	27	Fruit D'Or	48	ProPlan
7	Assurance Maladie	28	Fuca	49	Roc
8	Audika	29	Garnier Belle Color	50	Rothelec
9	BANature	30	Gourmet	51	Senoble
10	Barilla	31	Henri Dès	52	Signal
11	Bjorg	32	Hepatoum	53	StYorre
12	Boursin	33	Holiday On Ice	54	Synthol
13	Bridelight	34	James Bond	55	Taft
14	Brossard	35	Knor Epinards	56	Taille Fine
15	Calgon	36	KubOr	57	Tele2
16	cdNocturne	37	Lactel	58	Volvic
17	CenterParks	38	Leerdammer	59	Wattwiller
18	Cerise de Province	39	Libra	60	WCNet
19	Chess	40	Logo2	61	Weight Watchers
20	Codotussyl	41	L'Oréal Préférence	62	Wok
21	Darty	42	L'Oréal Revitalift		

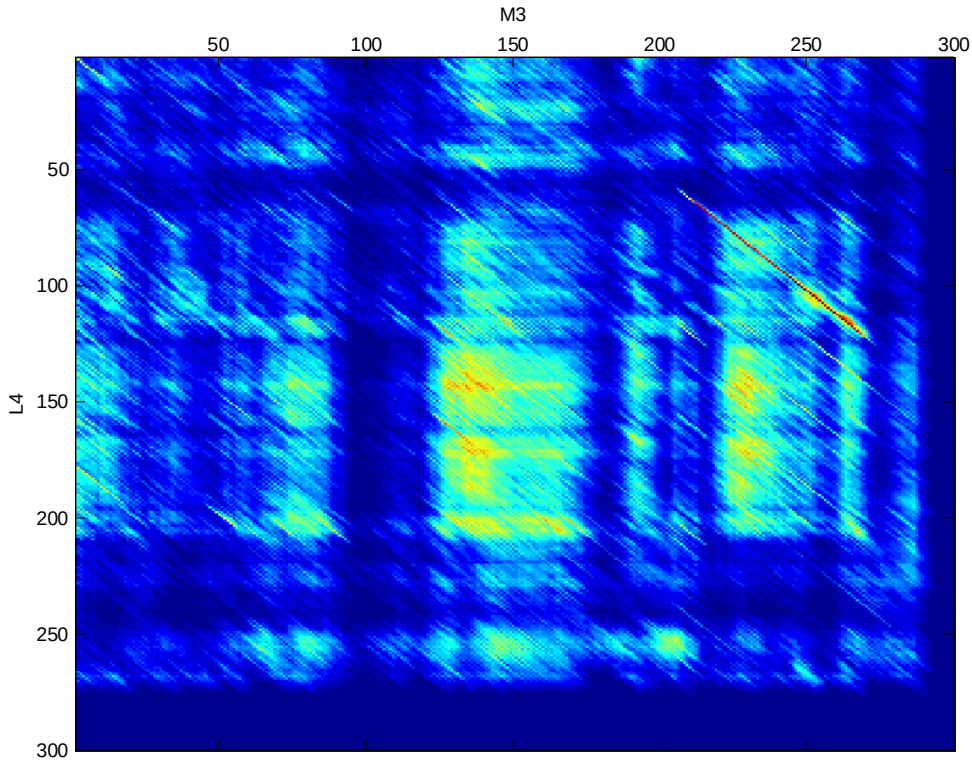
Nous avons calculé les matrices de comparaison pour chaque couple de plage. Nous montrons par la suite quelques unes de ces matrices. Sur ces matrices les valeurs de couleurs chaudes réparties diagonalement correspondent aux films communs. Les blocs de valeurs chaudes, désignent à leur tour des similarités entre les segments vidéo comparés. Ces similarités ne respectent pas l'ordre, et ne sont pas de valeurs très élevées. Les segments désignés sont par suite des segments de caractéristiques semblables mais non pas nécessairement identiques.

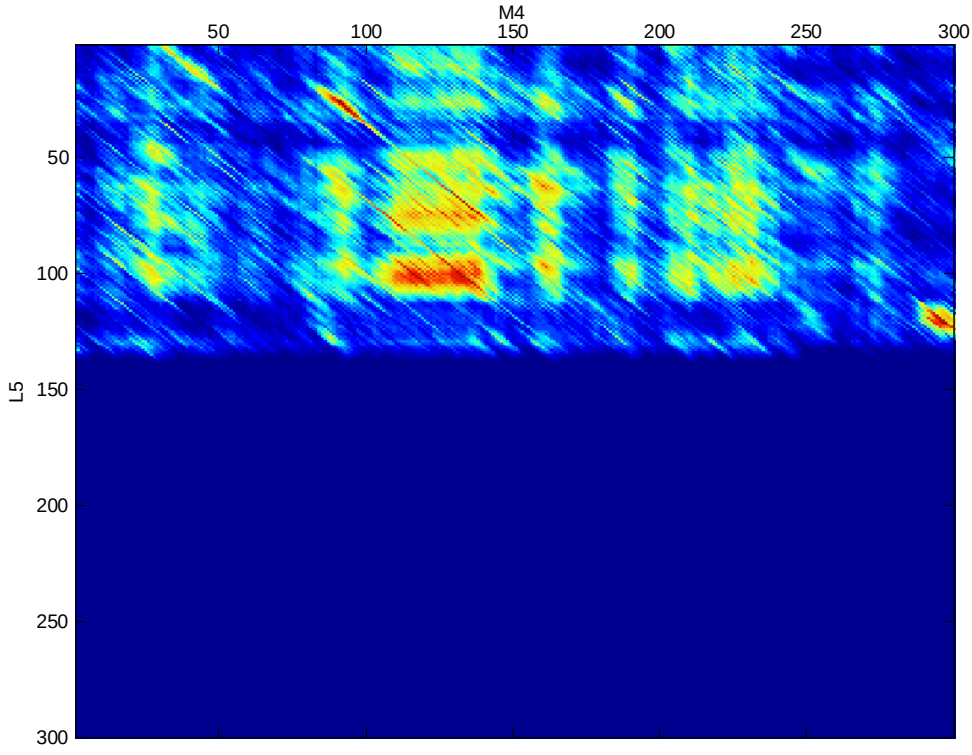

M1
50
100
150
200
250
300
350
400
L1
50
100
150
200
250
300
350
400

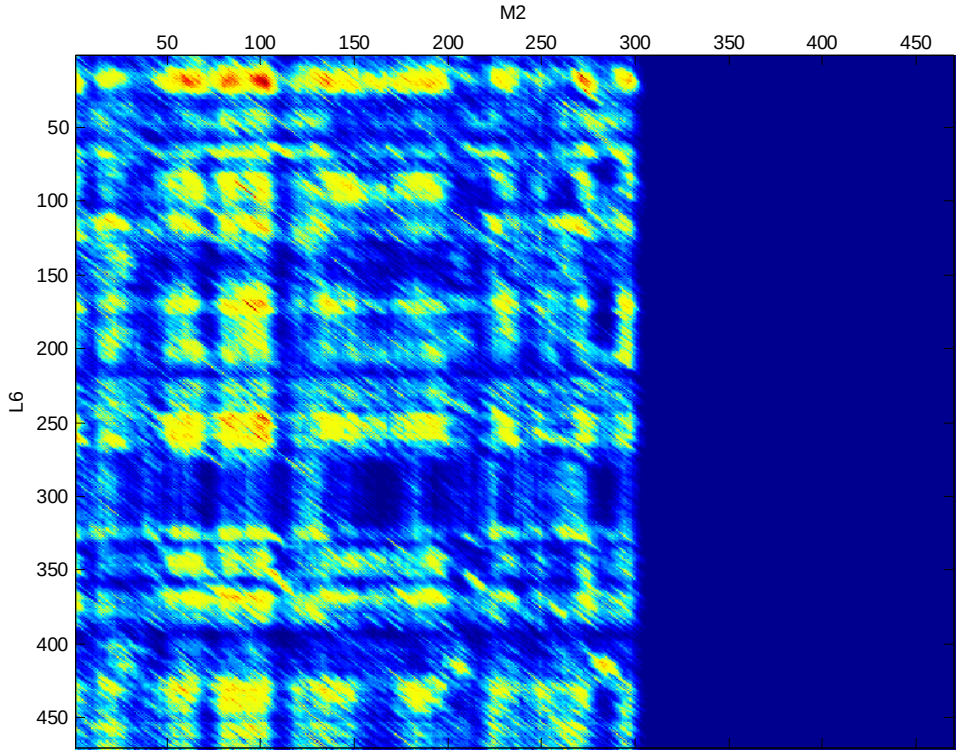

M2
50
100
150
200
250
300
350
400
L1
50
100
150
200
250
300
350
400

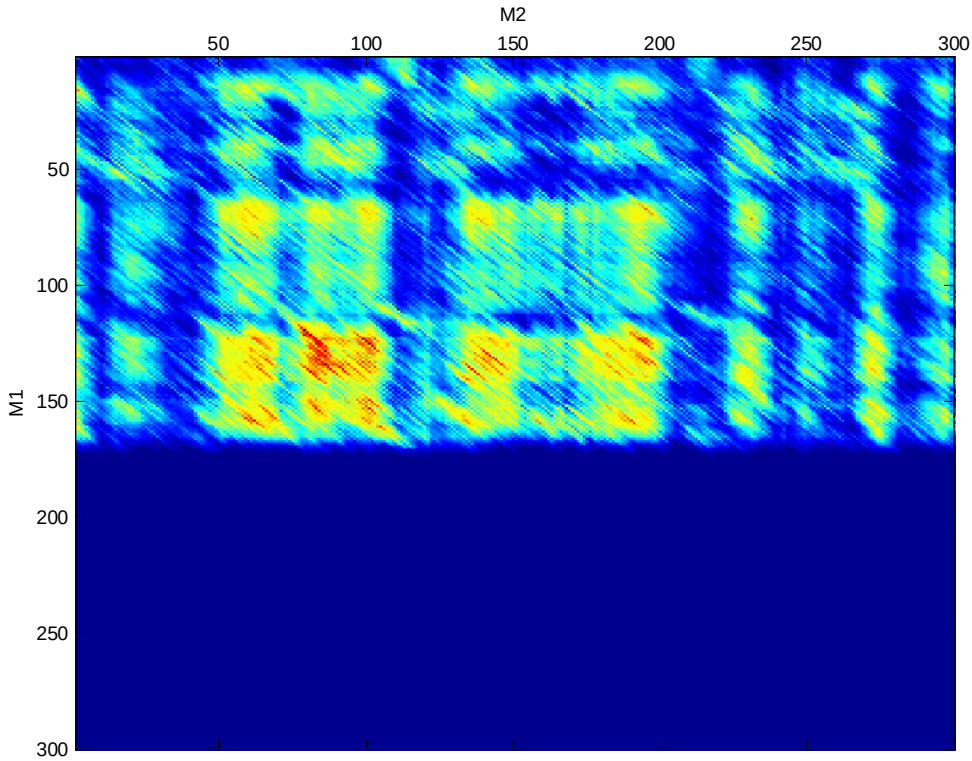

M3
50
100
150
200
250
300
350
400
L1
50
100
150
200
250
300
350
400


M2
50
100
150
200
250
300
L3
50
100
150
200
250
300


M3
50
100
150
200
250
300
L4
50
100
150
200
250
300


M4
50
100
150
200
250
300
L5
50
100
150
200
250
300


M2
50
100
150
200
250
300
350
400
450
L6
50
100
150
200
250
300
350
400
450


M2
50
100
150
200
250
300
M1
50
100
150
200
250
300

ANNEXE B : PARALLELISATION

Les algorithmes de comparaison définis dans le chapitre 2 sont des outils très rapides. Ils peuvent être facilement parallélisés pour réduire la durée de la comparaison en raison des appels quadratiques indépendants mis en oeuvre, d'une part, et en raison de l'exploitation séparée des caractéristiques audiovisuelles à comparer. Nous avons profité de ces propriétés pour générer une version effectuant des appels parallèles de notre outil de comparaison.

Choix et mise en oeuvre

La parallélisation multitâches existe depuis longtemps chez de nombreux constructeurs (ex. CRAY, NEC, IBM, ...), mais chacun avait son propre jeu de directives. Le retour en force des machines multiprocesseur à mémoire partagée a poussé à définir un standard. Un consortium d'industriels et de constructeurs a ainsi choisi d'adopter OpenMP (Open Multi Processing) comme un standard dit «industriel » en 1997. Les spécifications d'OpenMP appartiennent aujourd'hui à l'ARB (Architecture Review Board).

Un programme OpenMP est une alternance de régions séquentielles et de régions parallèles. Une région séquentielle est toujours exécutée par la tâche maîtresse. Une région parallèle peut être exécutée par plusieurs tâches à la fois. Les tâches peuvent se partager le travail contenu dans la région parallèle.

Un programme OpenMP est exécuté par un processus unique. Ce processus active des processus légers (threads) à l'entrée d'une région parallèle. Chaque processus léger exécute une tâche composée d'un ensemble d'instructions. Pendant l'exécution d'une tâche, une variable peut être lue et/ou modifiée en mémoire soit dans la pile d'un processus léger (variable privée), soit dans un espace mémoire partagé (variable partagée) [Chergui 04].

Le partage du travail consiste essentiellement à :

- exécuter une boucle par répartition des itérations entre les tâches ;
- exécuter plusieurs sections de code mais une seule par tâche ;
- exécuter plusieurs occurrences d'une même procédure par différentes tâches (orphaning).

Il est parfois nécessaire d'introduire une synchronisation entre les tâches concurrentes pour éviter, par exemple, que celles-ci modifient dans un ordre quelconque la valeur d'une même variable partagée (cas des opérations de réduction). Différents cas peuvent se produire sur une architecture multiprocesseur :

- au mieux, à chaque instant, il existe une tâche par processeur avec autant de tâches que de processeurs dédiés pendant toute la durée du travail ;
- au pire, toutes les tâches sont traitées séquentiellement par un et un seul processeur;
- en réalité, pour des raisons essentiellement d'exploitation sur une machine dont les processeurs ne sont pas dédiées, la situation est en général intermédiaire. Ce sera effectivement le cas pour l'implémentation que nous proposons.

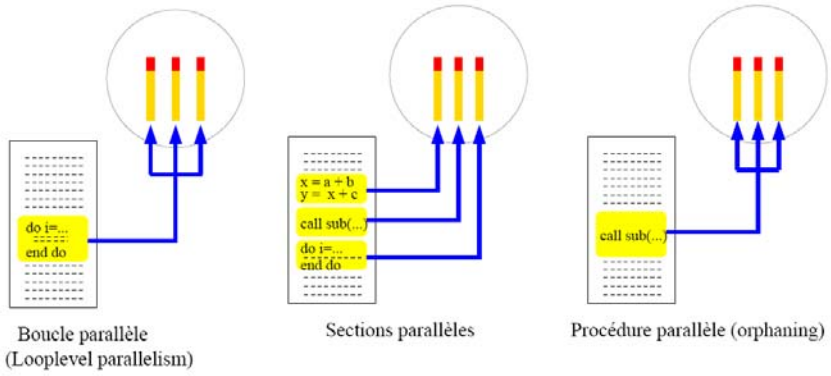


Fig. B1. Différents types de codes parallèles. [Chergui 04]

Structure d'OpenMP

OpenMP est composé :

- de directives et de clauses de compilation : elles servent à définir le partage du travail, la synchronisation et le statut privé ou partagé des données.
- de fonctions et de sous-programmes : ils font partie d'une bibliothèque chargée de l'édition de liens du programme.
- de variables d'environnement : une fois positionnées, leurs valeurs sont prises en compte à l'exécution.

OpenMP versus MPI

OpenMP est MPI sont deux modèles complémentaires de parallélisation. OpenMP, comme MPI, possède une interface Fortran, C et C++. Cependant MPI est un modèle multi pro-

cessus dont le mode de communication entre les processus est explicite (la gestion des communications est à la charge de l'utilisateur). OpenMP est un modèle multitâches dont le mode de communication entre les tâches est implicite (la gestion des communications est à la charge du compilateur).

MPI est utilisé en général sur des machines multiprocesseurs à mémoire distribuée. OpenMP est utilisé sur des machines multiprocesseurs à mémoire partagée. Puisque les mémoires du calculateur utilisées sont partagées par 4 processeurs chacune, nous pouvions choir l'un ou l'autre des modèles, notre choix s'est porté sur OpenMP en raison de simplicité de codage.

CALIF : le supercalculateur de l'IRIT

Bien que les « petites » comparaisons (entre documents d'une durée de l'ordre de 1 heure) sont parfaitement exécutables sur une machine de travail standard, l'utilisation d'un serveur de calcul était indispensable à la fois pour évaluer la performance de l'algorithme en contexte parallèle et pour réaliser des expériences à plus grande échelle (enregistrements de 24heures).

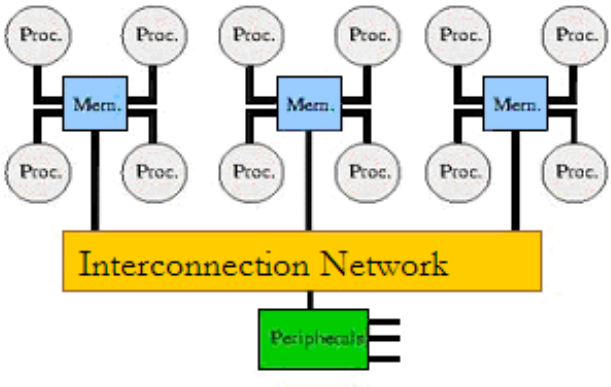


Fig. B2. L'architecture de CALIF.

Architecture de CALIF

CALIF est le serveur de calcul de l'IRIT. Ce serveur, dont le constructeur est Silicon Graphics, est composé de 12 processeurs Itanium 2 à 900MHz, répartis en 3 blocs de 4 processeurs chacun. Il possède 24 Go de mémoire réparti équitablement sur chaque bloc CPU. Il dispose d'un espace disque en interne de 8x73Go, soit 584 Go. De plus une baie de stockage spécifique lui est allouée, d'une capacité 2,3 To. Le système d'exploitation installé sur CALIF est le SGI Linux 64-bit, noyau 2.4.21, basé sur RedHat Enterprise Linux AS version 3 [CALIF].

Lancement de Job d'évaluation PBS

Pour utiliser CALIF, deux types de lancement d'exécutions (jobs) sont disponibles :

- **Interactif** : ce mode est réservé à la mise au point des applications. Les processus interactifs n'ont accès qu'à un total de 4 processeurs

- **Batch** : c'est le mode d'utilisation privilégié de la machine. Il permet d'accéder à l'ensemble des processeurs et d'utiliser pleinement ses ressources. Le planificateur de d'exécution installé est PBSpro.

- **Performance** : Tout test de performance doit se faire à travers le planificateur d'exécution PBS pour être significatif.

PBS (Portable Batch System) contrôle un certain nombre de paramètres des files d'attente définis par les administrateurs du système.

- L'affectation des jobs aux différentes files d'attente se fait suivant les ressources demandées. Si aucune exigence de ressources n'est spécifiée, des valeurs par défaut sont fixées.

- Les types de ressources qui peuvent être demandées sont : la taille mémoire, le nombre de CPU, ou le temps CPU.

- Les processeurs affectés au job sont exclusifs, et ce pour toute la durée du job,

- L'exécution des jobs est organisée suivant la politique déterminée, et les ressources de la machine disponible. En dehors des ressources demandées par l'utilisateur, PBS contrôle le nombre de jobs simultanés, afin de garantir une performance minimale aux jobs en cours d'exécution. Si le nombre maximal est atteint, le job utilisateur est mis en attente jusqu'à la libération d'une place suffisante.

Un script PBS comprend 2 parties:

- la liste des paramètres pour PBS, demande de ressources notamment. Ces lignes commencent par #PBS. Les principales limites qui peuvent être positionnées sous PBS concernent :

  - le nombre de cpus : *#PBS -l ncpus=n*, généralement pair,

  - la taille mémoire demandées : *#PBS -l mem=nu*, n=nombre, u=unité (mb,gb)

  - le temps CPU maximum de la requête : *#PBS -l cput=10:00:00*

```
#PBS -N FRANCE2
#PBS -l ncpus=8
#PBS -l cput=120:00:00
#PBS -l mem=12gb
#PBS -j oe
#PBS -M shaidar@irit.fr -m abe
cd $PBS_O_WORKDIR
./esv_F2 >& sortie_F2.txt
# -fin du script-
```

Fig. B3. Exemple d'un script PBS.

Ces commandes peuvent également servir à envoyer des ordres à PBS, comme par exemple :

- recevoir un mail à la fin du job : #PBS -M votre mail -m abe (argument : -m [option], options possibles = a, b, e ; a courrier en cas d'arrêt anormal du job, b courrier en début d'exécution, e courrier en fin d'exécution,
- donner un nom au job : #PBS -N un_nom
- fusionner les flots de sortie et d'erreur: #PBS -j oe. Dans ce cas, le flot d'erreur va sur la sortie, pour avoir l'inverse, il suffit de permuter les valeurs (#PBS -j eo)

– les commandes à exécuter.

Un exemple de paramétrage des ressources demandées est donné dans la figure B.3.

ANNEXE C : ELEMENTS SUR LE FILM MATRIX RELOADED

Critique diffusée sur le site [WebActu] par Fëanor:

Ce deuxième d'une trilogie a été critiqué d'avoir été plus spécialisé et comporte des discussions philosophiques souvent intéressantes mais un peu longues au final.

La première partie du film se perd un peu en longueurs. Là où les choses commencent réellement à s'accélérer et à devenir tout bonnement géniales, c'est dans la deuxième partie. On en apprend notamment beaucoup plus sur la matrice et le tout est parsemé de combats très agréables à voir, bien qu'un peu trop omniprésents.

Si beaucoup de personnages étaient présents dans le début de Matrix Reloaded, l'action de la deuxième heure se resserre autour des personnages principaux que sont Neo, Trinity et Morpheus. L'intervention du Mérovingien, alias Lambert Wilson, ajoute une touche d'humour au film, mais également apporte les prémices d'une opinion différente de la foi que porte Morpheus à Neo, l'Elu. Et c'est bien là que s'illustre Matrix Reloaded : l'aspect pseudo religieux, seul point un peu moins bien que le reste dans Matrix 1, subit un total retournement de situation véritablement excellent. Seul bémol, là aussi c'est la longueur de certaines scènes, comme la poursuite sur l'autoroute, qui, bien que spectaculaire, à tendance à s'éterniser ou encore la partie avec Monica Belluci.

La fin en elle-même est tout simplement l'apothéose du film, de son début en demi-teinte et de sa deuxième partie vraiment jouissive. Ce n'est pas vraiment une fin en réalité, puisque ce deuxième volet étant le milieu d'une trilogie, il n'a pas réellement de début ni de fin très prononcés.

Un film exceptionnel, dans la digne lignée de son prédécesseur qui avait révolutionné le cinéma des effets spéciaux. Si le film comprend beaucoup de dialogues philosophiques, il sait ne pas trop s'y égarer et foisonne d'effets visuels exceptionnels.

Les meilleurs jeux sont ceux qui tirent tous les partis de la plasticité inouïe du chronos virtuel, et c'est précisément ce que les frères Wachowski s'appliquent à transposer au cinéma avec leur fameux effet ralenti-accélération et l'affranchissement absolu de toutes les contraintes spatiotemporelles : Matrix décrète son temps et nous l'impose, fût-ce celui de la tétanie, et c'est sans doute une bonne part du secret de sa réussite.

L’effet Matrix d’après [Duong]

Les frères Wachowski ont demandé à John Gaeta de leur créer un effet leur permettant d'effectuer des mouvements de caméra autour du personnage filmés au ralenti et pouvoir modifier la vitesse de l'action au court d'un même plan. Par exemple, voir un personnage bondir au ralenti, puis à l'apogée de son saut donner un coup de pied fulgurant à vitesse réelle, et retomber comme une plume, au ralenti sur le sol ; et tout ceci sans coupure et avec la caméra tournant autour. Avant le film Matrix aucune technique ne permettait de réaliser une telle prise.

Pour montrer une action au ralenti, il faut que la caméra filme plus vite qu'à vitesse normale. Sachant que l'on compte aujourd'hui 25 images par seconde à vitesse normale, si on tourne à 50 images par secondes, l'action filmée va durer deux fois plus longtemps. Les caméras les plus perfectionnées permettent de filmer à 300 images par secondes, ainsi on a un ralenti de 13 fois. Mais si on filme à 300 images par secondes, il faut que le mouvement de la caméra soit 13 fois plus rapide pour que à la projection le mouvement de caméra ne soit pas au ralenti. Ces tournages nécessitent aussi des besoins en lumière beaucoup plus importants : en effet la pellicule a 13 fois moins le temps d'être impressionnée, il faut donc compenser en augmentant la lumière.

La seule solution, était l'utilisation de la photo. Il fallait photographier l'action simultanément par plusieurs appareils, et assurer une transition virtuelle entre chaque photo, grâce à l'informatique, de sorte que l'on croit à un mouvement de caméra. Ainsi toutes les scènes de ralentis virtuel ont put être pré visualisées sur ordinateur, alors les réalisateurs ont put définir avec une précision mathématique le mouvement de caméra qu'ils voulait, la distance par rapport à l'acteur, la vitesse du ralenti. Tout était réglé à l'avance sur ordinateur, ce qui leur permit une très grande flexibilité que les techniques traditionnelles n’auraient put offrir.

L’effet Matrix se résume comme suit :

1. Tout d’abord l’acteur est filmé en vidéo, jouant son jeu, sous plusieurs angles simultanément pour pouvoir décomposer les gestes de celui-ci, pour son double virtuel. Les vidéos sont ensuite numérisées ; la synchronisation entre le double virtuel et l'acteur se fait image par image.

2. Ensuite, avec l'environnement virtuel créé, les réalisateurs peuvent déterminer le timing de la séquence : point d'apparition du ralenti, reprise de vitesse normale, etc. Ces données sont ensuite traduites en nombre d'images nécessaires, puis on détermine le nombre d'appareils photos nécessaires. Les réalisateurs obtiennent ainsi, en 3D, une version simplifiée du futur plan.

3. On pré visualise le tournage réel. Le site pour le tournage est photographié sous tous les angles, puis grâce à un logiciel spécifique, on crée un décor en 3D, à partir des photos prises. Le double virtuel de l'acteur est ensuite placé dans le décor.

4. Il reste à réaliser l'étape de composition : l'acteur est photographié à 1000 images par secondes par 120 appareils qui se déclenche à un tiers de seconde l'un après l'autre. Ces appareils sont réparti en un arc de cercle qui reprend la trajectoire de la caméra virtuelle, et à chaque extrémité de cette arc, on place une caméra filmant l'avant et l'après ralenti. Toutes les photos et images filmées sont numérisées, puis mises bout à bout.

5. On établit un étalonnage de couleur et d'optique pour que le plan final soit homogène.

6. Puis on applique une interpolation, en créant les images intermédiaires reliant deux images prises par les appareils photos.

7. Enfin l'acteur est incrusté dans le décor virtuel, et on équilibre la couleur de ces deux éléments.