

Indexation et recherche de plans vidéo par le contenu sémantique

Fabrice Souvannavong

▶ **To cite this version:**

Fabrice Souvannavong. Indexation et recherche de plans vidéo par le contenu sémantique. domain_other. Télécom ParisTech, 2005. English. NNT : . pastel-00001298

HAL Id: pastel-00001298
https://pastel.hal.science/pastel-00001298

Submitted on 1 Jul 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.





Indexation et Recherche de Plans Vidéo par le Contenu Sémantique

Thèse présentée pour l'obtention du grade de docteur de Télécom Paris
dans la spécialité du traitement du signal et des images

par

Fabrice Souvannavong

Thèse soutenue publiquement le 03 juin 2005 en présence du jury composé de :

Mme Nozha BOUJEMAA	Directeur de recherche	Rapporteur
M. Georges QUENOT	Chargé de recherche	Rapporteur
Prof. Henri MAITRE	Professeur	Président
Prof. Liming CHEN	Professeur	Examinateur
M. Philippe JOLY	Maître de conférence	Examinateur
Prof. Bernard MERIALDO	Professeur	Directeur de thèse
M. Benoit Huet	Maître de conférence	Co-encadrant
Prof. Gerhard RIGOLL	Professeur	Invité

Résumé

Nous abordons dans ce mémoire le problème délicat de l'indexation de plans vidéo par le contenu et en particulier l'indexation automatique par le contenu sémantique. L'indexation est l'opération qui consiste à extraire une signature numérique ou textuelle qui décrit le contenu de manière précise et concise afin de permettre une recherche efficace dans une base de données. Une recherche est dite efficace si elle répond à la demande des utilisateurs en respectant des délais raisonnables. L'aspect automatique de l'indexation est important puisque nous imaginons bien la difficulté d'établir les signatures manuellement sur de grandes quantités de données. Jusqu'à présent les systèmes automatiques d'indexation et de recherche d'images ou de vidéos se sont concentrés sur la description et l'indexation du contenu purement visuel. Les signatures permettaient d'effectuer une recherche principalement sur les couleurs et les textures des images. A présent, le nouveau défi est d'ajouter à ces signatures une description sémantique du contenu de manière automatique.

Dans un premier temps, un éventail des techniques utilisées pour l'indexation du contenu visuel est présenté. La structure générique d'un système d'indexation et de recherche d'information est tout d'abord présentée. Cette présentation est suivie d'une introduction des principaux systèmes existants. Ensuite le support de la vidéo est décrit pour terminer sur un état de l'art des techniques d'indexation des couleurs, des textures, des formes et du mouvement.

Dans un second temps, nous introduisons une méthode pour calculer une signature précise et compacte à partir des régions des images clefs des plans. Cette méthode est une adaptation de l'analyse de la sémantique latente qui fut initialement introduite pour indexer le texte. Notre adaptation permet de capturer le contenu visuel au niveau des régions, ce qui offre la possibilité d'effectuer les recherches sur des parties des images contrairement à la majorité des méthodes existantes. Elle est ensuite comparée à une autre méthode d'indexation reposant sur les régions qui utilise la mesure EMD (ou « Earth mover's distance »). Finalement nous proposons deux méthodes d'enrichissement de notre signature pour prendre en compte la variabilité intra plan et inter échelles. En suivant la même logique nous étudions l'impact d'une boucle d'asservissement sur les résultats de la recherche d'information.

Dans un troisième temps, nous abordons la tâche difficile de la recherche par le contenu sémantique. Les expériences sont conduites dans le cadre de l'évaluation TRECVID qui nous permet d'obtenir une grande quantité de vidéo avec leurs annotations. Les vidéos annotées sont utilisées en partie pour entraîner des systèmes de classification qui permettent d'estimer le contenu sémantique de nouvelles vidéos. Nous étudions trois systèmes de classification, chacun représentant une des trois principales familles de systèmes : classification probabilistique, classification par mémoire et classification par frontière.

Dans un quatrième temps, nous poursuivons la tâche de classification sémantique en étudiant la fusion de systèmes de classification. Cette opération est nécessaire pour combiner de manière optimale les différents scores obtenus précédemment sur des caractéristiques différentes comme la couleur et la texture. Dans cet optique, nous proposons d'utiliser des opérations simples comme la somme ou le produit qui sont combinées par un arbre binaire. Cette structure d'arbre binaire permet de modéliser toutes les combinaisons possibles d'opération à deux opérandes. Les algorithmes génétiques sont utilisés afin de déterminer la meilleure structure et les meilleures opérations. Une comparaison de cette méthode avec une fusion par les machines à vecteurs de support est fournie pour montrer son efficacité. Ensuite de nouvelles modalités sont présentées (texte et mouvement) pour améliorer la classification. Nous soulevons alors le problème de la désynchronisation sémantique entre la parole et le contenu visuel. Puis à partir des scores obtenus par la fusion, nous comparons les différentes approches de recherche en fonction du type de requête : par une image exemple, par un exemple et des mots clefs ou uniquement par des mots clefs.

Finalement, ce mémoire se termine sur l'introduction d'une nouvelle méthode d'apprentissage actif. L'apprentissage actif est une technique itérative d'annotation dont l'objectif est de réduire au maximum l'effort d'annotation fourni par l'utilisateur en analysant l'impact de chaque annotation. A chaque itération le système choisit les échantillons qu'il considère les plus pertinents puis les fait annoter par l'utilisateur. Ainsi au fur et à mesure le système améliore rapidement sa connaissance des données et peut estimer les classes des données encore non annotées. Les systèmes existants ont le désavantage majeur de permettre uniquement la sélection d'un échantillon par itération sous peine d'avoir une chute de leur performance. Nous proposons une solution à ce problème et l'étudions dans le cas de l'annotation d'une classe et aussi dans le cas plus délicat de l'annotation multi-classes.

Abstract

In this thesis, we address the fussy problem of video content indexing and retrieval and in particular automatic semantic video content indexing. Indexing is the operation that consists in extracting a numerical or textual signature that describes the content in an accurate and concise manner. The objective is to allow an efficient search in a database. The search is efficient if it answers to user's needs while keeping a reasonable deadline. The automatic aspect of the indexing is important since we can imagine the difficulty to annotate video shots in huge databases. Until now, systems were concentrated on the description and indexing of the visual content. The search was mainly led on colors and textures of video shots. The new challenge is now to automatically add to these signatures a semantic description of the content.

First, a range of indexing techniques is presented. The generic structure of a content based indexing and retrieval system is presented. This presentation is followed by the introduction of major existing systems. Then, the video structure is described to finish on a state-of-the-art of color, texture, shape and motion indexing methods.

Second, we introduce a method to compute an accurate and compact signature from key-frames regions. This method is an adaptation of the latent semantic indexing method originally used to index text documents. Our adaptation allows capturing the visual content at the granularity of regions. It offers the opportunity to search on local areas of key-frames by contrary to most of existing methods. Then, it is compared to another region-based method that uses the Earth mover's distance. Finally, we propose two methods to improve our signatures and make them more robust to intra-shot and inter-scale variabilities. Following the same logic, we study the effects of the relevance feedback loop on search results.

Third, we address the difficult task of semantic content retrieval. Experiments are led in the framework of TRECVID. It allows to have a huge amount of videos and their labels. Annotated videos are used to train classifiers that allow to estimate the semantic content of unannotated videos. We study three classifiers ; each of them represents one of the major classifier family : probabilistic classifiers, memory-based classifiers and border-based classifiers.

Fourth, we pursue on the semantic classification task through the study of fusion mechanisms. This operation is necessary to combine efficiently outputs from a number of classifiers trained on various features, such as color and texture. For this purpose, we propose to use simple operations such as the sum or the product that are combined by a binary tree. This structure of binary tree allows to model all combinations of operations on two operands. Genetic algorithms are then used to determine the best structure and operators. A comparison with a fusion system based on SVM is proposed to show its efficiency. Then, new modalities (text and motion) are introduced to improve classifica-

tion performances. We, then, raise the desynchronization problem that exists between speech and visual content. From obtained detection scores, we compare different retrieval systems depending on the query type : by image example, by example and keywords or only by keywords.

Finally, this thesis concludes on the introduction of a new active learning approach. Active learning is an iterative technique that aims at reducing the annotation effort. The system selects samples to be annotated by a user. The selection is done depending on the current knowledge of the system and the estimated usefulness of samples. The system quickly increases its knowledge at each iteration and can therefore estimate the classes of remaining unlabeled data. However, current systems have the drawback to allow only the selection of one sample per iteration, otherwise their performance decreases. We propose a solution to this problem and study it in the case of one-label annotation and the more challenging task of multi-label annotation.

Remerciements

La thèse résulte d'un travail personnel dont l'aboutissement implique à différents degrés de nombreuses personnes. Je prends donc le temps de remercier dans les quelques lignes qui suivent les acteurs du bon déroulement de ces trois années et demi de doctorat.

Ce travail n'aurait pas vu le jour sans mon encadrant le Prof. Bernard Merialdo, qui m'a offert la possibilité de travailler dans le domaine de l'indexation et de la recherche de vidéos. Avec Benoit Huet qui a grandement participé à l'encadrement de mes recherches et à la retranscription de mes travaux, ils m'ont permis de mener à bien mes recherches en m'apportant leur expertise tout en me laissant une grande liberté.

J'ai déjà pu remercier les membres de mon jury mais ils méritent une redite. Je remercie donc Nozha Boujemaa et Georges Quenot d'avoir accepté d'être les rapporteurs de mon mémoire. Je remercie le Prof. Liming Chen, Philippe Joly et le Prof. Henri Maitre d'être les examinateurs de mon jury. Et finalement je remercie le Prof. Gerhard Rigoll d'avoir accepté d'être un membre invité.

La suite des remerciements est destinée à mon entourage. Alex, néophyte dans le domaine, a bien voulu relire et corriger mon mémoire. Mes co-bureaux, Joakim, Lukas, Daniele et Eric m'ont toujours apporté une grande quiétude propice à la réflexion. Gwen et Caro m'ont permis de libérer la pression au quotidien et en particulier les week-ends lors d'expéditions ou de soirées. Gwen a également toujours été dévoué à trouver des réponses aux problèmes que je pouvais lui poser. Je remercie le personnel d'Eurécom en général et notamment ceux qui vous saluent avec le sourire et apporte leur bonne humeur dans leur travail et pendant les repas ou les pauses. Je remercie Evandro et Cuong de m'avoir fait découvrir la montagne. L'ambiance de la montagne et leur compagnie ont très certainement contribuées à ma quiétude physique et morale. Pour terminer je remercie Marie, ma future femme, pour son sourire inaltérable, sa présence et son attention quotidiennes, et pour tout le reste.

Les derniers remerciements vont à ceux que je n'ai pas cités et qui pourtant le méritent. Je suis sûr qu'ils se reconnaîtront et c'est le principal.

Sommaire

Résumé i

Abstract iii

Remerciements v

Sommaire vii

Liste des figures xi

Liste des tableaux xv

Liste des abbréviations xvii

Introduction 1

Bibliographie 4

1 Indexation et recherche par le contenu 5

1.1 Description générale du système 6

1.1.1 Indexation 6

1.1.2 Recherche 6

1.1.3 Systèmes existants 7

1.2 Représentation numérique du signal vidéo 8

1.3 Segmentation temporelle 10

1.3.1 Segmentation en plans 11

1.3.2 Segmentation en scènes 11

1.3.3 Sélection de l'image représentative 12

1.4 Description du plan 13

1.4.1 Description visuelle d'une image 13

1.4.2 Description du mouvement . . . 15
1.5 Conclusion . . . 16
Bibliographie . . . 18

2 Utilisation des régions pour l'analyse du contenu 23
2.1 Description du plan par des régions . . . 24
2.1.1 Segmentation spatiale . . . 24
2.1.2 Description des régions . . . 24
2.1.3 Dictionnaire visuel . . . 26
2.2 Description du contenu latent . . . 27
2.3 Méthodes d'évaluation . . . 31
2.3.1 Mesures d'évaluation . . . 31
2.3.2 Les données utilisées . . . 31
2.3.3 Recherche . . . 33
2.4 Evaluation des performances . . . 33
2.4.1 Choix du nombre de mots clefs visuels . . . 33
2.4.2 Impact de la LSA . . . 34
2.4.3 Comparaison avec l'EMD . . . 36
2.5 Utilisation d'une représentation floue . . . 38
2.6 Description localisée du contenu latent . . . 41
2.6.1 Analyse locale du contenu latent . . . 41
2.6.2 Indexation et recherche . . . 41
2.6.3 Expériences . . . 43
2.7 Description enrichie du plan . . . 43
2.7.1 Méthodes d'enrichissement . . . 43
2.7.2 Asservissement . . . 44
2.7.3 Expériences . . . 44
2.8 Conclusion . . . 49
Bibliographie . . . 50

3 Analyse du contenu sémantique des plans vidéo 53
3.1 La base de données TRECVID . . . 54
3.2 Classification . . . 56
3.2.1 Objectif . . . 56
3.2.2 Un aperçu de la fusion . . . 57
3.2.3 Evaluation des systèmes de classification . . . 57
3.3 Modèles utilisés . . . 58

3.3.1 Mélange de Gaussiennes . . . 58
3.3.2 Classification par mémoire . . . 62
3.3.3 Machine à vecteurs de support . . . 65
3.4 Analyse du problème de classification . . . 67
3.4.1 Intérêts de la LSA . . . 69
3.4.2 La classification de plans . . . 70
3.5 Conclusion . . . 71
Bibliographie . . . 72

4 Fusion de systèmes de classification 73
4.1 La fusion . . . 73
4.1.1 Les difficultés de la fusion . . . 74
4.1.2 La fusion par les SVMs . . . 74
4.1.3 La fusion par les algorithmes génétiques . . . 75
4.1.4 Les résultats . . . 79
4.2 Introduction du texte et du mouvement . . . 80
4.2.1 Texte . . . 80
4.2.2 Mouvement . . . 84
4.3 Etude de la fusion . . . 87
4.3.1 Comparaison des systèmes de fusion sur plusieurs modalités . . . 87
4.3.2 Fonctions de fusion de taille variable . . . 87
4.3.3 Recherche mixte . . . 88
4.4 Conclusion . . . 89
Bibliographie . . . 91

5 Apprentissage actif 93
5.1 L'apprentissage actif . . . 93
5.1.1 Présentation . . . 93
5.1.2 Sélection par incertitude . . . 94
5.1.3 Système de classification . . . 96
5.2 Résultats préliminaires . . . 96
5.2.1 Les données . . . 96
5.2.2 Evaluation des systèmes d'apprentissage actif . . . 98
5.2.3 Expériences . . . 98
5.3 Sélection par partitionnement . . . 100
5.3.1 Théorie . . . 100
5.3.2 Implémentation . . . 101

5.4 Evaluation 101
5.4.1 Evaluation du partitionnement 102
5.4.2 Evaluation de la sélection 102
5.5 Annotation de plusieurs classes 103
5.6 Conclusion 105
Bibliographie 107

6 Conclusion **109**
6.1 Résumé 109
6.2 Perspectives 110
6.2.1 Amélioration de la capture de l'information visuelle 110
6.2.2 Fusion 112
Bibliographie 114

A Filtres de Gabor **115**

B Extraits des nouvelles télévisées **117**

C Logiciel de recherche de plans vidéo **119**

Liste des figures

1.1 Phase d'indexation par le contenu 7
1.2 Phase de recherche par le contenu 7
1.3 Principe général du codage MPEG. 9

2.1 Rendu de la segmentation automatique obtenue avec trois valeurs du paramètre K (100, 200 et 300) qui module la finesse de la segmentation. 25
2.2 Forme des enveloppes des filtres de Gabor dans le domaine fréquentiel. 4 échelles et 6 orientations. 27
2.3 Représentation du contenu reposant sur les régions. 28
2.4 Principe de l'analyse du contenu latent. 30
2.5 Objets manuellement sélectionnés pour l'évaluation. Source : dessins animés fournis par la production Docon à la bibliothèque MPEG-7 32
2.6 Choix du nombre de mots clefs visuels pour la quantification. 34
2.7 Evolution de la précision moyenne en fonction du nombre de mots clefs visuels et du pourcentage de composantes actives (dureté de la LSA). 35
2.8 Résultats sur des plans de journaux télévisés TRECVID avec 2 000 mots clefs visuels. 35
2.9 Etude de la mesure EMD et comparaison avec le VSM et la LSA. 37
2.10 Comparaison détaillée du système utilisant l'EMD et du système utilisant la LSA . 38
2.11 Fiabilité de la quantification par 1 000 mots clefs visuels sur des dessins animés . . 39
2.12 Fiabilité de la quantification par 2 000 mots clefs visuels sur les plans de journaux télévisés TRECVID 40
2.13 Quantification floue et performance de l'analyse de la sémantique latente 40
2.14 Performances obtenues à l'aide d'une description localisée du contenu latent pour $\gamma = 2$. 43
2.15 Utilisation d'une boucle d'asservissement avec une itération sur la série de dessins animés. 45

2.16 Utilisation d'une boucle d'asservissement et sélection des objets dans les images clefs retournées sur la série de dessins animés. 46
2.17 Utilisation d'une boucle d'asservissement avec 10 plans visualisés et une itération sur les plans de journaux télévisés. 47
2.18 Evolution des performances en fonction d'une nombre d'itérations pour 10% de composantes conservées par la LSA sur les plans de journaux télévisés. 47
2.19 Sélection aléatoire des plans visualisés sur la série de dessins animés. 48

3.1 Principe de la classification supervisée. 56
3.2 Modélisation d'une classe par un mélange de gaussiennes à deux composantes. *Illustration en une dimension. Les points en haut de la figure sont les points de la classe à modéliser. La courbe en pointillés représente la densité de probabilité estimée. Les deux autres courbes sont les composantes gaussiennes de cette estimation. L'ordanncement des plans est effectué en fonction de la valeur de la densité de probabilité au point correspondant.* 59
3.3 Classification par les mélanges de Gaussiennes 62
3.4 Classification par les quatre plus proches voisins. *La classification d'un nouvau plan se fait par la recherche de ces quatres plus proches voisins, puis par héritage de leurs classes. Les valeurs obtenues permettent ensuite d'ordonner les plans.* . . 63
3.5 Classification par les k-plus proches voisins. 65
3.6 Machines à vecteurs de support. *L'entraînement consiste à identifier « l'hyperplan » qui sépare au mieux les deux classes en maximisant la marge et en minimisant les erreurs ϵ. Un nouveau plan est classé en fonction de sa distance signée à « l'hyperplan ». Cette dernière permet ensuite de les ordonner.* 66
3.7 Classification par les machines à vecteurs de support. 67
3.8 Impact de l'optimisation de la dureté de la LSA. 69

4.1 Exemple d'une fonction du fusion basée sur des opérations simples. 76
4.2 Exemple d'une fonction de fusion décrite par un chromosome. 77
4.3 L'identité de Rémy pour ajouter ou supprimer une feuille. 78
4.4 Opérateurs génétiques conservant l'existence et l'unicité des valeurs. 79
4.5 Fusion des informations de couleur et de texture. 80
4.6 Structure de la fonction de fusion pour deux classes. 81
4.7 Classification du texte. 82
4.8 Détection des mots clefs. 84
4.9 Classification du mouvement des plans. 86

4.10 Fusion des différents systèmes de classification et des différentes modalités et caractéristiques. 88
4.11 Comparaison des performances de la fusion sur des arbres de taille fixe et de taille variable. 89
4.12 Comparaison des différents modes de recherche. *Recherche à partir d'une image exemple sans et avec une boucle d'asservissement. Recherche uniquement sur la classe. Recherche mixte à l'aide d'un exemple et d'une classe.* 90

5.1 Principe de l'apprentissage actif par échantillonnage sélectif : *Méthode d'annotation itérative qui permet de réduire le nombre d'annotation à effectuer. Elle demande uniquement l'annotation des échantillons qui sont estimés les plus informatifs.* 95
5.2 Ensembles synthétiques générés pour étudier le comportement de l'apprentissage actif. *Le premier ensemble est composé de classes bien distinctes alors qu'elles sont plus entrelacées dans le second.* 97
5.3 Apprentissage actif sur les ensembles synthétiques. 99
5.4 Apprentissage actif sur deux concepts sémantique de TRECVID. 100
5.5 Influence du nombre de partitions sur les performances de l'apprentissage. 102
5.6 Etude de la nécessité d'effectuer le partitionnement à chaque itération. 103
5.7 Apprentissage actif avec une stratégie par incertitude et partitionnement sur les ensembles synthétiques. *Comparison des performances.* 104
5.8 Apprentissage actif avec une stratégie par incertitude et partitionnement sur les vidéos de TRECVID. *Comparison des performances.* 105
5.9 Apprentissage actif sur un ensemble synthétique et sur un problème multi classes. . 106
5.10 Apprentissage actif sur la base de données TRECVID et sur un problème multi classes. 106

6.1 Comparaison de l'approche sans structure et de l'approche avec structure sur les vidéos de nouvelles de TRECVID 112

Liste des tableaux

3.1 Répartition des plans vidéo dans les différents ensembles par concept sémantique. *Le quantité relative de chaque classe de l'ensemble de test dans l'ensemble de test est précisée pour donner une idée de la borne inférieure des performances à obtenir (précision moyenne avec un ordonnancement aléatoire).* 55
3.2 Pourcentage de composantes conservées par concept sémantique. Classification par les mélanges de gaussiennes. 61
3.3 Taille du voisinage suivi du pourcentage de composantes conservées par concept sémantique. Classification normée par les k plus proches voisins. 64
3.4 Pourcentage de composantes conservées par concept sémantique. Classification par les SVM. 68

4.1 Occurrence des mots clefs autour du plan courant. 83

Liste des abbréviations

DCT	Discreet cosine transform	transformée discrète en cosinus
EMD	Earth mover's distance	
FFT	Fast fourrier transform	transformée de fourrier rapide
GMM	Gaussian mixture model	Mixture de gaussiennes
IVSM	Image vector space model	Modèle d'espace vectoriel appliqué aux images
KNN	K-nearest neighbors	k-plus proches voisins
LSA	Latent semantic analysis	Analyse de la sémantique latente
MPEG	Motion picture expert's group	
PCA	Principal components analysis	analyse des composantes principales
SVD	Singular value decomposition	Décomposition en valeurs singulières
SVM	Support vector machine	Machine à vecteurs de support
VSM	Vector space model	Modèle d'espace vectoriel

Introduction

« Mieux vaut une tête bien faite qu'une tête bien pleine. »
Montaigne.

Motivations

Comment s'y retrouver dans ces montagnes de documents multimédias amassés par le réseau Internet ? Comment s'y retrouver parmi les milliers de reportages réalisés chaque jour ? Comment s'y retrouver dans le dédale de nos archives personnelles de photographies et de vidéos ? Des brides de réponses sont apparues ces dernières années mais un long chemin reste à parcourir pour indexer convenablement et de manière automatique toutes ces images et ces vidéos qui s'agglutinent sans ordre dans notre quotidien.

L'indexation est une pratique ancienne indispensable pour retrouver rapidement les documents voulus. Jusqu'à une époque récente, elle semblait réservée à l'intelligence humaine. Car indexer ne consiste pas à créer des index (tâche facilement automatisable) mais à affecter aux documents des indices, des marques significatives de leur contenu, à la suite d'une série d'opérations mentales complexes et encore mal connues. Les dernières recherches en traitement informatique des langues (traduction automatique) et en sémantique (analyse conceptuelle, réseaux sémantiques, analyseur automatique de texte) ont mis à la disposition des concepteurs des outils efficaces pour les documents textuels.

Les avancées technologiques ont permis aux professionnels et aux particuliers de numériser et stocker sans limite de nombreux documents qui ne se limitent plus au texte mais qui incluent à présent la photo et la vidéo. Le moindre accessoire est maintenant capable d'acquérir de petites séquences d'images de notre quotidien. Et le tout peut facilement être partagé au travers des réseaux Internet ou UMTS. Nous assistons à une numérisation quotidienne de notre environnement et à la création d'une multitude de documents multimédias qui, noyés dans la masse, deviennent difficilement utilisables.

De nouveaux outils puissants et automatiques d'indexation et de recherche par le contenu sont vivement attendus dans de nombreux domaines allant de la vidéo sur demande à la télé éducation en passant par les systèmes multimédias distribués, les bases de données multimédias, les jeux, la surveillance, le commerce électronique et les systèmes d'information géographique.

De nombreux systèmes ont été proposés durant les dix dernières années pour répondre à la demande croissante de systèmes automatiques d'indexation et de recherche par le contenu. Et nous assistons aujourd'hui à une grande évolution de ces systèmes de recherche.

Cadre de la thèse

Jusqu'à présent, les systèmes de recherche par le contenu s'efforçaient de retrouver des documents ayant les mêmes caractéristiques que la requête. Cette dernière pouvait être un dessin ou une image. Et les caractéristiques extraites automatiquement comprenaient les informations de couleur, de texture, de forme et de mouvement. Malheureusement ces systèmes précurseurs permettaient de retrouver uniquement des documents visuellement similaires et n'effectuaient pas une réelle recherche sémantique. Par exemple, si vous recherchez des photos contenant un coucher du soleil, vous souhaitez sûrement avoir comme réponse une collection de photos avec des couchers du soleil sous différentes conditions : ciel dégagé/ciel nuageux, en bord de mer/en montagne, ... et non pas uniquement les couchers du soleil dans les mêmes tons que votre requête. Actuellement, la nécessité de formuler des requêtes et d'effectuer des recherches sémantiques s'intensifie. Et les nouveaux systèmes automatiques de recherche d'information par le contenu devront franchir l'abîme séparant le contenu visuel du contenu sémantique.

Le travail de thèse présenté dans ce mémoire se place donc dans le contexte très actif de la recherche d'information par le contenu sémantique où la recherche de plans vidéo nous intéresse particulièrement. Nous allons voir dans la prochaine partie comment le sujet est traité et les contributions qui sont apportées.

Aperçu et Contributions

Dans notre étude nous avons suivi la logique d'un système automatique de recherche d'information.

Le second chapitre commence par présenter la structure des systèmes automatiques de recherche d'information. Ensuite il poursuit par la présentation du support sur lequel nous allons travailler, c'est à dire la vidéo. Puis il termine par un état de l'art sur les différentes méthodes existantes pour indexer et rechercher des images et des vidéos en utilisant les similarités visuelles.

Dans le troisième chapitre, nous introduisons l'analyse du contenu latent (Souvannavong et al. [4]). C'est une méthode que nous avons empruntée à l'indexation de documents textuels. Elle se place dans le cadre d'une indexation des régions segmentées d'une image. Une comparaison avec une autre méthode de la même catégorie sera fournie. Puis le chapitre se termine sur trois extensions de la méthode de l'analyse du contenu latent (Souvannavong et al. [5, 6]).

Le quatrième chapitre aborde le sujet délicat de la classification sémantique des plans vidéo. Nous allons comparer trois méthodes de classification pour estimer les concepts sémantiques présents dans les plans vidéos (Souvannavong et al. [7]).

Dans le cinquième chapitre, nous proposons une nouvelle méthode de fusion modélisée par des arbres binaires complets. Nous verrons comment les algorithmes génétiques sont utilisés pour trouver la meilleure fonction de fusion (Souvannavong et al. [9]).

Le sixième chapitre s'attaque à la difficulté de la tâche d'annotation qui est nécessaire pour construire les modèles précédents. Pour cela, nous proposons un nouvel algorithme d'apprentissage actif qui permet de limiter l'effort de l'annotation (Souvannavong et al. [8, 10]).

Le septième chapitre est consacré aux perspectives de recherches ouvertes par ces travaux. Tout d'abord nous débutons par un bref résumé de ce mémoire. Ensuite nous developpons deux axes

principales de recherche qui sont l'amélioration de la capture du contenu visuel et la fusion de la multitude des sources d'information disponibles. En particulier nous présentons nos travaux préliminaires sur l'inclusion du voisinage des régions pour décrire plus fidèlement le contenu (Hohl et al. [1, 2], Souvannavong et al. [3]).

Bibliographie

[1] Lukas Hohl, Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Enhancing latent semantic analysis video object retrieval with structural information. In *Proceedings of the IEEE International Conference on Image Processing*, 2004.

[2] Lukas Hohl, Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Using structure for video object retrieval. In *International Conference on Image and Video Retrieval*, 2004.

[3] Fabrice Souvannavong, Lukas Hohl, , Bernard Merialdo, and Benoit Huet. Structurally enhanced latent semantic analysis for video object retrieval. *(to appear) IEE Proceedings Vision, Image and Signal Processing*, (submitted in september 2004).

[4] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Video content modeling with latent semantic analysis. In *Third International Workshop on Content-Based Multimedia Indexing*, 2003.

[5] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Improved video content indexing by multiple latent semantic analysis. In *International Conference on Image and Video Retrieval*, 2004.

[6] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Latent semantic analysis for an effective region-based video shot retrieval system. In *Proceedings of the ACM International Workshop on Multimedia Information Retrieval*, 2004.

[7] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Latent semantic analysis for semantic content detection of video shots. In *Proceedings of the International Conference on Multimedia and Expo*, 2004.

[8] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Partition sampling for active video database annotation. In *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services*, 2004.

[9] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Multi-modal classifier fusion for video shot content retrieval. In *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services*, 2005.

[10] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Partition sampling : an active learning selection strategy for large database annotation. *(to appear) IEE Proceedings Vision, Image and Signal Processing*, (submitted in june 2004).

Chapitre 1

Indexation et recherche par le contenu

L'aire du numérique a énormément changé le visage du monde de l'indexation et de la recherche de documents. L'exemple le plus commun de cette mutation s'observe dans les bibliothèques scientifiques.

Pour ordonner les documents de leur bibliothèque, les documentalistes font appel aux techniques de l'indexation. L'indexation consiste à identifier dans un document certains éléments significatifs qui serviront de clef pour retrouver ce document au sein d'une collection. Le problème de savoir comment choisir l'indexation d'un document est particulièrement complexe ; en général, le titre ne fait guère plus que mettre en relief un ou deux mots importants. La classification figure depuis longtemps parmi les outils fondamentaux de la méthode scientifique. Ainsi, pour ordonner de façon systématique l'ensemble des collections d'une bibliothèque, il faut comprendre, en théorie et en pratique, comment les connaissances humaines sont structurées et comment il convient de grouper les documents afin de montrer les relations qu'il y a entre leurs sujets, ce qui aide le lecteur à mieux comprendre le classement et à mieux utiliser la collection.

Le traitement informatique a permi de simplifier et de rendre plus efficace la recherche de documents écrits. A présent la recherche n'est plus effectuée uniquement par catégorie et il n'est plus nécessaire de connaître le fonctionnement des systèmes d'indexation comme la classification décimale de Dewey. L'indexation informatique offre la possibilité d'effectuer une requête directement à partir de mots clefs. La recherche est ensuite réalisée sur le titre, les mots clefs associés aux documents, les auteurs, la date et aussi sur la table des matières et le résumé. Le système fournit alors une liste des documents les plus pertinents avec leurs références pour une identification rapide dans la bibliothèque. Le traitement informatique a repoussé encore plus loin les limites des systèmes d'indexation manuelle en offrant la possibilité d'indexer les documents dans leur intégralité. Un bel exemple de cette extension de la recherche est fourni par Internet et ses moteurs de recherche. De plus la numérisation des documents permet un accès rapide et direct en ligne ainsi qu'une consultation immédiate et partagée.

L'indexation et la recherche sur le texte sont maintenant automatiques, rapides et efficaces. Toutefois ils restent en constante évolution pour permettre de gérer des collections de documents toujours plus importantes. Depuis quinze ans environ, le nouveau défi est d'étendre ces systèmes automatiques à la recherche d'images et plus récemment de vidéos. En effet ces supports de communication ont aussi bénéficié des progrès de l'informatique et ils occupent une grande place dans

notre quotidien. Malheureusement leur indexation automatique est un problème délicat.

Dans ce chapitre, nous décrivons tout d'abord la structure générale d'un système de recherche d'information par le contenu. Ensuite un état de l'art des techniques d'indexation de l'image et de la vidéo est donné au travers de l'étude du support numérique de la vidéo puis des différentes méthodes d'extraction des caractéristiques visuelles du signal.

1.1 Description générale du système

Les systèmes de recherche d'information basés sur le contenu, communément appelés CBIR (acronyme de « Content Based Information Retrieval »), font intervenir deux phases qui sont l'indexation et la recherche. Nous commençons par une présentation des ces deux phases avant de poursuivre par une rapide énumération des systèmes existants aujourd'hui.

1.1.1 Indexation

L'indexation consiste à extraire, représenter et organiser efficacement le contenu des documents d'une base de données (figure 1.1). Pour cela, les documents sont tout d'abord représentés par une signature numérique qui permettra leur identification sous certains critères. Cette opération est réalisée en deux étapes. La première étape implique l'analyse des documents pour en extraire l'essentiel. Il s'agira par exemple de capturer les couleurs ou les textures caractéristiques, de déterminer les visages ou les objets présents. La seconde étape permet éventuellement de compresser l'information extraite tout en conservant l'essentiel. Il est important d'avoir des signatures compactes pour éviter d'avoir des données trop importantes à stocker et à traiter. Finalement, les signatures sont organisées au mieux afin d'optimiser la recherche de l'information. Dans la plupart des cas, une structure hiérarchique est utilisée pour organiser et faciliter l'accès aux signatures pertinentes. Habituellement une structure en arbre R* (ou « R*-tree ») est adoptée (Beckmann et al. [4]).

Actuellement, la communauté scientifique du domaine ne se restreint plus à la capture du contenu visuel mais elle travaille de plus en plus sur l'inclusion automatique du contenu sémantique dans les signatures (Li and Wang [28]). La détection des objets et des visages est déjà un premier pas dans ce sens ; et l'avancée se poursuit en détectant automatiquement des concepts de plus haut niveau qui vont permettre une réelle recherche sur le contenu similaire à celle réalisée sur les documents écrits. Les futures bases de données contiendront alors des signatures représentant le contenu brut et des signatures représentant explicitement le contenu sémantique. Et cette indexation mixte sera réalisée automatiquement.

1.1.2 Recherche

La recherche d'information est l'ensemble des opérations nécessaires pour répondre à la demande d'un utilisateur (figure 1.2). Tout d'abord, l'utilisateur doit construire une requête. Cette opération évidente pour le texte est bien plus difficile pour les images et encore plus pour la vidéo. La requête peut inclure différentes données : un exemple (image, vidéo, son), un dessin ou une animation. Dans le cas plus rare où des signatures sémantiques sont disponibles, l'utilisateur peut alors inclure des mots clefs (Lu et al. [32]). En règle générale la difficulté est d'exprimer correctement l'objet de la requête en utilisant au mieux les moyens proposés par le système.

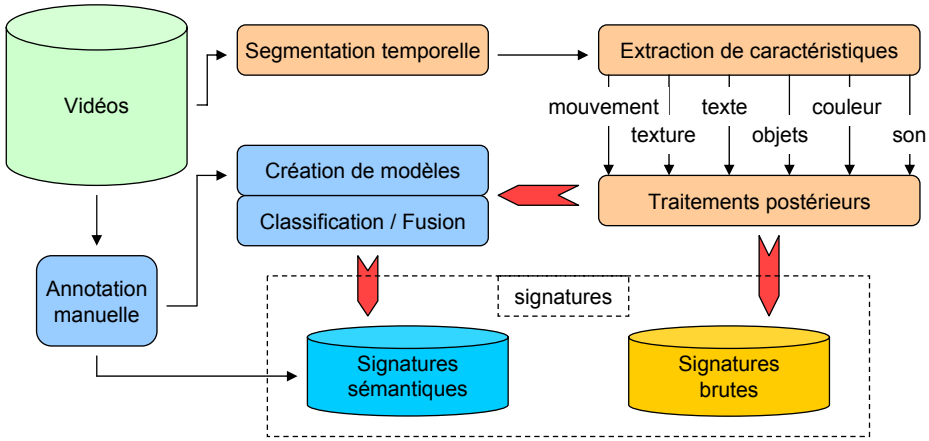


FIG. 1.1 – Phase d'indexation par le contenu

La requête est ensuite transformée en signature en suivant un procédé similaire à l'indexation. Cette signature est alors comparée aux signatures de la base de données afin de retrouver l'information la plus pertinente. Toutefois il est particulièrement difficile de répondre aux exigences des utilisateurs à partir d'une seule requête. Il est alors utile d'intégrer un bouclage de pertinence incluant l'avis de l'utilisateur pour améliorer la requête en fonction du résultat précédemment obtenu. Un tel système permet également à l'utilisateur de clarifier sa demande qui est souvent mal formulée au début de la recherche.

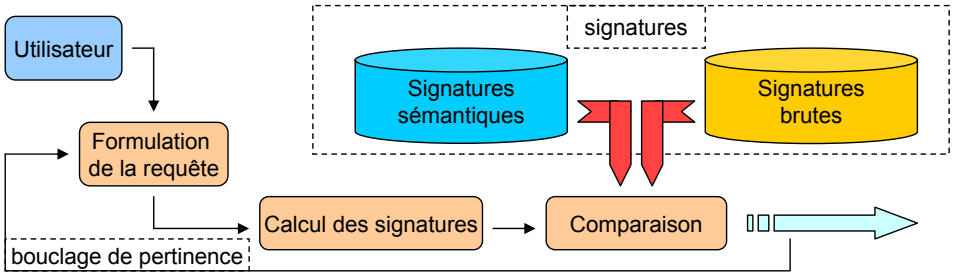


FIG. 1.2 – Phase de recherche par le contenu

1.1.3 Systèmes existants

Depuis le début des années 90 l'indexation et la recherche d'image par le contenu sont devenues un pôle très actif de la recherche et de nombreux systèmes commerciaux et académiques ont été proposés. Puis rapidement des extensions sont apparues pour réaliser des systèmes d'indexation et de recherche de vidéos. A présent l'avancée technologique continue avec l'intégration automatique du contenu sémantique pour accéder à une recherche de plus haut niveau.

Il serait fastidieux et inutile dans le cadre de ce mémoire d'entrer dans une analyse détaillée et

exhaustive des nombreux systèmes existants. Les prochaines sections aborderont plus en profondeur les techniques d'indexation et de recherche en général. Les systèmes les plus populaires sont clairement présentées par (Rui et al. [42]). Dans la suite, nous abordons la présentation des principaux systèmes en s'efforçant de mettre en valeur les évolutions réalisées au fil du temps.

Les premiers systèmes introduits concernaient l'indexation et la recherche d'images. La recherche était principalement effectuée sur des exemples fournis par l'utilisateur. Les caractéristiques visuelles sont alors extraites sur la requête puis comparées à celles extraites sur les images de la base de données. Dans Photobook (Pentland et al. [39]) la requête est formulée à partir d'une image exemple. Ensuite l'utilisateur choisit parmi trois modules d'analyse pour effectuer la recherche : visage, forme ou texture. Le système QBIC (Faloutsos et al. [12]) est plus complet puisqu'il permet d'effectuer une recherche sur l'ensemble des caractéristiques de couleur, de forme et de texture. De plus les requêtes peuvent être formulées directement, c'est à dire que l'utilisateur peut faire un dessin et sélectionner des couleurs et des textures dans une palette. Virage (Bach et al. [2]) va plus loin en autorisant la combinaison de plusieurs types de requêtes. Le système VisualSeek (Smith and Chang [44]) ajoute aux caractéristiques de couleur et de texture une contrainte spatiale sur les régions. L'agencement des régions de la requête est alors pris en compte par le module de recherche.

Les mêmes techniques sont ensuite employées sur les vidéos. Pour cela, les séquences sont découpées en plan puis les images clefs des plans sont retenues pour l'indexation et la recherche. Ces deux opérations feront l'objet d'une recherche active comme nous le verrons dans la prochaine section. Le système Informedia (Wactlar et al. [51]) exploite réellement l'information du flux vidéo en détectant le mouvement de la caméra et en réalisant une reconnaissance vocale automatique. Le système Netra-V (Yining and Manjunath [57]) utilise le mouvement pour obtenir une segmentation spatio-temporelle et réaliser une indexation des objets sur leur couleur, leur texture, leur forme, leur position et leur mouvement dans l'image clef. VideoQ (Chang et al. [8]) généralise l'approche en modélisant la trajectoire des objets. L'utilisateur peut également dessiner une requête animée contenant plusieurs objets aux mouvements différents. Finalement ClassView (Jianping et al. [22]) introduit la notion de contenu sémantique. Dans un contexte spécifique, des modèles sont construits pour classer les plans dans les catégories proposées par le contexte. La recherche est ensuite effectuée conjointement sur les caractéristiques primaires et sémantiques. ViBe (Taskiran et al. [47]) emploie une approche similaire.

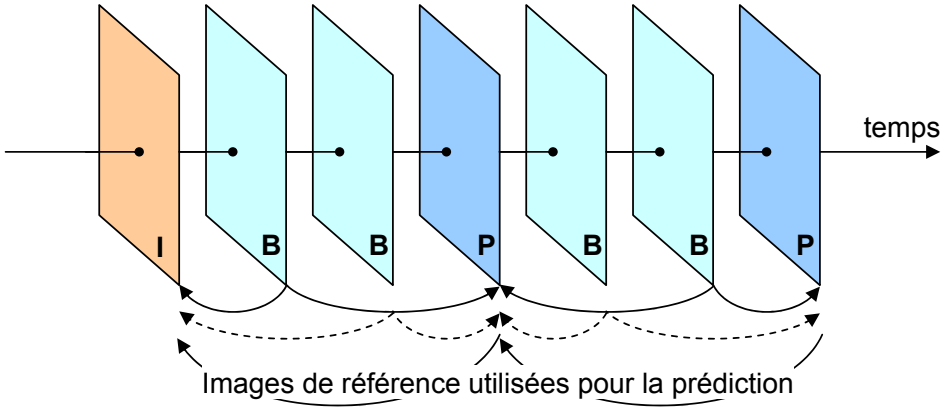
Nous avons vu l'évolution d'ensemble des systèmes d'indexation. Les sections suivantes traitent plus particulièrement des caractéristiques qui peuvent être extraites des images ou des vidéos. Pour cela, la prochaine section commence par présenter le support numérique de la vidéo.

1.2 Représentation numérique du signal vidéo

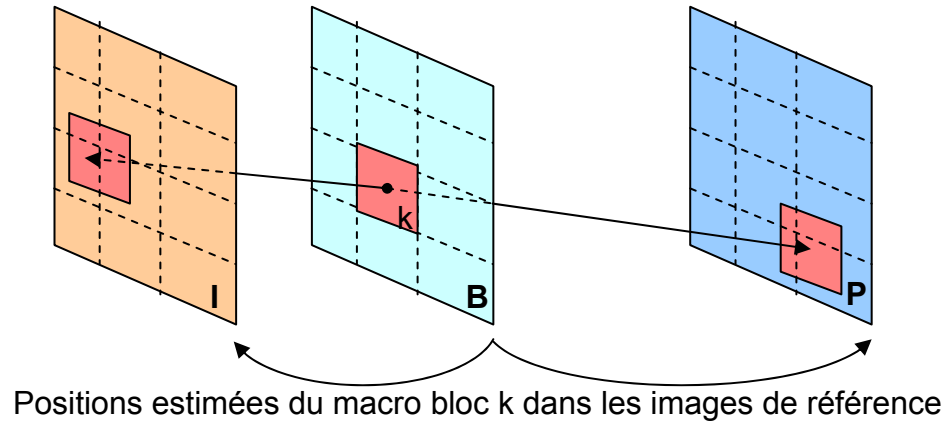
Nous n'attardons pas notre étude sur le codage des images. En effet le nombre de formats existants est important et de ce fait le traitement est le plus souvent réalisé sur l'image brute, c'est à dire une succession de pixels avec une information de couleur. Par contre les formats de codage vidéo sont moins nombreux et les normes MPEG-1 et MPEG-2 ont été largement adoptées. Un intérêt particulier a donc été porté sur le traitement direct du signal compressé.

Les vidéos contiennent deux sources supplémentaires d'information par rapport à l'image et à la photographie. Il s'agit du mouvement et du son. Toutefois comme le cerveau est plus sensible au

mouvement au détriment des détails, les images composant une vidéo ne requièrent pas la même qualité ni la même résolution qu'une photographie. Les systèmes de codage et de compression ne se privent pas de cette observation pour numériser une vidéo. Il en résulte le gain de deux sources d'information et la dégradation de la qualité visuelle. Les traitements sur le contenu des vidéos deviennent alors plus délicats.



(a) Ordonancement des images MPEG et position des images de référence.



(b) Estimation du mouvement d'un macro bloc d'une image de type B

FIG. 1.3 – Principe général du codage MPEG.

La plupart des systèmes de codage ajoutent au codage basique des images un système prédictif sur une séquence d'images. Ainsi uniquement certains types d'images sont entièrement codés (images de type I) tandis que les paramètres de la prédiction et son erreur sont codés pour les images des types restants (images de types B et P). Nous allons principalement travailler sur des vidéos encodées aux formats MPEG-1 ou MPEG-2 qui sont les formats les plus répandus. Sans entrer dans les détails, le codage MPEG fait intervenir 3 types d'images nommées I, P et B. I signifie intra codée, P prédictive et B bidirectionnelle. Chaque image indépendamment de son type

est découpé en macro blocs de taille 16×16 pixels. Ces macro blocs vont servir pour le codage par la transformée en cosinus discrète (ou « Dicrete Cosine Transform » : DCT) des images I mais également pour les prédictions. Les images I sont codées d'une manière proche des images JPEG. La DCT est appliquée à chaque macro bloc pour chaque canal de couleur (YUV). Ensuite une quantification est appliquée pour réaliser une partie de la compression avec perte. Les macro blocs des images P et B sont estimés puis les erreurs sont compensées. Pour cela le vecteur mouvement de chaque macro bloc d'une image P indique sa position estimée dans l'image I ou P précédente comme le montre les schémas 1.3. Dans le cas des images B, deux vecteurs mouvements sont disponibles par macro bloc : le premier indiquant la position du macro bloc dans l'image I ou P précédente et le second indiquant la position du macro bloc dans l'image I ou P suivante. Les erreurs sont ensuite codées pour chaque canal après l'application de la transformée en cosinus discrète. Habituellement, une séquence d'images d'une seconde est composée de la manière suivante : **I**BB*P*BB*P*BB*P*BB**I**BB*P*BB*P*BB*P*BB*P*.

Plus tard nous verrons que ces caractéristiques du codage MPEG sont utilisées pour mettre au point des méthodes de traitement de la vidéo travaillant directement dans le flux compressé. Ces méthodes ont l'avantage d'éviter partiellement les lourdes/coûteuses opérations de décompression tout en travaillant sur un ensemble de données moins grand. Mais avant de se précipiter sur l'analyse du contenu des vidéos, il est important d'optimiser la capture de ce contenu. En effet il n'est pas raisonnable d'indexer toutes les images composant une vidéo. A titre indicatif, un film d'1h30 contient 129 600 images à raison de 24 images par secondes. Pour cela, jetons un coup d'oeil aux propriétés intrinsèques d'un document audio-visuel. En particulier, voyons comment il peut être découpé de manière intelligente pour réduire la quantité des données à traiter tout en capturant l'essentiel de l'information.

1.3 Segmentation temporelle

Un film est réalisé de manière structurée. Nous pouvons définir deux phases qui sont le tournage et le montage. La première phase consiste à filmer ce que nous appelons des plans. Un plan est défini par une séquence d'images durant laquelle l'acquisition du signal est continue. L'acquisition de ces plans se fait dans une zone limitée de l'espace où se situe et se joue/déroule l'action. La phase du montage consiste à regrouper les plans en scènes puis à assembler les scènes pour former le film. La scène est définie comme la plus petite unité sémantique d'un film. Elle regroupe par définition l'ensemble des plans consécutifs se situant dans le même espace, au même instant ou ayant un lien sémantique étroit. Il s'agira par exemple d'un dialogue, d'une conversation téléphonique, d'une action observée de plusieurs points de vue, de deux évènements se produisant en parallèle, ...Le montage se termine ensuite par la juxtaposition des scènes pour former le récit. L'analyse du contenu d'une vidéo passe alors par sa segmentation temporelle, tout d'abord en plans puis en scènes afin de mener une analyse cohérente et efficace. Nous verrons ensuite comment sélectionner les images représentatives des plans afin de conserver uniquement l'essentiel du contenu.

1.3.1 Segmentation en plans

La segmentation en plans peut paraître simple mais de nombreux artifices sont utilisés pour effectuer les transitions entre les plans. Ces artifices rendent cette tâche beaucoup moins évidente. Parmi les transitions fréquemment utilisées, nous trouvons la coupure, le fondu et la dissolution. Les méthodes de segmentation en plans sont nombreuses et sont effectuées soit dans le domaine compressé soit dans le domaine décompressé. La littérature aborde séparément ces deux catégories qui font intervenir des méthodes propres au support (compressé ou décompressé) (Koprinska and Carrato [24]).

Les méthodes de segmentation dans le domaine décompressé sont divisées en deux sous catégories : d'un côté les méthodes par comparaison des données, de l'autre côté les méthodes par application de modèles mathématiques. La première sous catégorie consiste essentiellement à comparer les caractéristiques d'images successives. Selon la différence calculée entre deux images, le système conclue sur la présence ou non d'un changement de plan. La littérature présente l'utilisation de nombreuses caractéristiques accompagnées de leur mesure de comparaison. Nous retrouvons principalement les caractéristiques suivantes : la couleur des pixels, les histogrammes locaux ou globaux (Zhang et al. [60]) et la structure (essentiellement contours et coins) (Zabih et al. [59]). La seconde catégorie emploie des modèles mathématiques pour modéliser la manière dont les transitions entre les plans se déroulent. Plusieurs modèles sont construits et le plus adapté aux données permet de conclure sur la nature de la transition ou son absence. Les modèles peuvent explicitement représenter la transition (Hampapur et al. [16]) ou être construit automatiquement en utilisant par exemple un modèle de Markov caché (Boreczky and Wilcox [5]).

Les méthodes de segmentation dans le domaine compressé font intervenir une ou plusieurs propriétés du flux MPEG. Elles vont de la simple utilisation de la composante continue de la transformée DCT, à l'utilisation combinée des coefficients DCT, du mouvement, du type des macro blocs et du débit (Taskiran et al. [47]).

La segmentation en plan est une première étape primordiale afin de pouvoir correctement indexer ou analyser le contenu d'une vidéo comme nous le verrons par la suite. Elle est également nécessaire pour réaliser la segmentation en scènes comme nous allons le voir.

1.3.2 Segmentation en scènes

La scène est une unité sémantique importante puisqu'elle contient une séquence de plans dont la logique permet d'exprimer une idée. Par contre, le plan est une unité technique qui est souvent de courte durée et son étude isolée ne permet pas de comprendre réellement le déroulement de la scène. La détermination des scènes est alors utile pour la navigation, la visualisation des données audiovisuelles et aussi pour leur analyse sémantique. Elles permettent aux utilisateurs d'avoir une bonne idée de l'action s'y déroulant ou de l'ambiance dégagée. Toutefois le problème de la segmentation en scènes est délicat et leur utilisation pour l'indexation et la recherche d'information en est pour l'instant à ses débuts. La première étape incontournable est le découpage en plans, ensuite l'étude de l'organisation des plans permet de grouper les plans en scènes. Deux catégories se distinguent parmi les approches existantes. La première catégorie comprend les approches utilisant les algorithmes de regroupement. Les plans sont regroupés en fonction de leur similarité et éventuellement de contraintes temporelles (Yeung and Yeo [55]). Un graphe des transitions est ensuite construit et

il permet de capturer la logique présente dans la séquence des plans. Dans la deuxième catégorie se trouvent les méthodes séquentielles qui regroupe au fur et à mesure les plans. Un ensemble de règle permet de définir si un plan appartient à la scène courante ou à une nouvelle scène (Hanjalic et al. [17], Tavanapong and Zhou [48]).

L'analyse des scènes offre l'opportunité d'avoir une indexation sémantique. Malheureusement le problème de la segmentation en scènes n'est actuellement pas correctement résolu dans le cas général et la plupart des méthodes d'indexation et de recherche de la vidéo reposent uniquement sur le découpage en plans. Par ailleurs le contenu d'une scène peut-être visuellement très varié et la notion de plan est toujours nécessaire pour identifier et caractériser les différents contenus. L'indexation des scènes repose donc sur les plans et plus particulièrement leurs images caractéristiques.

1.3.3 Sélection de l'image représentative

Il est inutile d'entrer dans des calculs compliqués et longs pour toutes les images d'un plan afin de correctement en extraire les caractéristiques visuelles. En effet, il serait par la suite impossible de conserver et d'utiliser cette information qui est par ailleurs redondante. Le processus de simplification de la vidéo continue donc par la sélection d'une ou plusieurs images représentatives des plans. Idéalement les images clefs doivent capturer le contenu sémantique du plan. Malheureusement les techniques de traitement de l'information ne sont pas assez avancées pour déterminer de telles images clefs. Les algorithmes utilisent donc les caractéristiques brutes obtenues sur les images (couleur, texture, mouvement). Lorsqu'un plan est statique, les images le composant sont souvent très similaires. Théoriquement il suffit alors de choisir l'image qui est la plus similaire aux autres. Malheureusement cette recherche exhaustive est difficilement réalisable en pratique. Les approches empiriques sélectionnent simplement la première, la dernière ou l'image médiane du plan. Yueting et al. [58] proposent une approche par regroupement des images similaires pour obtenir la ou les images représentatives du plan. Toutefois lorsqu'un plan est mouvementé, il est intéressant de sélectionner les images représentatives en fonction de l'intensité ou des variations du mouvement. Kobla et al. [23] présentent une approche dans le domaine compressé MPEG qui mesure le déplacement de la caméra dans le plan et découpe ce dernier en sous plans afin de limiter l'amplitude du mouvement. Wolf [53] utilisent le flux optique pour détecter l'image avec l'activité la plus faible. Liu et al. [31] proposent une mesure de l'énergie du mouvement dans le domaine MPEG afin d'identifier les images clefs du plan.

Une approche différente consiste à créer une mosaïque du plan (Irani and Anandan [21]). Cette approche est beaucoup plus rare car les mosaïques sont difficiles à construire dans le cas général. Une mosaïque est une unique image représentant l'ensemble de la scène. Elle est construite à partir des images du plan et fournit une représentation complète et compacte du fond. Les éléments mobiles sont décrits à part ainsi que leur trajectoire sur la mosaïque.

Au final de nombreuses méthodes d'indexation et de recherche sur le contenu visuel des vidéos sont très similaires aux méthodes employées sur les images puisqu'elles se concentrent uniquement sur les images clefs. Toutefois la vidéo permet l'utilisation de caractéristiques propres comme le son, les sous-titre et les mouvements de la caméra et des objets. La difficulté sera alors de combiner cet ensemble de caractéristiques hétérogènes de manière efficace. Dans un premier temps nous concentrons notre étude sur les caractéristiques calculées sur les images représentatives ou des séquences d'images.

1.4 Description du plan

De nombreuses méthodes ont été proposées dans la litérature pour capturer le contenu des plans et obtenir des signatures efficaces. La première partie traite de l'ensemble des méthodes d'indexation du contenu purement visuel. Ces méthodes sont généralement appliquées aux images représentatives des plans. Elles sont donc souvent similaires à celles que nous trouvons pour l'indexation d'images. La seconde partie traite de l'analyse du mouvement qui est naturellement étudié sur l'ensemble du plan.

1.4.1 Description visuelle d'une image

La littérature propose de nombreuses approches pour décrire le contenu visuel et aussi de nombreux états de l'art (Mandal et al. [34], Rui et al. [42]). Une première étape consiste à définir les caractéristiques requises en fonction du problème. Les différentes caractéristiques pouvant être extraites afin de décrire au mieux le contenu, peuvent être regroupées principalement dans trois catégories primaires : les caractéristiques de couleur, texture et structure, et une catégorie hybride. La description des couleurs est réalisée essentiellement par des histogrammes calculés dans différents espaces de couleur. La description de la texture est réalisée par une analyse fréquentielle ou l'étude d'occurrences. La description de la structure est réalisée par les coins, les angles, les contours et les formes. Les caractéristiques hybrides permettent de décrire conjointement les caractéristiques primaires. Elles se retrouvent principalement dans les domaines transformés, par exemple FFT, DCT et PCA. La deuxième étape consiste à définir les régions de l'image dans lesquelles les données vont être extraites. Selon l'application, il sera suffisant de décrire l'image comme une unique entité. D'autres applications plus complexes peuvent nécessiter une description locale du contenu faisant appel à une détection des régions ou des objets. Des compromis doivent être établis entre l'objectif de l'application, la complexité de la représentation et les temps de calcul requis pour le traitement des données.

A partir du flux MPEG

L'extraction des caractéristiques des plans à partir du flux MPEG a montré un grand intérêt de la part de nombreux chercheurs. Le but principal est de limiter les temps de calcul en utilisant l'information existante en l'état. Le flux vidéo permet l'accès direct aux coefficients DCT des macro blocs dans les images I. Ils fournissent à la fois une information locale de couleur et de texture. Le flux vidéo permet également l'accès à une information de mouvement qui peut s'avérer particulièrement riche (Kobla et al. [23], Taskiran et al. [47]). Une méthode a été proposée dans (Shen and Delp [43]) pour estimer les coefficients DC des images prédites P et B sans effectuer la décompression complète des images I et P. Il est alors possible de traiter toutes les images de la vidéo en se contentant des coefficients DC. Finalement de rares méthodes ont été proposées pour extraire les contours à partir des coefficients DCT (Lee et al. [25]).

Malheureusement le traitement effectué dans le domaine compressé demeure grossier. Cela est principalement dû à la notion même de macro bloc. De plus il est étroitement lié au mode de compression. L'alternative est alors d'effectuer le traitement sur la représentation universelle de l'image par des pixels.

A partir des pixels

Le passage au domaine décompressé est incontournable pour extraire plus d'information et gagner en précision. L'indexation donne alors accès aux deux caractéristiques primaires qui sont la couleur et la texture.

La couleur est une caractéristique très importante et elle est habituellement décrite par un histogramme calculé dans un des nombreux espace de couleur existant. L'histogramme de la requête est alors comparé aux histogrammes cibles de la base de données en utilisant une mesure de similarité. Les mesures les plus communes sont l'intersection d'histogramme, la distance pondérée entre les couleurs ou la distance euclidienne. Afin d'introduire des contraintes spatiales, plusieurs méthodes ont été proposées comme le découpage de l'image en zones d'intérêt (Stricker and Dimai [45]) ou l'étude de la corrélation spatiale des couleurs (Huang et al. [20]) (ou « color correlogram »).

La texture est également importante pour caractériser les motifs présents dans l'image, cependant elle n'a aucune définition précise. Quatre types d'approches se distinguent (Tuceryan and Jain [49]) : les approches statistiques, géométriques, spectrales et par modélisation. Dans la première catégorie nous trouvons les matrices de co-occurrence (Haralick [18]). Dans la seconde nous trouvons les descripteurs de Tamura (H. Tamura [15]) qui caractérisent la granularité, la direction et le contraste. Dans la troisième nous trouvons les ondelettes avec les filtres de Gabor (Turner [50], Manjunath and Ma [35]) qui permettent de capturer les fréquences et les directions principales. Finalement dans la dernière catégorie nous trouvons la décomposition de Wold (Liu and Picard [30]) qui caractérise la périodicité, la direction et le désordre ; ainsi que les modèles autorégressifs simultanés et multi résolution (Mao and Jain [36]) (ou « multiresolution simultaneous autoregressive models : MSAR ») qui modélisent la texture à différents niveaux de granularité en fonction du voisinage des pixels.

Malgré les contraintes spatiales qui peuvent être imposées par certaines méthodes, la notion de région et idéalement d'objet fait toujours défaut.

A partir des régions

Les systèmes basés sur les régions composant une image tentent de capturer le contenu d'une manière qui reflète le comportement humain. Pour cela, l'image est segmentée en régions homogènes puis les algorithmes travaillent au niveau de la granularité des régions. Ainsi les propriétés locales de l'image sont analysées.

Les systèmes qui proposent d'indexer les régions des images sont rares malgré leurs attraits. En effet deux barrières s'opposent à leur développement. La première et principale est la segmentation en objets. Ce délicat problème est actuellement loin d'être résolu dans le cas général. Les algorithmes de segmentation en régions sont nombreux mais ne parviennent pas à discerner les objets. Par ailleurs ils sont souvent très sensibles et il est difficile d'obtenir une segmentation homogène entre plusieurs images. La seconde barrière est le coût de l'analyse, du stockage et des calculs de similarité. La notion de région multiplie le nombre de paramètres et par conséquent la taille des données à stocker, ensuite elle multiplie le nombre de comparaisons possibles, ce qui rend beaucoup plus complexe les mesures de similarité. Deux types d'approches peuvent être identifiés selon la méthode de comparaison employée. Les méthodes de la première catégorie travaillent seulement sur les régions. Les opérations d'indexation et de recherche se font sur les régions qui sont dissociées de

l'image à laquelle elles appartiennent (Fauqueur and Boujemaa [13]). Le système Blobworld (Carson et al. [7]) utilise cette approche. Les méthodes de la seconde catégorie utilisent l'ensemble des régions composant une image pour effectuer l'indexation et la recherche. Deux mesures de similarité sont communément employées. La méthode IRM (acronyme de « Integrated Region Matching ») du système SIMPLIcity (Wang et al. [52]) permet de mettre en correspondance toutes les régions de deux images. La mesure EMD (Rubner et al. [41]) (acronyme de « Earth Mover's Distance ») permet de mesurer le coût nécessaire pour transformer un ensemble de régions en un deuxième ensemble de régions. Trois autres représentations des images segmentées ont également été proposées. La représentation de l'image et de ses régions peut être faite par des graphes (Matas et al. [37]) qui permettent une comparaison simultanée des propriétés et de l'agencement des régions. Elle peut également être faite par des chaînes qui décrivent les relations entre les régions (Lee and Hsu [26]). Finalement elle peut être réalisée par un vecteur de dénombrement des régions d'un dictionnaire (Lim [29]).

Le découpage des images en régions permet aussi de capturer de nouvelles caractéristiques concernant les structures des images comme nous allons le voir.

A partir des formes

La forme est un critère important pour caractériser un objet. Les caractéristiques globales sont habituellement décrites par la taille (périmètre et superficie), l'excentricité et les moments. Des caractéristiques plus précises ont aussi été proposées comme les coins, les points de courbures et la transformée de Fourier. Toutefois la recherche par la forme est un problème complexe principalement dû à la difficulté d'identifier les objets composant une scène puis aussi à la difficulté de représenter les formes de façon robuste.

1.4.2 Description du mouvement

Le mouvement est une information riche qui renseigne sur l'activité d'un plan et celle de ses objets. A partir de la séquence des images formant un plan, le mouvement de la caméra peut être estimé ; ensuite le mouvement des objets peut être déterminé.

L'ensemble des vecteurs de mouvement des points ou des régions est communément appelé flux optique. Il est à la base de tous les systèmes d'analyse du mouvement des scènes. De nombreuses techniques de calcul de flux optique ont été développées à partir des années 80 (Barron et al. [3], Quénot [40]) et de nouvelles techniques sont encore proposées de nos jours (Paulin et al. [38], Durik and Benois-Pineau [11], Foroosh [14]). Le flux optique est employé dans des domaines très variés comme l'imagerie médicale, la robotique, la télésurveillance, la compression et l'indexation, qui ont leurs propres contraintes. Sans être exhaustif, il permet d'effectuer la détection et le suivi d'objet, de modéliser l'environnement par reconstruction 3D, d'estimer le mouvement de la caméra, d'effectuer une segmentation spatio-temporel en région, ... Dans la suite nous portons notre attention sur les méthodes utilisées dans le cadre de l'indexation de documents vidéos.

Contrairement aux problèmes de vision comme la localisation ou la reconstruction, l'indexation ne requiert pas nécessairement une grande précision dans l'estimation du flux optique. Le choix de la méthode de calcul ne sera donc pas abordé puisque de nombreuses méthodes conviennent. Les deux méthodes les plus répandues sont (Horn and Schunck [19], Lucas and Kanade [33]),

cependant la plupart des méthodes utilisent directement les vecteurs mouvements fournis par les compressions MPEG-1 ou MPEG-2. L'objectif est alors d'éviter le lourd calcul du flux optique pour chaque image en utilisant le flux optique sur les macro blocs. A partir du flux optique, nous retrouvons essentiellement des méthodes d'estimation du mouvement de la caméra et des objets et de segmentation spatio-temporel en régions.

Une première approche à l'indexation du mouvement est l'utilisation directe du flux optique. Kobla et al. [23] proposent un vecteur caractéristique qui représente la direction quantifiée du mouvement pour chaque macro bloc du flux MPEG. Le mouvement peut également être indexé par un histogramme de l'ensemble des vecteurs disponibles (Yining and Manjunath [56]) ou des attributs caractéristiques (Ardizzone and Cascia [1]). Bruno and Pellerin [6] proposent une approche originale qui utilise les ondelettes. Les plans sont alors indexés par la moyenne des coefficients de la décomposition en ondelettes à plusieurs échelles.

Le mouvement de la caméra est principalement modélisé par un modèle affine et l'estimation des paramètres est souvent obtenue par une minimisation par les moindres carrées. De plus une méthode de rejection des points indésirables est ajoutée pour éliminer les points dont le mouvement est mal estimé ou qui appartiennent à un objet mobile (Yap-Peng et al. [54]). Les paramètres obtenus peuvent alors servir à l'indexation ou être classés dans les catégories zoom, rotation, translation verticale et horizontale pour une indexation sémantique.

La segmentation spatio-temporel et le suivi des objets sont un domaine très vaste et actuellement très actif. Toutefois l'indexation des objets et de leur mouvement demeure rare. VideoQ (Chang et al. [8]) fait partie de ces systèmes et il permet l'indexation de la trajectoire des objets. Dagtas et al. [9] présentent une approche d'indexation des trajectoires accompagnée d'une méthode de recherche des trajectoires qui a l'avantage d'être invariante dans l'espace et robuste aux changements d'échelle. DeMenthon and Doermann [10] proposent une description spatiale et temporelle. Une segmentation spatio-temporelle est d'abord réalisée. Les régions obtenues dans chaque image sont ensuite indexées avec l'objet auquel elles appartiennent. La recherche permet alors de retrouver les objets aux mêmes propriétés visuelles et de mouvement. Syeda-Mahmood [46] propose une modélisation en 3D de l'objet et de son évolution dans le temps. Cette modélisation permet ensuite de retrouver les objets identiques et au mouvement similaire sous différents points du vue.

Cependant le mouvement est rarement utilisé directement pour l'indexation mais plutôt pour la détection du contenu sémantique. Un des principal exemple est la détection d'évènements sportifs (Leonardi et al. [27]). Les descripteurs sémantiques caractérisant le mouvement global de la caméra ou l'activité des plans sont souvent suffisants pour les tâches d'indexation et de recherche.

1.5 Conclusion

Ce chapitre était dédié à l'introduction des systèmes de recherche d'images ou de vidéos par le contenu. Tout d'abord nous avons décrit le fonctionnement général de tels systèmes. Dans le descriptif fournit, une place était réservée aux systèmes qui permettent une réelle indexation sémantique par l'intermédiaire de mots clefs, et notamment aux futurs systèmes automatiques d'indexation sémantique. Une présentation du support vidéo a été effectuée. En particulier nous nous sommes attardés sur les problèmes de segmentation temporelle en plans et en scènes. Finalement nous avons donné un aperçu des caractéristiques visuelles qui sont habituellement employées dans les systèmes

d'indexation et de recherche par le contenu.

Cette présentation a mis en place avec précision le cadre des travaux présentés dans les prochains chapitres. Ainsi elle permettra de créer un lien entre les travaux existants et les travaux que nous avons effectués. Le prochain chapitre se place dans le contexte de la caractérisation du contenu visuel des plans en proposant une analyse de la sémantique latente.

Bibliographie

[1] E. Ardizzone and M. La Cascia. Video indexing using optical flow field. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 831–834, 1996.

[2] Jeffrey R. Bach, Charles Fuller, Amarnath Gupta, Arun Hampapur, Bradley Horowitz, Rich Humphrey, Ramesh Jain, and Chiao-Fe Shu. The virage image search engine. In *Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases*, 1996.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *Int. J. Comput. Vision*, 12(1) :43–77, 1994. ISSN 0920-5691.

[4] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r*-tree : an efficient and robust access method for points and rectangles. *ACM-SIGMOD Internation Conference on Management of Data*, 19(2) :322–331, 1990.

[5] John Boreczky and Lynn Wilcox. A hidden markov model framework for video segmentation using audio and image features. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3741–3744, may 1998.

[6] E. Bruno and D. Pellerin. Video structuring, indexing and retrieval based on global motion wavelet coefficients. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 287–290, 2002.

[7] Chad Carson, Megan Thomas, and Serge Belongie. Blobworld : A system for region-based image indexing and retrieval. In *Third international conference on visual information systems*, 1999.

[8] Shih-Fu Chang, W. Chen, H.J. Meng, H. Sundaram, and Di Zhong. A fully automated content-based video search engine supporting spatiotemporal queries. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 8, pages 602– 615, 1998.

[9] S. Dagtas, W. Al-Khatib, A. Ghafoor, and R.L. Kashyap. Models for motion-based video indexing and retrieval. *IEEE Transactions on Image Processing*, 9(1) :88–101, January 2000.

[10] Daniel DeMenthon and David Doermann. Video retrieval using spatio-temporal descriptors. In *Proceedings of the ACM International Conference on Multimedia*, pages 508–517, 2003.

[11] Mojmir Durik and Jenny Benois-Pineau. Robust motion characterisation for video indexing based on MPEG2 optical flow. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, pages 57–64, 2001.

[12] Christos Faloutsos, Ron Barber, Myron Flickner, Jim Hafner, Wayne Niblack, Dragutin Petkovic, and William Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4) :231–262, 1994.

[13] J. Fauqueur and N. Boujemaa. Region-based image retrieval : Fast coarse segmentation and fine color description. *Journal of Visual Languages and Computing*, 15(1) :69–95, february 2004.

[14] H. Foroosh. Pixelwise-adaptive blind optical flow assuming nonstationary statistics. *IEEE Transactions on Image Processing*, 14(2) :222–230, february 2005.

[15] T. Yamawaki H. Tamura, S. Mori. Textural features corresponding to visual perception. *IEEE Transaction on Systems, Man and Cybernetics*, 8 :460–482, 1978.

[16] Arun Hampapur, Ramesh Jain, and Terry E. Weymouth. Production model based digital video segmentation. *Multimedia tools and applications*, 1(1) :9–46, march 1995.

[17] Alan Hanjalic, Reginald L. Lagendijk, and Jan Biemond. Automated high-level movie segmentation for advanced video-retrieval systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4) :580–588, 1999.

[18] Robert M. Haralick. Statistical and structural approaches to texture. *Proceedings of the I.E.E.E.*, 67(5) :786–804, May 1979.

[19] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17 :185–203, 1981.

[20] Jing Huang, S. Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–768, 1997.

[21] M. Irani and P. Anandan. Video indexing based on mosaic representation. *IEEE Transaction on PAMI*, 86(5) :905–921, 1998.

[22] Fan Jianping, A.K. Elmagarmid, Zhu Xingquan, W.G. Aref, and Wu Lide. Classview : hierarchical video shot classification, indexing, and accessing. *IEEE Transactions on Multimedia*, 6 (1) :70–86, february 2004.

[23] Vikrant Kobla, David S. Doermann, King-Ip Lin, and Chritos Faloutsos. Compressed domain video indexing techniques using DCT and motion vector information in MPEG video. In *Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases*, volume 3022, pages 200–211, february 1997.

[24] Irena Koprinska and Sergio Carrato. Temporal video segmentation : A survey. *Signal Processing : Image Communication*, 16 :451–460, 2001.

[25] Michael Lee, Surya Nepal, and Uma Srinivasan. Role of edge detection in video semantics. In *Pan-Sydney Workshop on Visual Information Processing (VIP2002)*, volume 22 of *Conferences in Research and Practice in Information Technology*, page 59, 2003.

[26] Suh-Yin Lee and Fang-Jung Hsu. Spatial reasoning and similarity retrieval of images using 2d c-string knowledge representation. *Pattern Recogn.*, 25(3) :305–318, 1992.

[27] R. Leonardi, P. Migliorati, and M. Prandini. Semantic indexing of soccer audio-visual sequences : a multimodal approach based on controlled markov chains. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5) :634–643, may 2004.

[28] Jia Li and James Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9) :1075–1088, 2003.

[29] Joo-Hwee Lim. Learning visual keywords for content-based retrieval. In *IEEE International Conference on Multimedia Computing and Systems*, volume 2, pages 169–173, 1999.

[30] F. Liu and W. Picard. Periodicity, directionality, and randomness : Wold features for image modeling and retrieval. *IEEE Transaction Pattern Analysis and Machine Intelligence*, 18 : 722–733, 1996.

[31] Tianming Liu, Hong-Jiang Zhang, and Feihu Qi. A novel video key-frame extraction algorithm based on perceived motion energy model. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(10) :1006–1013, october 2003.

[32] Ye Lu, Hongjiang Zhang, Liu Wenyin, and Chunhui Hu. Joint semantics and feature based image retrieval using relevance feedback. *IEEE Transactions on Multimedia*, 5(3) :339–347, september 2003.

[33] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[34] M.K. Mandal, F. Idris, and S. Panchanathan. A critical evaluation of image and video indexing techniques in the compressed domain. *Image and Vision Computing Journal*, 17(7) :513–529, may 1999.

[35] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8) :837–842, 1996.

[36] Jianchang Mao and Anil K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2) :173–188, 1992.

[37] J. Matas, R. Marik, and J. Kittler. On representation and matching of multi-coloured objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 726–732, 1995.

[38] Damien Paulin, Dinesh Kumar, Raghav Bhaskar, and Georges Quénot. Recovering camera motion and mobile objects in video documents. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, pages 39–46, 2001.

[39] A. Pentland, R. Picard, and S. Sclaroff. Photobook : Content-based manipulation of image databases. In *Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases*, february 1994.

[40] Georges Quénot. Computation of optical flow using dynamic programming. In *Proceedings of IAPR Workshop on Machine Vision Applications*, 1996.

[41] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 59–66, january 1998.

[42] Y. Rui, T. Huang, and S. Chang. Image retrieval : current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(4) :39–62, April 1999.

[43] K. Shen and Edward J. Delp. A fast algorithm for video parsing using MPEG compressed sequences. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 252–255, october 1998.

[44] John R. Smith and Shih-Fu Chang. Visualseek : A fully automated content-based image query system. In *Proceedings of the ACM International Conference on Multimedia*, pages 87–98, 1996.

[45] Markus A. Stricker and Alexander Dimai. Color indexing with weak spatial constraints. In *Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases*, volume 2670, pages 29–40, 1996.

[46] Tanveer Syeda-Mahmood. Retrieving actions embedded in video. In *Proceedings of the ACM International Conference on Multimedia*, pages 513–522, 2002.

[47] Cuneyt Taskiran, Jau-Yuen Chen, Alberto Albiol, Luis Torres, Charles A. Bouman, and Edward J. Delp. Vibe : a compressed video database structured for active browsing and search. *IEEE Transactions on Multimedia*, 6(1) :103–118, february 2004.

[48] Wallapak Tavanapong and Junyu Zhou. Shot clustering techniques for story browsing. *IEEE Transactions on Multimedia*, 6(4) :517–527, august 2004.

[49] Mihran Tuceryan and Anil K. Jain. *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, chapter Texture Analysis, pages 207–248. World Scientific Publishing Co., 1998.

[50] M. R. Turner. Texture discrimination by gabor functions. *Biol. Cybern.*, 55(2-3) :71–82, 1986.

[51] Howard Wactlar, Takeo Kanade, Michael A. Smith, and Scott M. Stevens. Intelligent access to digital video : The informedia project. *IEEE Computer*, 29(5), 1996.

[52] James Z. Wang, Jia Li, and Gio Wiederhold. SIMPLIcity : Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9) :947–963, 2001.

[53] W. Wolf. Key frame selection by motion analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1228–1231, 1996.

[54] Tan Yap-Peng, D.D Saur, S.R. Kulkami, and P.J. Ramadge. Rapid estimation of camera motion from compressed video with application to video annotation. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(1) :133–146, february 2000.

[55] Minerva M. Yeung and Boon-Lock Yeo. Time-constrained clustering for segmentation of video into story unites. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 375–380, 1996.

[56] Deng Yining and B.S. Manjunath. Content-based search of video using color, texture, and motion. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 534–537, 1997.

[57] Deng Yining and B.S. Manjunath. Netra-v : toward an object-based video representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5) :616–627, september 1998.

[58] Zhuang Yueting, Rui Yong, T.S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Proceedings of the IEEE International Conference on Image Processing*, volume 1, pages 866–870, 1998.

[59] Ramin Zabih, Justin Miller, and Kevin Mai. A feature-based algorithm for detecting and classifying production effects. *Multimedia Systems*, 7(2) :119–128, march 1999.

[60] Hong Jiang Zhang, Atreyi Kankanhalli, and Stephen W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1) :10–28, june 1993.

Chapitre 2

Utilisation des régions pour l'analyse du contenu

Les premiers systèmes d'indexation d'images par le contenu, comme VisualSeek ou QBIC, ont démontré un certain succès dans la gestion de requêtes générales en utilisant des caractéristiques globales de l'image. Toutefois, ces systèmes ont leurs limites. Tout d'abord l'utilisation de caractéristiques globales élimine les contraintes d'organisation spatiale de l'information. Ensuite elle ne reflète pas la manière dont nous percevons le contenu. Finalement elle ne permet pas de représenter efficacement le contenu sémantique. Une première réponse, que nous allons développer dans ce chapitre, remplace la description globale du contenu par une description locale. Les images sont alors décomposées en régions qui vont permettre de décrire le contenu. Cette approche s'apparente plus au comportement que nous adoptons pour observer notre environnement. De plus elle répond mieux au besoin réel des utilisateurs qui recherchent des objets présents dans des scènes ou des scènes dont la composition change.

Comme nous avons pu le voir dans le chapitre précédent, les méthodes d'indexation des régions sont peu répandues. La difficulté de la segmentation, la complexité de la représentation et des mesures de comparaison sont les principales barrières à leur développement. Cependant l'intérêt de travailler sur des régions est immense. Outre le fait d'apporter une description qui est en accord avec notre système visuel, l'analyse des régions ouvre les portes d'une étude plus approfondie du contenu comme la détection des objets, la séparation de l'avant plan et de l'arrière plan ou l'analyse des interactions entre les régions. Nous avons donc porté notre attention sur la représentation efficace et compacte du contenu des plans en utilisant les régions. Et les travaux sont conduits dans le cadre de l'indexation de plans vidéo par le contenu visuel.

Ce chapitre débute par une présentation du modèle vectoriel appliqué aux images ou IVSM pour représenter le contenu visuel des plans vidéo. Une présentation des caractéristiques visuelles utilisées pour décrire le contenu des régions est également donnée. Ensuite nous présentons une nouvelle approche, dite analyse du contenu latent ou LSA, qui permet d'améliorer les performances du modèle vectoriel appliqué aux images. Puis nous comparons les performances de ce nouveau modèle avec une méthode de base utilisant la mesure « Earth Mover's Distance » qui est présentée. Finalement des extensions sont proposées pour rendre le modèle plus robuste à la segmentation et au mouvement. Dans ce même cadre, nous étudions les effets de la boucle d'asservissement qui

permet d'améliorer les résultats de la recherche en inter agissant avec l'utilisateur.

2.1 Description du plan par des régions

L'image est segmentée en régions homogènes afin d'analyser les propriétés locales de l'image. Le problème de l'excédent d'information se pose alors. En effet, une image contient en moyenne soixante régions et chaque région est décrite par un vecteur de taille moyenne 100. La dimension de la signature devient alors rapidement énorme et les opérations de comparaison complexes et interminables. Nous proposons de travailler sur une approche basée sur les vecteurs de dénombrement qui sera appelé modèle vectoriel appliqué aux images (IVSM pour « Image Vector Space Model »). C'est une représentation répandue dans le domaine de l'indexation de documents textuels sous le nom de modèle vectoriel (ou « vector space model »). Il s'agit de représenter un document par le dénombrement des mots d'un dictionnaire défini au préalable. Dans le cas présent, les régions sont assimilées à des mots décrivant le contenu de l'image. Par contre le dictionnaire n'existe pas naturellement. Cependant il peut facilement être créé à l'aide des techniques aveugles d'agglomération pour obtenir des « visual keywords » (Lim [8]).

2.1.1 Segmentation spatiale

La segmentation d'images est un problème particulièrement difficile qui n'a actuellement pas de solution générique et de nombreux chercheurs se concentrent sur des tâches particulières comme l'extraction de route, la recherche des objets mobiles, ... La recherche est toujours très active dans ce vaste domaine et il existe de nombreuses méthodes de segmentation aux applications différentes (Rochery et al. [13], Comanicu and Meer [2], Shi and Malik [15], Cheng et al. [1], Lucchese and Mitra [9]). L'étude et l'analyse de nouveaux algorithmes de segmentation spatiale n'entre pas dans le cadre de ces travaux. Nous allons donc simplement présenter la méthode pour laquelle nous avons opté afin de réaliser cette tâche difficile. L'algorithme présenté par Felzenszwalb and Huttenlocher [4] a été sélectionné pour sa capacité à préserver les détails dans les images avec peu de variations et de les ignorer dans les images avec de grandes variations. De plus l'algorithme est assez rapide pour pouvoir traiter de nombreuses images et notamment le contenu des vidéos ce qui en fait un candidat adapté à nos besoins.

L'algorithme repose sur un graphe de l'image afin de détecter des preuves de l'existence de limites entre chaque paire de régions voisines. Initialement les noeuds du graphe sont identifiés aux pixels de l'images et les arcs sont construits entre un pixel et ses quatre voisins (en haut, en bas, à gauche et à droite). Les noeuds du graphe sont ensuite itérativement regroupés en fonction d'un seuil dynamique qui dépend du contenu des régions. L'algorithme est alors linéaire par rapport au nombre d'arc initiaux. Le seul paramètre qui intervient dans l'algorithme est la valeur d'un coefficient K qui permet de moduler l'évolution du seuil dynamique. La figure 2.1 montre l'effet de ce paramètre sur la segmentation.

2.1.2 Description des régions

La segmentation spatiale étant réalisée, la prochaine étape est la description des régions. Nous avons retenu deux types de caractéristiques qui sont la couleur et la texture. La forme n'a pas été

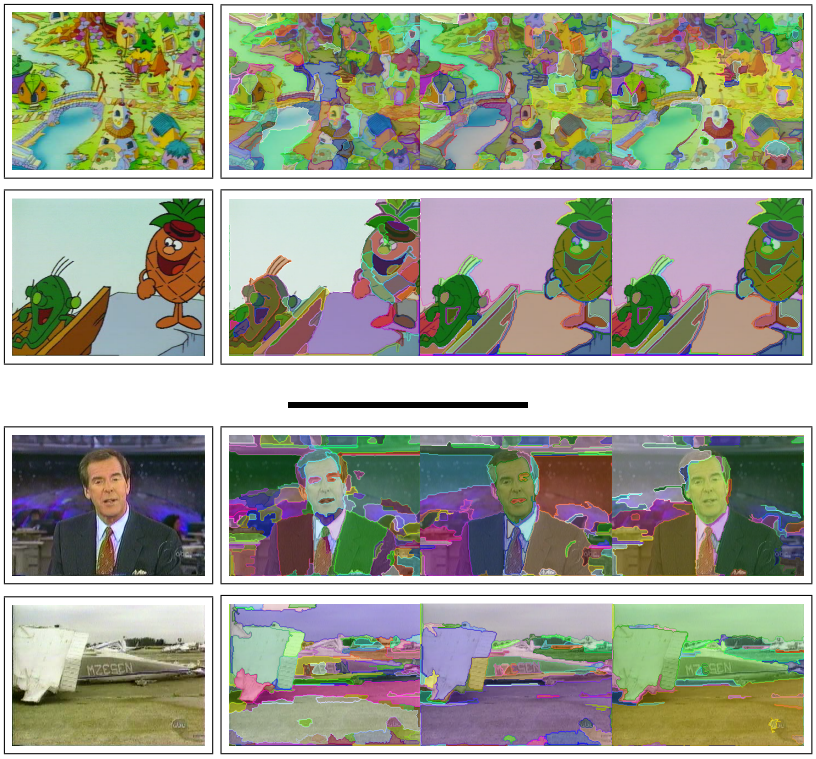


FIG. 2.1 – Rendu de la segmentation automatique obtenue avec trois valeurs du paramètre K (100, 200 et 300) qui module la finesse de la segmentation.

prise en compte puisque nous avons constaté que cette caractéristique n'était pas fiable avec l'algorithme de segmentation utilisé. La couleur est décrite par un histogramme des couleurs HSV et la texture par l'énergie des réponses aux filtres de Gabor. Ces deux modèles ont été choisis pour leur aptitude à correctement décrire le contenu (Ma and Zhang [10]).

Couleur

Les couleurs d'une région sont modélisées par leur histogramme, c'est à dire une estimation de la densité de probabilité des couleurs. L'histogramme est construit dans l'espace de couleur HSV qui est perceptiblement plus approprié que l'espace de couleur RGB. Nous rappelons qu'un histogramme est construit en deux phases. La première est la quantification des couleurs. La deuxième est le dénombrement des couleurs quantifiées, qui constituera l'histogramme. Chaque composante du vecteur de dénombrement donne la quantité d'une couleur présente dans l'image. Les histogrammes sont souvent normalisés par le nombre de pixels pour les rendre invariant à la taille de l'image ou des régions. Nous avons opté pour une quantification uniforme avec quatres vecteurs de quantification par canal de couleur. L'histogramme possède alors 64 composantes.

Texture

La texture composant une région est modélisée par les énergies des réponses à 24 filtres de Gabor. Les filtres de Gabor ont la particularité d'effectuer un filtrage proche de celui réalisé par notre système de perception visuelle. Ils ont une grande sensibilité à l'orientation et à la fréquence. De plus ils ont l'avantage d'avoir une résolution optimale conjointe en fréquence et dans l'espace. Un ensemble de filtres permet alors de capturer les directions et les fréquences principales de l'image. Un filtre de Gabor est un filtre de fréquence pure modulé par une gaussienne. Dans le domaine fréquentiel, en coordonnées polaires et pour une direction i et une échelle j, il se met sous la forme :

$$G_{i,j}(\rho,\theta) = \exp\left(\frac{(\rho-\mu_{\rho,i,j})^2}{\sigma^2_{\rho,i,j}} + \frac{(\theta-\mu_{\theta,i,j})^2}{\sigma^2_{\theta,i,j}}\right) \tag{2.1}$$

Les paramètres sont ensuite calculés tels que les fréquences centrales $(\mu_{\rho,i,j}, \mu_{\theta,i,j})$ décroissent en octave et que les variances $(\sigma_{\rho,i,j}, \sigma_{\theta,i,j})$ permettent le chevauchement des elipses de Gabor à la moitiée de leur amplitude maximale. Cette approche permet d'optimiser la couverture de l'espace tout en limitant la redondance comme le montre la figure 2.2. Au final trois paramètres doivent être spécifiés par l'utilisateur, la fréquence maximale, le nombre d'orientations et le nombre d'échelles. Nous avons choisi les paramètres habituels, c'est à dire une fréquence maximale de 0.5, six orientations et quatre échelles. Une illustration dans le domaine fréquentiel et spatial des filtres utilisés est proposée dans l'annexe A à la page 115 .

2.1.3 Dictionnaire visuel

La première étape pour obtenir une représentation à l'aide d'un vecteur de dénombrement est de définir les entités à dénombrer. Le principe est illustré sur le schéma 2.3(a). Dans le cadre d'une description des plans reposant sur les régions, ces entités sont des régions de référence. Elles

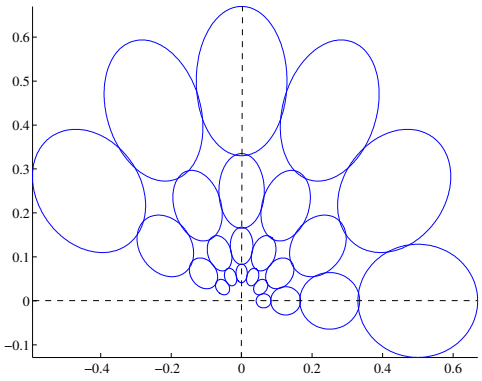


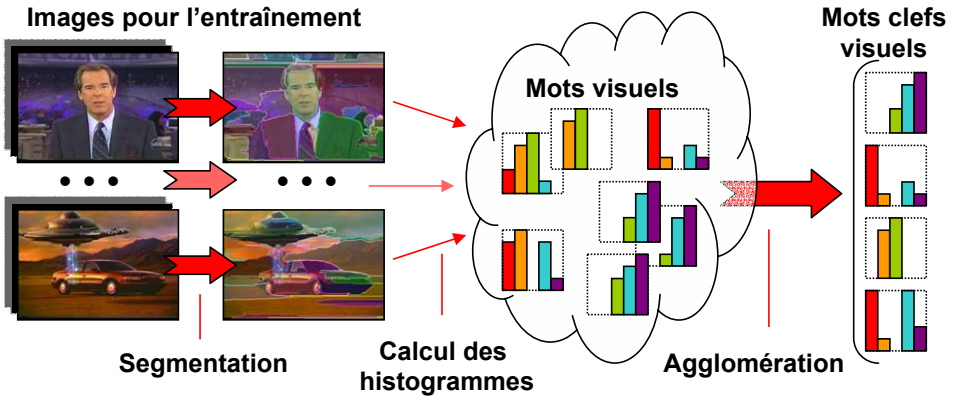
FIG. 2.2 – Forme des enveloppes des filtres de Gabor dans le domaine fréquentiel. 4 échelles et 6 orientations.

peuvent être obtenues de différentes manières et notamment par les techniques aveugles d'agglomération. Nous avons utilisé dans nos travaux l'algorithme connu sous le nom de « k-means » (ou k-moyennes). Il a été selectionné pour la simplicité de sa mise en oeuvre et sa rapidité d'exécution. A partir des signatures brutes fournies sur un échantillon de vidéos, l'algorithme permet de déterminer un nombre prédéfini de centres qui représentent au mieux l'ensemble des signatures. Ces centres composent alors le dictionnaire visuel. Plusieurs dictionnaires peuvent être construits. En particulier il est intéressant de définir un dictionnaire par type de caractéristique, par exemple un premier pour la couleur et un second pour la texture. Notons que le problème d'agglomération devient plus délicat dès lors que la dimension des vecteurs traités croît et il est préférable d'analyser les différentes sources d'information indépedamment.

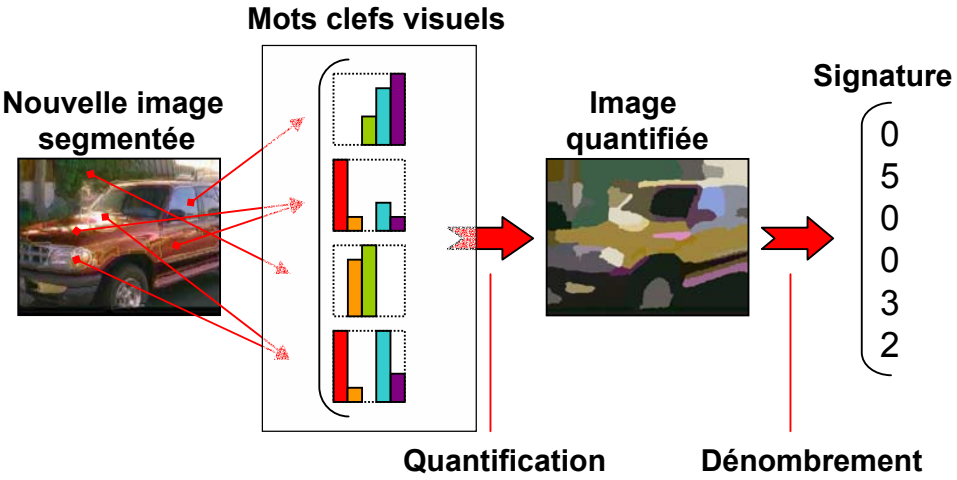
Une fois le dictionnaire défini, la signature est construite de la manière suivante. Tout d'abord l'image est découpée en régions. Chaque région est ensuite associée à un ou plusieurs mots clefs visuels dans un ou plusieurs dictionnaires. Cette association est obtenue par une recherche du ou des plus proches mots clefs visuels en utilisant une mesure appropriée ; comme la distance euclidienne dans notre cas. La signature est simplement le dénombrement des mots clefs visuels. Deux signatures sont comparées en étudiant les points communs, c'est à dire les mots clefs visuels communs aux signatures. Pour cela le produit scalaire ou le cosinus peuvent être calculés pour avoir une mesure numérique de la similarité des deux signatures. Le schéma 2.3(b) illustre le principe de l'indexation qui vient d'être décrit.

2.2 Description du contenu latent

La description du contenu visuel à l'aide de mots clefs visuels est une approche fort séduisante. Elle permet de conserver l'aspect local de l'information visuelle tout en conservant une mesure de similarité simple. Malheureusement elle souffre de deux inconvénients majeurs issus des différentes étapes impliquées dans le calcul des signatures. Le premier inconvénient est le décalage de l'information résultant de la segmentation. En effet la segmentation est rarement robuste aux changements de luminosité, de contraste ou au mouvement. Les éléments d'une scène ne sont donc pas systéma-



(a) Création du dictionnaire



(b) Obtention de la signature

FIG. 2.3 – Représentation du contenu reposant sur les régions.

tiquement segmentés de la même manière. Le deuxième inconvénient est l'imprécision résultant de la quantification : deux régions peuvent être visuellement proches et pourtant être décrite par deux mots clefs visuels différents.

Afin de réduire l'impact de ses problèmes, nous proposons d'appliquer la méthode d'analyse de la sémantique latente, LSA (acronyme de « Latent Semantic Analysis »). Cette méthode fut initialement introduite par Deerwester et al. [3] pour indexer les documents textuels. Elle consiste à établir des relations de synonymie et polysémie entre les mots. En d'autres termes, la présence d'un mot dans un texte induit automatiquement la présence des mots sémantiquement similaires. Une méthode linéaire permet de mettre en valeur automatiquement les relations existant entre les mots. Il s'agit de la décomposition en valeurs singulières de la matrice d'occurrence des mots dans leur contexte. Nous présentons tout d'abord les rares travaux antérieurs qui ont utilisé la LSA pour l'analyse d'image, ensuite la méthode d'analyse puis la méthode d'indexation.

Contexte

De rares extensions au son ou à l'image ont été présentées (Kurimo [6], Zhao and Grosky [18], Li et al. [7]). La plupart des travaux abordent le problème comme une réduction de la dimension d'un vecteur décrivant l'ensemble de l'image. Zhao and Grosky [18] décrivent le contenu global des images par des « color anglogram » pour déterminer les plans d'une vidéo. Li et al. [7] utilise la LSA pour décrire conjointement le contenu audio et visuel dans le but de détecter les personnes en train de parler. Une idée similaire est proposée par Monay and Gatica-Perez [11] qui utilisent des mots clef à la place du son pour réaliser l'annotation automatique d'image. Contrairement aux approches citées précédemment, les caractéristiques visuelles sont extraites sur trois régions pour être ensuite concaténées. Ce mémoire présente une étude différente qui s'apparente plus à l'analyse de documents écrits. Une représentation du contenu par des mots clefs visuels est alors adoptée comme nous l'avons vu dans la section précédente. L'idée est que chaque région participe au contenu sémantique du plan. La LSA est alors appliquée sur l'occurrence de ces mots clefs visuels dans les plans vidéos. Notons que la méthode proposée inclut les méthodes précédentes qui sont alors un cas particulier dans lequel les régions sont réduites à un pixel.

En 1999 un cadre probabiliste de l'analyse de la sémantique latente, PLSA, est proposé par Hofmann [5] dans le cadre de l'indexation de documents écrits. Monay and Gatica-Perez [11] ont ensuite comparé les deux méthodes LSA et PLSA pour l'annotation automatique d'images et ils ont observé que la LSA permettait l'obtention de meilleurs résultats. Dans la suite, nous concentrons nos travaux sur l'analyse de la sémantique latente par une méthode d'algèbre linéaire.

Analyse

La procédure suivie pour l'analyse des plans vidéo est la suivante. La signature d'un plan représente les occurrences des mots clefs visuels dans ce dernier. Assimilons le plan à un contexte. La matrice formée par la juxtaposition des signatures correspond alors à la matrice d'occurrence des mots clefs visuels dans leur contexte. Soit C cette matrice de taille MxN tel que $M < N$. La décomposition en valeurs singulières (SVD) permet de représenter C sous la forme :

$$C = UDV^t \tag{2.2}$$

$$\text{avec } UU^t = U^tU = I_M,\ V^tV = I_M$$

$$\text{et } D = diag(\sigma_1, \sigma_2, ..., \sigma_M) \text{ tel que } \sigma_1 \geq \sigma_2 \geq ...$$

Une illustration est proposée dans le schéma 2.4. La décomposition n'apporte en soi aucun changement. Par contre des effets très intéressants sont observés lorsque la matrice diagonale D est approximée par ses p plus grandes valeurs singulières $\{\sigma_i\}$. La matrice obtenue $\hat{C}_k$ contient de nouveaux coefficients indiquant de nouvelles relations entre les mots clefs visuels et les plans.

$$\hat{C}_k = U_k D_k V_k^t \tag{2.3}$$

$$\text{avec } U_k^t U_k = I_k,\ V_k^t V_k = I_k$$

$$\text{et } D_k = diag(\sigma_1, \sigma_2, ..., \sigma_k)$$

Le choix du nombre de valeurs propres à sélectionner est actuellement un problème non résolu et seuls des essais permettent de choisir le meilleur nombre. L'impact de ce choix est le suivant.

Plus le nombre de valeurs singulières supprimées est élevé, plus d'équivalences seront créées y compris celles qui n'auraient pas lieu d'être. Toutes les signatures finissent par être identiques. Dans le cas contraire lorsque toutes les valeurs singulières sont conservées, aucune relation n'est établie et les signatures ont facilement tendance à être différentes à cause des problèmes précédemment cités (dus à la segmentation suivie de la quantification). Dans le reste de ce mémoire nous désignerons par *dureté* la quantité absolue ou relative de valeurs propres qui sont supprimées. Plus ce nombre sera élevé plus la LSA sera qualifiée de dure.

Indexation

Il est inutile de travailler sur les nouvelles signatures fournies par $\hat{C}$ puisqu'il est possible de tirer partie de l'annulation des valeurs singulières afin de réduire la dimension des signatures. En effet, la comparaison des signatures peut se réaliser dans l'espace latent défini par la transformation orthonormale U comme nous allons le voir. La comparaison de tous les plans peut s'écrire de la manière suivante en utilisant le produit scalaire :

$$C^tC = (UDV^t)^t(UDV^t) = VD^tDV^t = (DV^t)^t(DV^t) \tag{2.4}$$

DV^t est alors l'ensemble des signatures dont la taille est M (ou k lorsque la projection est effective). Ensuite la comparaison d'un plan s avec les plans initiaux s'écrit :

$$s^tC = s^t(UDV^t) = (s^tU)(DV^t) \tag{2.5}$$

L'indexation d'un nouveau document est alors réalisée par la projection dans l'espace latent défini par U, espace où les comparaisons entre les plans peuvent être réalisées. La taille des signatures peut être réduite dans cet espace en fonction du nombre de valeurs singulières conservées. La comparaison de deux plans se fait alors en utilisant le produit scalaire ou le produit scalaire normalisé (c'est à dire le cosinus). Par la suite, la projection dans l'espace latent sera notée $p = (s^tU)$.

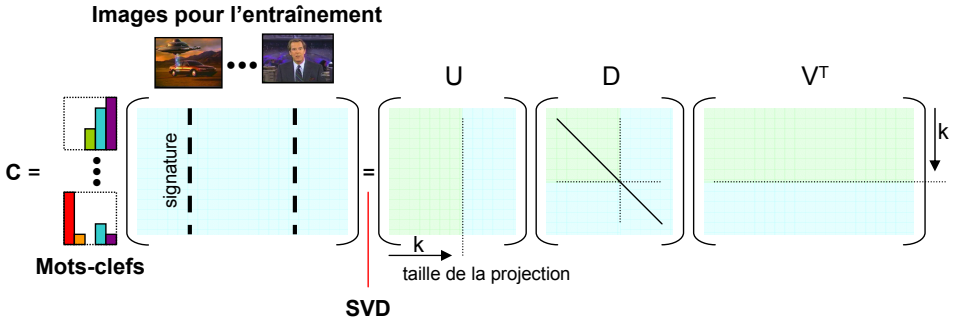


FIG. 2.4 – Principe de l'analyse du contenu latent.

2.3 Méthodes d'évaluation

Cette section présente la méthode d'évaluation que nous avons choisie. La première partie présente les mesures utilisées. La seconde partie décrit les vidéos sur lesquelles nous avons travaillé. La troisième partie explique comment la recherche de plans vidéo est réalisée.

2.3.1 Mesures d'évaluation

Les systèmes de recherche d'information sont essentiellement évalués à l'aide des valeurs de précision et de rappel lorsque n documents sont retournés. Si nous considérons un document correct lorsqu'il répond à la requête ; alors la précision p mesure la proportion de documents trouvés et corrects n_c parmi tous les documents trouvés n ; le rappel r mesure la proportion de documents trouvés et corrects n_c parmi tous les documents indexés et corrects N_c. Soit :

$$p = \frac{n_c}{n} \text{ et } r = \frac{n_c}{N_c} \tag{2.6}$$

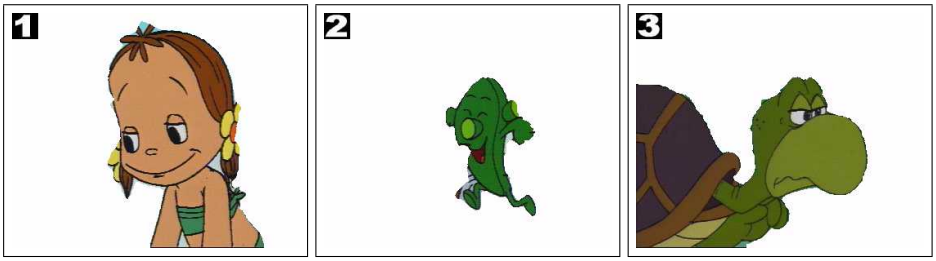
La courbe de la précision en fonction du rappel permet de suivre la qualité du résultat en fonction du nombre de documents retournés. Toutefois elle est très sensible à la requête et le comportement d'un système peut être très différent d'une requête à une autre. Pour pallier à cette sensibilité, l'évaluation repose plutôt sur la précision moyenne obtenue sur plusieurs requêtes pour un rappel donné. La courbe de la précision moyenne en fonction du rappel fournit alors une information plus fiable. Toutefois une valeur unique est parfois préférable à une courbe, en effet deux courbes sont souvent difficiles à comparer. Dans ce cas une précision moyenne est calculée par requête, puis une moyenne de cette valeur est calculée sur plusieurs requêtes. Cette valeur unique permet une comparaison rapide et univoque des performances des différents systèmes. Les courbes de précision en fonction du rappel sont alors retenues lorsqu'une étude précise est requise.

Il est nécessaire de connaître le résultat attendu des requêtes pour calculer ses valeurs et évaluer un système de recherche d'information. Malheureusement cette connaissance est souvent fastidieuse à obtenir. Tout d'abord elle n'est pas unique : il peut être difficile de juger si deux plans sont vraiment similaires. Par ailleurs l'être humain recherchera une similarité sémantique plutôt qu'une similarité visuelle. Ensuite elle demande un effort considérable et croissant avec la taille de la base de données. Tous les éléments doivent être annotés pour obtenir une évalution correcte. De ce fait, il existe actuellement très peu de base de données de vidéos fournies avec des annotations. Et les annotations sont sémantiques puisqu'elles correspondent aux réels besoins des utilisateurs.

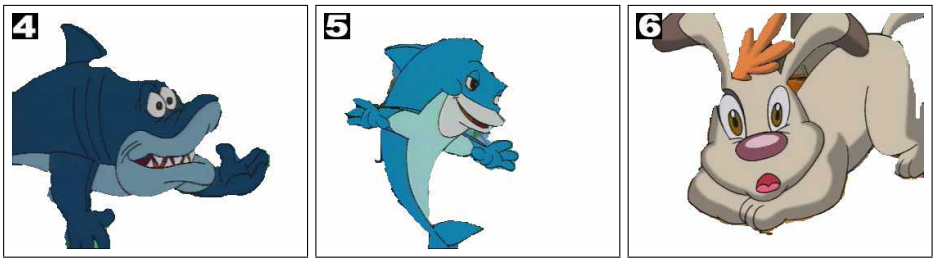
2.3.2 Les données utilisées

Nous travaillons en premier lieu sur une série de dessins animés provenant de la librairie MPEG-7. Sept personnages qui feront l'objet des recherches, ont été manuellement sélectionnés et annotés (figure 2.5 à la page 32). La requête est alors composée uniquement du personnage recherché, ensuite la tâche du système est de retourner toutes les images où ce personnage apparaît. Le travail au niveau du personnage ou des objets permet de démontrer la polyvalence de la méthode introduite qui peut s'appliquer aussi bien sur une image complète ou sur un ensemble de régions. Malheureusement le travail manuel d'identification des objets est particulièrement long et éprouvant. Et il a

été réalisé uniquement sur une vidéo et sept personnages. Au total 350 requêtes sont possibles et le nombre de requêtes par objets est précisé sur la figure 2.5.



(a) Fille, 21 occurrences (b) Cactus, 12 occurrences (c) Tortue, 114 occurrences



(d) Requin, 87 occurrences (e) Dauphin, 19 occurrences (f) Chien, 62 occurrences



(g) Entraîneur, 37 occurrences

FIG. 2.5 – Objets manuellement sélectionnés pour l'évaluation. Source : dessins animés fournis par la production Docon à la bibliothèque MPEG-7

En second lieu, les expériences sont conduites sur l'ensemble de vidéos fourni par TRECVID en 2003 ; ce qui permet de réaliser une évaluation à grande échelle. L'ensemble des vidéos de TRECVID est présenté succinctement. Une présentation plus approfondie sera donnée dans le prochain chapitre. Les vidéos de la base de données sont issues des nouvelles quotidiennes de deux chaînes américaines CNN et ABC. La base de données est composée de 60 heures de vidéos au total. Les plans de ces vidéos sont annotés ce qui permet d'évaluer les systèmes développés. Dans ce chapitre sept concepts ou classes sont retenus pour l'évaluation : (1) basket-ball, (2) climat, (3) studio, (4) visage, (5) Bill Clinton, (6) avion, (7) train. La base de données est partagée en deux parties égales. La première moitié de l'ensemble constitue la base de données dans laquelle les recherches sont

évaluées. La deuxième moitié constitue l'ensemble des plans requêtes possibles. Des exemples des images clefs de la base de données se trouvent dans l'annexe B à la page 117.

2.3.3 Recherche

Un système de recherche utilise une mesure de similarité pour déterminer les documents les plus pertinents. Dans ce mémoire le problème de l'organisation des signatures ne sera pas abordé, l'évaluation se place alors dans un cadre idéal où la requête est comparée à toutes les signatures de la base de données. La recherche est donc linéaire en fonction du nombre de plans composant la base de données. Les plans les plus similaires sont ensuite retournés dans l'ordre d'importance à l'utilisateur. Comme nous l'avons vu dans la partie précédente, le cosinus est une fonction de similarité appropriée à nos signatures. Par contre nous n'avons pas encore abordé le problème de la fusion des informations de couleur et de texture. Dans de précédents travaux (Souvannavong et al. [16]) dont les expériences ne sont pas reprises ici, nous avons observé que la fusion était plus efficace au moment de la recherche et non pas au niveau de la quantification ou de l'analyse de la sémantique latente. C'est pour cela que nous avons effectué une présentation de la LSA pour une caractéristique. La mesure de similarité employée dans les expériences permet alors de réaliser la fusion des caractéristiques notamment de couleur et de texture. Elle est définie par une somme pondérée des mesures de similarité sur chaque caractéristique :

$$sim(s,t) = \sum_{c \in \{couleur, texture\}} \alpha_c cos(s^t U_c, t^t U_c) \tag{2.7}$$

Par commodité , les poids α_c sont unitaires et la dureté relative de la LSA est la même pour les deux caractéristiques. Toutefois dans le cadre d'une application intéractive, ces paramètres peuvent être individuellement modifiés par l'utilisateur. L'annexe C à la page 119 montre le système de recherche que nous avons développé pour visualiser l'impact des différents paramètres.

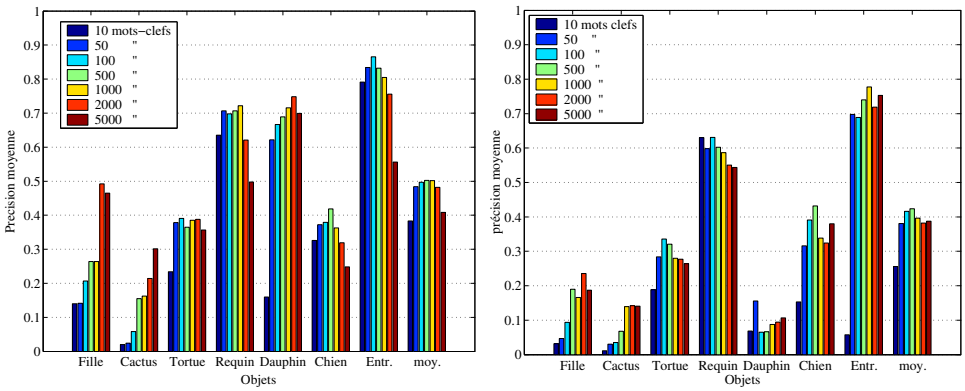
2.4 Evaluation des performances

Pour évaluer les performances, les expériences sont réalisées dans le cadre de la recherche d'information et nous utilisons la valeur de la précision moyenne pour mesurer l'efficacité des systèmes pour un ensemble de requêtes. L'évaluation repose tout d'abord sur le choix du nombre de mots clefs visuels, puis sur l'impact de la LSA. Enfin nous comparons la méthode proposée à une approche répandue utilisant la mesure « Earth Mover's Distance » (EMD) qui est alors présentée.

2.4.1 Choix du nombre de mots clefs visuels

Cette section a pour but d'étudier l'impact du nombre de mots clefs visuels sur les performances du système. Le choix de ce nombre est souvent arbitraire ou obtenu par l'observation des performances moyennes. En effet le nombre optimal de mots clefs visuels pour décrire le contenu visuel est étroitement lié aux requêtes comme le montre la figure 2.6 de la page 34. Le choix ne sera donc pas optimal pour toutes les requêtes envisageables. Toutefois nous observons sur la figure 2.6 qu'en moyenne les performances sont stables sur l'intervalle [100..1000] qui est assez large. Ce qui

nous permet de choisir un nombre de mots clefs visuels sans une dégradation trop importante du contenu sur l'ensemble des objets étudiés. Les dégradations de performance proviennent de deux sources lorsqu'une représentation par les vecteurs de dénombrement est utilisée. Soit trop peu de mots clefs visuels sont choisis. Tous les contenus sont décrits par une palette très pauvre et tout semble identique. Soit beaucoup de mots clefs visuels sont choisis. Tous les contenus sont décrits par une palette très riche et tout semble différent. Dans ce dernier cas, une métrique comme la EMD permet de conserver de bonnes performances.



(a) Pour les histogrammes de couleur

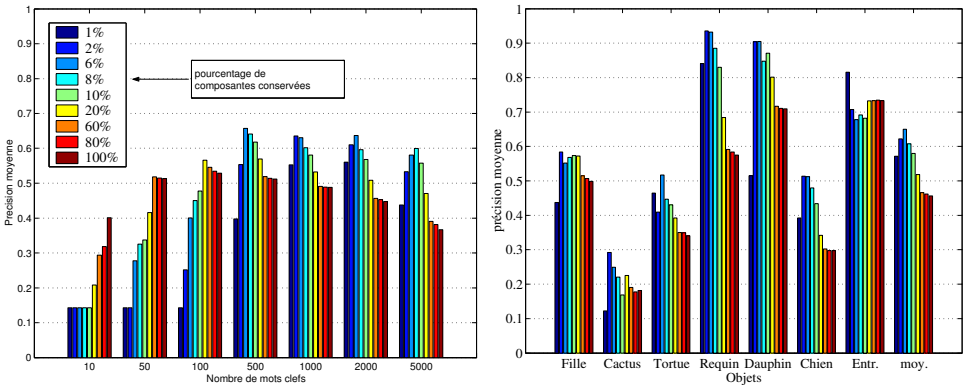
(b) Pour les énergies des réponses aux filtres de Gabor

FIG. 2.6 – Choix du nombre de mots clefs visuels pour la quantification.

2.4.2 Impact de la LSA

Cette section a pour but de démontrer le gain de performances obtenu avec la LSA. Dans un premier temps nous étudions l'impact des différents paramètres sur la recherche d'objets. Nous avons étudié la précision moyenne du système en fonction du nombre de mots clefs visuels et de la dureté de la LSA sur la série de dessins animés. Le résultat se trouve sur la figure 2.7 de la page 35 sous forme d'histogramme. Les meilleures précisions sont obtenues entre 500 et 2 000 mots clefs visuels en conservant entre 6% et 8% des composantes. Comme nous pouvons l'observer, la LSA permet d'améliorer significativement les performances avec une augmentation de 10% de la précision moyenne. De plus uniquement 6% à 8% des composantes sont nécessaires pour obtenir les meilleures performances. Les signatures de couleur et de texture sont donc réduites à une taille de moins de 200 composantes pour 2 000 mots clefs visuels. Outre un gain de place, cette réduction de la taille apporte également un gain de temps non négligeable pour toutes les opérations de comparaison des signatures.

Dans un second temps, la LSA est évaluée sur les vidéos des nouvelles télévisées. La figure 2.8 de la page 35 donne un bon aperçu des performances moyennes en fonction du nombre de composantes conservées lors de la projection. Les performances moyennes indiquent un gain de 5% entre l'absence de la LSA (100% des composantes conservées) et une projection draconienne (2% des composantes conservées). Par contre au cas par cas, les comportements ne sont pas tous identiques.

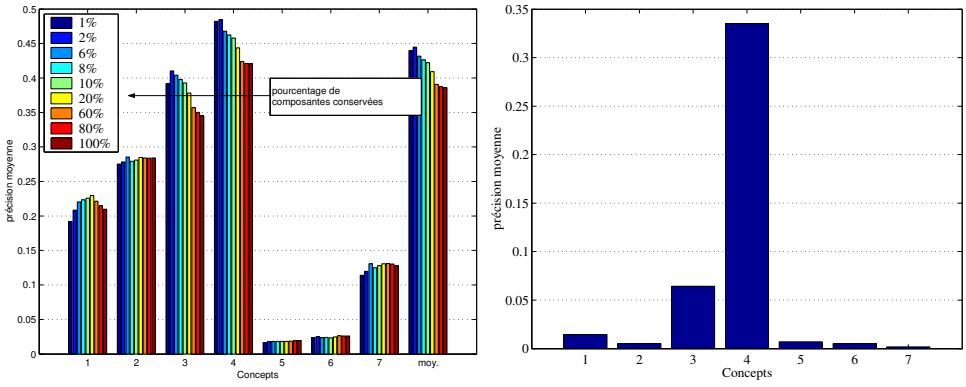


(a) Evolution moyenne en fonction du nombre de mots clefs visuels.

(b) Evolution par objet pour 2 000 mots clefs visuels.

FIG. 2.7 – Evolution de la précision moyenne en fonction du nombre de mots clefs visuels et du pourcentage de composantes actives (dureté de la LSA).

D'une part des performances très faibles sont obtenues pour la recherche de plans avec Bill Clinton(5), un avion(6) ou un train(7). Ceci est compréhensible puisque les requêtes sont effectuées sur l'ensemble de l'image or ces concepts sont très localisés. De plus ils ne sont pas fréquents dans la base de données comme le montre l'histogramme des connaissances a priori 2.8(b). Les concepts exprimés au niveau de l'ensemble du plan sont retrouvés plus efficacement même lorsqu'ils ne sont pas particulièrement fréquents comme les concepts basket-ball(1), climat(2) et studio(3). La LSA ne permet pas systématiquement d'améliorer la précision moyenne de la recherche mais il est toujours possible d'effectuer une réduction significative de la taille des projections sans altérer significativement les performances.



(a) Evolution moyenne par concept en fonction de la taille de la projection.

(b) Connaissance a priori.

FIG. 2.8 – Résultats sur des plans de journaux télévisés TRECVID avec 2 000 mots clefs visuels.

La comparaison de la LSA avec une représentation par les espaces vectoriels d'images a fourni

des résultats très concluants. Toutefois la méthode d'analyse de la sémantique latente présentée requiert la quantification des vecteurs décrivant le contenu des images tout comme la représentation par les espaces vectoriels d'images. Une comparaison avec une autre méthode utilisant les régions sans quantification serait donc pertinente.

2.4.3 Comparaison avec l'EMD

Dans cette section, nous proposons de comparer l'approche présentée utilisant la LSA avec une approche directe sur les images segmentées. Nous entendons par approche directe une méthode qui utilise les vecteurs décrivant le contenu des régions en l'état. C'est à dire que les distances sont calculées en utilisant par exemple les histogrammes de couleur ou les énergies des réponses aux filtres de Gabor. Pour cela nous allons utiliser la distance EMD (Rubner et al. [14]) pour comparer les différentes régions composant une image. Cette distance permet de calculer le coût minimal pour transformer une distribution en une autre. Une interprétation plus imagée consiste à assimiler les distributions à un relief ; la distance entre les deux reliefs est alors la quantité minimale de terre à déplacer de la première distribution de terre à la deuxième. Une distribution est représentée par un vecteur d'occurrence d'un nombre fini d'éléments qui ne sont pas obligatoirement communs à toutes les distributions. La mesure EMD requiert alors une distance de base entre les éléments de la distribution afin de calculer le coût de la transformation. Ce dernier est calculé en résolvant le problème dit du transporteur (« transportation problem »). Soit $P = \{(p_1, w_{p1}), ..., (p_m, w_{pm})\}$ la première signature avec m régions de poids w_i et décrites par p_i. Soit $Q = \{(q_1, w_{q1}), ..., (q_n, w_{qn})\}$ la deuxième signature. Désignons par $D = [d_{ij}]$ l'ensemble des distances entre les régions p_i de la première signature et les régions q_j de la deuxième. Nous voulons trouver le flux global $F = [f_{ij}]$, où f_{ij} est le flux entre les régions p_i et q_j, qui minimise la fonction de coût suivante :

$$C(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij} \tag{2.8}$$

avec les contraintes :

$$f_{ij} \geq 0, \ \forall (i, j) \tag{2.9}$$

$$\sum_{j=1}^{n} f_{ij} \leq w_{pi}, \ \forall i \tag{2.10}$$

$$\sum_{i=1}^{m} f_{ij} \leq w_{qj}, \ \forall j \tag{2.11}$$

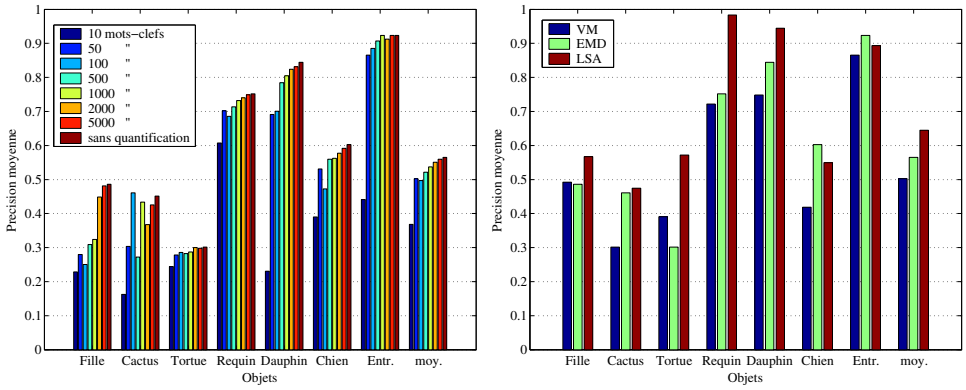
$$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min(\sum_{i=1}^{m} w_{pi}, \sum_{j=1}^{n} w_{qj}) \tag{2.12}$$

Une fois le problème du transporteur résolu (Rubner et al. [14]), la distance entre l'image P et Q est donnée par :

$$EMD(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}} \tag{2.13}$$

Dû aux limitations des capacités de stockage, les éléments des distributions sont souvent quantifiés et les distributions sont décrites par un vecteur de dénombrement basé sur un dictionnaire commun. La figure 2.9(a) de la page 37 fournit les performances en terme de précision moyenne pour différentes tailles du dictionnaire (c'est à dire différents nombres de mots clefs visuels ou vecteurs de quantification). Comme nous pouvons le voir, les performances diminuent avec la taille du dictionnaire. Et contrairement au cas précédent les performances croissent continuellement avec la taille du dictionnaire. En effet, l'EMD calcule une distance à partir de la valeur des mots clefs visuels et non pas seulement à partir des vecteurs de dénombrement. Toutefois, la quantification ne dégrade pas tant les performances à partir de 1 000 mots clefs visuels. Et il est tout à fait raisonnable d'effectuer la quantification pour éviter d'avoir une base de données des signatures trop importante. Remarquons que la quantification ne diminue pas les temps de calcul nécessaire pour résoudre le problème du transporteur. Et le temps requis pour effectuer les comparaisons constitue un frein majeur à la méthode.

Finalement, une comparaison des performances par objets est fournie dans les figures 2.9 et 2.10 aux pages 37 et 38. L'EMD qui conserve la valeur des mots clefs visuels obtient de meilleures performances qu'une approche par espace vectoriel d'image. Les performances de la LSA sont surprenantes puisqu'elles sont meilleures que celles obtenues avec l'EMD. Nous pouvons remarquer sur la figure 2.10 que l'EMD fournit une précision importante pour de faibles valeurs du rappel, puis la précision chute rapidement. Au contraire, la LSA effectue un départ avec une précision plus modérée mais qui se maintient plus longtemps. Ceci est dû aux différences entre les deux méthodes. La première méthode, c'est à dire l'EMD, est beaucoup plus précise et elle est donc moins tolérante aux changements. Par contre la LSA effectue sa mesure sur des signatures qui approxime le contenu et elle permet alors de retrouver des contenus légèrement différents et toujours pertinents. Les expériences n'ont pas pu être réalisées sur les vidéos de TRECVID puisque les temps de calcul de l'EMD étaient trop longs.

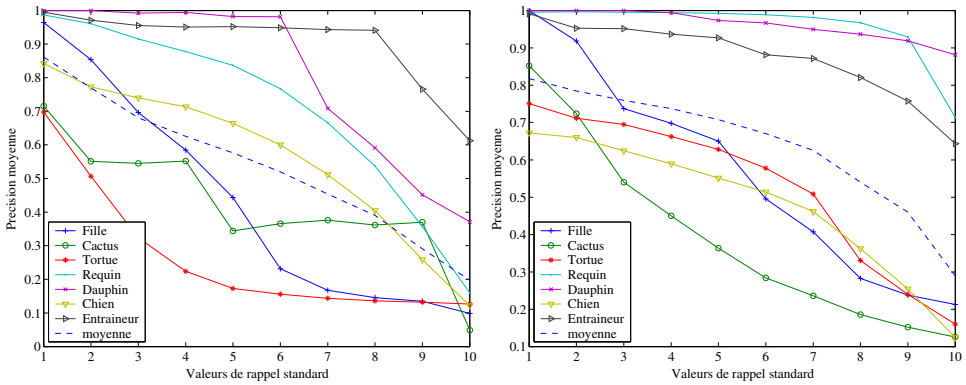


(a) Mesure EMD pour comparer deux images.

(b) Comparaison des différentes méthodes utilisant les régions.

FIG. 2.9 – Etude de la mesure EMD et comparaison avec le VSM et la LSA.

Pour améliorer la représentation du contenu par les espaces vectoriels d'image, nous proposons une représentation floue de la quantification.

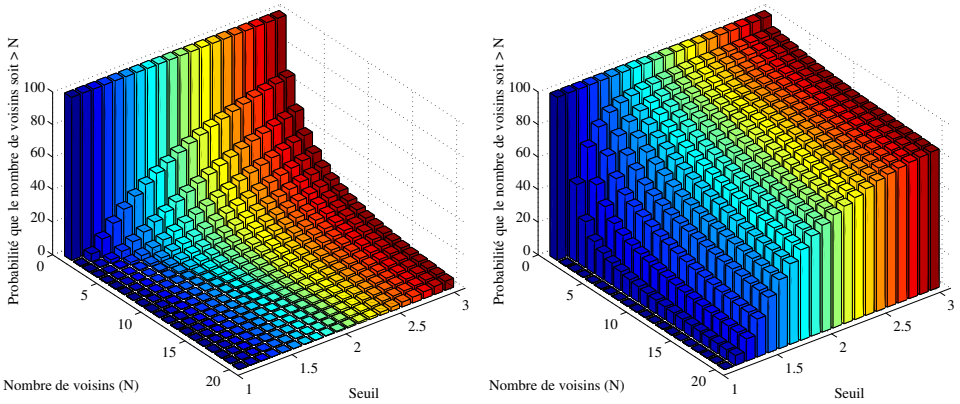


(a) EMD (b) LSA

FIG. 2.10 – Comparaison détaillée du système utilisant l'EMD et du système utilisant la LSA

2.5 Utilisation d'une représentation floue

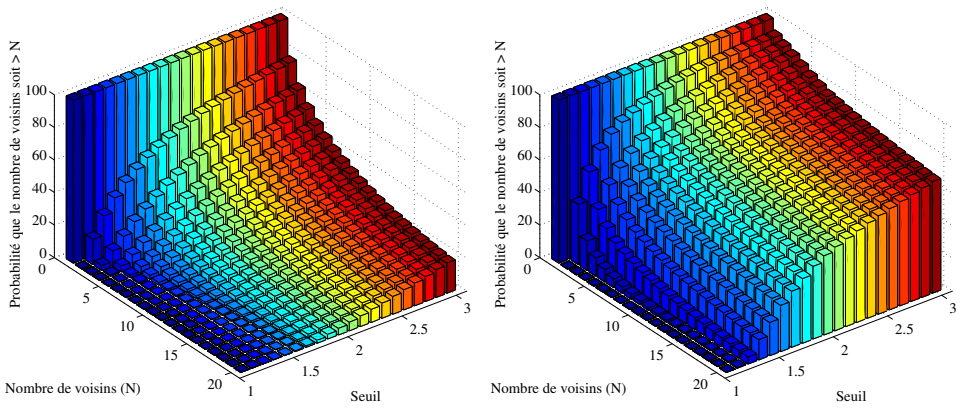
La représentation du contenu par un vecteur de dénombrement accentue à l'extrême les différences entre les régions. Deux régions attribuées à deux mots clefs visuels différents sont absolument différentes tandis que deux régions attribuées au même mot clef sont parfaitement similaires. La LSA ne permet pas de s'affranchir totalement de cette mesure binaire des différences. Il convient alors d'utiliser une représentation floue. Pour cela les régions sont dorénavant attribuées aux n plus proches mots clefs visuels avec la contrainte que leur similarité soit suffisante. Pour cela nous définissons deux paramètres lors de la quantification, le nombre de voisins n_r et un facteur f_s qui permet de calculer une distance maximum entre une région et son mot clef le plus distant. Le vecteur de dénombrement d'un plan (ou d'une image) est alors mis à jour pour chaque région r de la manière suivante : Soit $d_{min}(r)$ la distance de r au plus proches mots clefs visuels. Nous définissons le seuil $s = f_s \times d_{min}(r)$. L'index des mots clefs visuels satisfaisants $d(r, mc_i) \leq d_{min}(r)$ est incrémenté d'une unité dans la limite des n_r plus proches mots clefs visuels. Les figures 2.11 et 2.12 aux pages 39 et 40 illustrent la quantité de régions satisfaisant ces conditions pour différents facteurs f_s et différentes tailles du voisinage n_r. Les régions décrites par la couleur sont faiblement impliquées pour des facteurs raisonnables ($< 1,5$). Par contre les régions décrites par la texture sont particulièrement sensible à la représentation floue. Dans les expériences qui suivront, n_r et f_s sont empiriquement fixés à 10 et $1,25$. Le facteur est choisi de manière à ne pas rendre toutes les régions similaires. Une valeur de $1,25$ assure que deux régions semblables r_1 et r_2, c'est à dire décrites par le même mot clef mc_j, sont au plus distantes de $2,5 \times \max\{d(mc_j, r_1), d(mc_j, r_2)\}$. D'autres part nous pensons que dix mots clefs visuels sont amplement suffisants pour compenser les erreurs de quantification. La figure 2.13 de la page 40 montre l'évolution des performances en fonction du nombre de mots clefs visuels et de la dureté de la LSA sur la série de dessins animés à gauche et sur l'ensemble de vidéos de TRECVID à droite. Ces figures sont comparées à l'approche initiale dont les résultats sont respectivement sur les figures 2.7(b) et 2.8(a) des pages 35 . La première observation est une augmentation générale des performances d'une représentation simple par les IVSM (ces performances sont données par une LSA de dureté nulle, c'est à dire sans projection soit 100% des



(a) Quantification de la couleur (b) Quantification de la texture

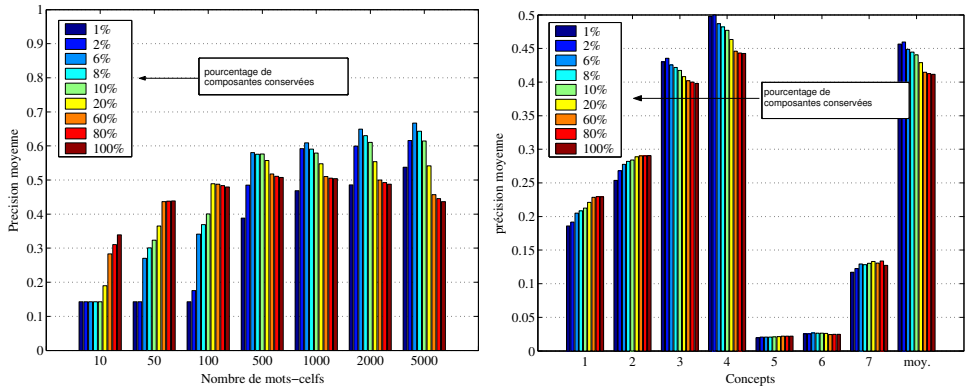
FIG. 2.11 – Fiabilité de la quantification par 1 000 mots clefs visuels sur des dessins animés

composantes conservées). Toutefois cette augmentation ne rivalise pas avec le gain significatif et toujours présent de la LSA. L'approche par la quantification floue ne permet donc pas d'obtenir les mêmes performances qu'une analyse de la sémantique latente qui fournit une indexation efficace. La deuxième observation est un gain général de performances pour une quantification au-delà de 2 000 mots clefs visuels (figure 2.13(a)), ce qui prouve l'efficacité d'avoir une quantification floue dès lors que de nombreux mots clefs visuels sont nécessaires pour correctement décrire le contenu. Cette observation est confirmée sur l'évaluation à grande échelle réalisée sur l'ensemble de TRECVID (figure 2.13(b)). Nous noterons tout de même que pour la classe *basket-ball*(1) le gain de performance habituellement procuré par la LSA est inexistant avec une quantification floue. Par contre la dimension des signatures latentes peut toujours être réduite.



(a) Quantification de la couleur (b) Quantification de la texture

FIG. 2.12 – Fiabilité de la quantification par 2 000 mots clefs visuels sur les plans de journaux télévisés TRECVID



(a) Sur la série de dessins animés (b) Sur les plans de journaux télévisés de TRECVID

FIG. 2.13 – Quantification floue et performance de l'analyse de la sémantique latente

2.6 Description localisée du contenu latent

La description du contenu latent repose sur une transformation linéaire qui a ses limites. Nous proposons d'introduire une description locale du contenu pour tenir compte des variations locales. L'idée est de détecter les relations entre les régions en fonction du contexte. Pour cela plusieurs contextes contenant des plans similaires sont définis ; ensuite l'analyse de la sémantique latente est menée sur chaque contexte. La première partie définit la notion d'analyse sémantique locale, ensuite nous abordons le problème de l'indexation et de la recherche dans ce nouveau cadre.

2.6.1 Analyse locale du contenu latent

Afin d'améliorer les effets de la SVD qui est une opération linéaire, nous proposons de l'appliquer localement à des partitions P_i de l'espace des plans. Cette opération a pour objectif de détecter les directions principales avec plus de précision. Ainsi nous espérons que la LSA sera localement plus efficace.

Pour créer une partition de l'espace des plans $\mathcal{P} = \{P_i\}$, l'algorithme k-means est utilisé. L'analyse latente est ensuite appliquée à chaque partition, ce qui donne la décomposition suivante pour la partition i :

$$C_i = U_i D_i V_i^t \tag{2.14}$$

Avec C_i, la matrice d'occurrence des mots clefs visuels dans les plans de la partition P_i. A présent, la définition du nombre de composantes à conserver par l'analyse n'est pas immédiate puisque la taille des matrices n'est pas identique dans toutes les partitions. Nous allons donc travailler en pourcentage de composantes conservées l et empiriquement ce pourcentage est imposé commun à toutes les partitions. Ce qui donne :

$$\hat{C}_{i,k} = U_{i,k} D_{i,k} V_{i,k}^t \tag{2.15}$$

$$\text{où } k = l \times \min(\text{nombre de plans} \in P_i, \text{nombre de mots clefs visuels}) \tag{2.16}$$

Dans la suite, $U_{i,k}$ sera noté $U_i(l)$ pour faire apparaître le pourcentage l et non plus directement le nombre de composantes k.

A partir de cette nouvelle analyse locale du contenu latent, une nouvelle représentation des plans est définie conjointement avec une nouvelle mesure de similarité.

2.6.2 Indexation et recherche

L'approche directe consiste à indexer un plan en fonction de la partition à laquelle il appartient. Son index est alors composé de l'identifiant de la partition et de sa projection dans l'espace correspondant. Soit $s \in P_i$ et $p = U_i^t(l)s$, la mesure de similarité avec un plan q est alors définie comme il suit :

$$sim(s,t) = \begin{cases} cos(p,q) & \text{if } q \in P_i \quad (q = U_i^t(l)t) \\ -1 & sinon \end{cases} \tag{2.17}$$

Malheureusement, cette formulation n'autorise pas les comparaisons inter partition mais uniquement intra partition. Cet inconvénient devient particulièrement ennuyeux lorsque les requêtes concernent les objets. Ces derniers n'occupant qu'une faible partie de l'image ne seront pas systématiquement classés dans la meilleure partition. D'ailleurs, il est fort probable que les objets recherchés ne sont pas tous dans la même partition mais répartis sur plusieurs d'entres elles. Cela suggère de projeter les plans dans les espaces latents de toutes les partitions, d'effectuer les comparaisons avec les plans correspondants puis de combiner les mesures de similarité en une valeur unique. Cette seconde proposition néglige les erreurs de projection qui peuvent être élevées tandis que les plans projetés sont proches. Dans ce cas, la mesure de proximité n'est pas fiable. Pour prendre en compte ces paramètres, nous proposons une mesure de similarité de la forme :

$$sim(s,t) = \max_i \{(cos(s,\hat{s}_i)(cos(t,\hat{t}_i))^\gamma cos(p,q_i)\} \tag{2.18}$$

$$\begin{aligned} \text{où } p_i = U_i^t(l)s \quad & \text{et } \hat{s} = U_i(l)p_i \\ \text{et } q_i = U_i^t(l)t \quad & \text{et } \hat{t} = U_i(l)q_i \end{aligned}$$

Les deux premières fonctions cosinus mesurent la similarité entre le plan et sa reconstruction après projection. Le cosinus permet d'avoir une mesure normalisée quelque soit la valeur de l. Les cosinus sont élevés à la puissance γ pour pondérer l'impact des erreurs de projection. Finalement le troisième cosinus mesure la similarité entre les projections. Intuitivement la mesure de similarité proposée permet de favoriser les plans similaires dans une partition et ayant une faible erreur de projection. L'inconvénient de cette formulation est la présence des termes $cos(s,\hat{s}_i)$ et $cos(s',\hat{s}'_i)$. Sous cette forme il est nécessaire de connaître les vecteurs dans leur intégrité puisque les estimations de s à partir de la projection dépendent du nombre de composantes conservées par les LSA et par conséquent la valeur du cosinus doit être recalculée. Toutefois un raisonnement dans les espaces singuliers permet de s'affranchir de ce défaut. Soit :

$$\begin{aligned} p_i = U_i^t(1)s \text{ et } p_i(l) = U_i^t(l)s \\ e_i(l) = p_i - p_i(l) \text{ avec } e_i(l)\cdot p_i(l) = 0 \end{aligned}$$

Nous avons alors :

$$\begin{aligned} cos(s,\hat{s}_i) &= cos(p_i, p_i(l)) \\ &= \frac{p_i \cdot p_i(l)}{||p_i|| \cdot ||p_i(l)||} = \frac{p_i(l)\cdot p_i(l)}{||p_i(l)+e_i(l)|| \cdot ||p_i(l)||} \\ &= \frac{||p_i(l)||}{\sqrt{||p_i(l)||^2 + ||e_i(l)||^2}} \end{aligned}$$

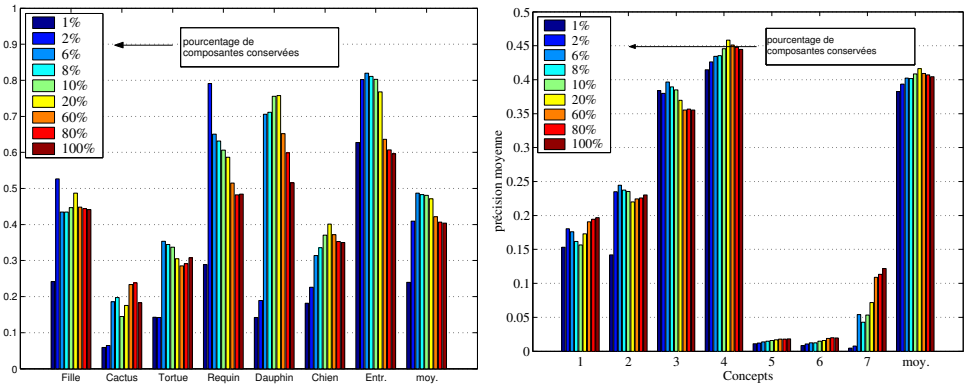
Il est juste nécessaire de connaître l'erreur $||e_i(l)||$ et la projection $p_i(l)$ au moment de l'indexation pour pouvoir calculer $sim(s,t), \forall l' \leq l$ dans l'équation 2.18 :

$$sim(s,t) = \max_i \left\{ \left[\frac{||p_i(l')||}{\sqrt{||p_i(l)||^2 + ||e_i(l)||^2}} \frac{||q_i(l')||^2}{\sqrt{||q_i(l)||^2 + ||f_i(l)||^2}} \right]^\gamma cosinus(p_i(l'), q_i(l')) \right\} \tag{2.19}$$

$||f_i(l)||^2$ est tout comme $||e_i(l)||^2$ l'erreur faite par la projection lors de l'indexation pour le plan de la base de données.

2.6.3 Expériences

Avec une grande surprise, aucune amélioration n'est observée. Les meilleures performances que nous pouvons voir sur les figures 2.14(a) et 2.14(b) à la page 43, ont été obtenues avec une valeur de γ égale à 2. Comparées à l'approche initiale (se référer aux figures 2.7(b) et 2.8(a) des pages 35 et 35), les performances subissent une dégradation notable. Nous avons remarqué que le principal acteur dans cette dégradation est la mesure des erreurs. Elle est pourtant nécessaire puisqu'une valeur de γ nulle détériore plus les performances. Cette méthode n'est donc pas adaptée à l'analyse du contenu latent puisqu'elle repose trop sur la notion d'erreur de reconstruction qui va à l'encontre de celle-ci.



(a) Sur la série de dessins animés avec deux modèles.

(b) Sur la base de données TRECVID avec dix modèles.

FIG. 2.14 – Performances obtenues à l'aide d'une description localisée du contenu latent pour $\gamma = 2$.

2.7 Description enrichie du plan

Les méthodes d'indexation qui utilisent les régions des images reposent beaucoup sur l'efficacité des algorithmes de segmentation utilisés. Or la tâche de la segmentation est très délicate et actuellement une segmentation fiable n'existe pas. D'autre part ni les problèmes d'échelle à laquelle le contenu est acquis, ni la quantification utilisée par la suite n'améliorent les choses. Nous avons déjà vu dans la partie précédente comment réduire les défauts de la quantification en la remplaçant par une quantification floue. A présent, nous allons aborder les problèmes d'échelle et de segmentation en enrichissant les signatures des plans. L'enrichissement s'effectue de trois manières. La première méthode utilise plusieurs images clefs du plan, la seconde utilise plusieurs niveaux de segmentation et la dernière utilise les retours des utilisateurs. Nous présentons tout d'abord l'idée générale. Puis nous étudierons en particulier l'asservissement pour finir par les résultats des expériences.

2.7.1 Méthodes d'enrichissement

L'approche par le dénombrement offre l'avantage de pouvoir intégrer de l'information supplémentaire sans pour autant changer la taille de la signature. La solution est d'effectuer le dénombre-

ment sur plusieurs sources d'information possédant le même dictionnaire. La signature finale est alors composée de la somme pondérée des signatures obtenues sur chaque source d'information. L'avantage majeur est de conserver une taille fixe tout en incluant plus d'information, à la fois redondante pour confirmer la présence d'un mot clef ou supplémentaire pour mettre en évidence un mot clef plus discret. Les temps de recherche ne sont alors pas affectés par cet enrichissement. Par contre, l'utilisation d'un plus grand nombre de sources demande une analyse plus lourde lors de l'indexation comme nous allons le voir.

Nous avons identifié trois nouvelles sources d'information qui sont à notre disposition pour améliorer la robustesse de notre signature de base. Tout d'abord plusieurs niveaux de segmentation sont utilisés pour augmenter la robustesse aux changements d'échelle lors de l'acquisition. Ensuite plusieurs images sont utilisées pour caractériser le plan. Une indexation plus robuste au mouvement et à la segmentation est alors obtenue. Finalement, un système de recherche avec une boucle d'asservissement permet à l'utilisateur d'identifier les plans pertinents pour améliorer la recherche. Toutefois ces plans sont introduits suivant une méthode légèrement différente.

2.7.2 Asservissement

La boucle d'asservissement a pour objectif d'améliorer le résultat de la recherche en tenant compte du jugement de la recherche précédente fait par l'utilisateur. Ce dernier sélectionne les plans pertinents, les plans neutres et les plans non pertinents parmi ceux retournés par le système. Le système de recherche utilise cette information a posteriori pour améliorer la requête et le résultat de la prochaine recherche.

L'algorithme de Rocchio [12] est l'un des plus populaires et des plus utilisés par la communauté scientifique de la rechercher d'information. Lorsque les documents sont ordonnés en fonction de leur similarité à la requête, la requête idéale doit favoriser les documents pertinents et pénaliser les autres. Rocchio a donc proposé de maximiser la moyenne des documents pertinents ($d \in P$) moins la moyenne des autres documents ($d \in N$). Le résultat est une requête Q', telle que :

$$Q' = aQ + b \sum_{P} d - c \sum_{N} d \tag{2.20}$$

Q est la requête d'origine et a, b et c sont les poids de Rocchio qui ont souvent les valeurs suivantes :

$$a = 1, b = \frac{1}{Card(P)}, c = \frac{1}{Card(N)}$$
$$a = b = c = 1$$
$$c = 0$$

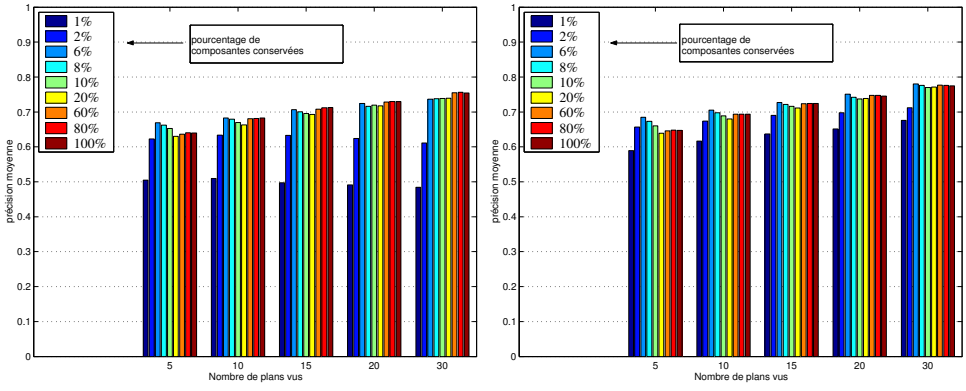
Ce calcul d'une nouvelle requête s'intègre parfaitement dans notre cadre. Et le principe de calcul d'une nouvelle requête est très proche de celui décrit dans la partie précédente. Dans les expériences nous fixerons $a = b = c = 1$ ou $a = b = 1, c = 0$.

2.7.3 Expériences

L'inclusion de plusieurs images clefs ou de plusieurs niveaux de segmentation apporte uniquement une amélioration très faible sur les performances (Souvannavong et al. [17]). Suite aux temps

de calcul particulièrement longs pour extraire toutes les caractéristiques, les expériences ont été effectuées uniquement sur la série de dessins animés où l'intérêt de l'analyse multi échelles ou multi images n'est pas vraiment mis en valeur. Pour toutes ces raisons, les expériences de ce mémoire se concentrent essentiellement sur l'étude de la boucle d'asservissement qui est un point crucial des systèmes de recherche d'information présents et à venir.

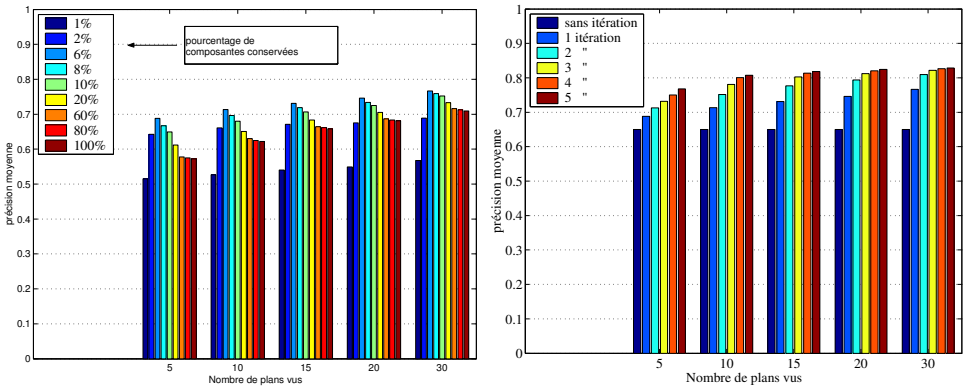
Les expériences consistent à évaluer l'impact de la boucle d'asservissement sur les performances de la recherche de plans vidéo. Pour cela l'influence de trois paramètres est étudiée : la dureté de la LSA, le nombre de plans vus par l'utilisateur et le nombre d'itérations de l'asservissement. Le nombre de plans vus par l'utilisateur, noté n_v, correspond au nombre de plans retournés par le système que l'utilisateur doit évaluer comme étant pertinent ou non. Ces plans sont ensuite combinés à la requête précédente pour la compléter de manière efficace. Les plans présentés à l'utilisateur ne sont pas systématiquement les n_v plus pertinents car à l'itération n, les plans pertinents précédemment choisis ont de fortes chances de se trouver parmi les premiers. Pour éviter à l'utilisateur une évaluation redondante et inutile du contenu, les plan retournés subissent un décalage progressif à chaque itération. A la première itération, les n_v premiers sont retournés et à l'itération n, les plans de $(n-1)n_v$ à $(n)n_v$ sont retournés. Les figures 2.15 à la page 45 montrent l'évolution des performances moyennes en fonction de la dureté de la LSA et du nombre n_v pour une itération. La figure de gauche inclut les éléments positifs et négatifs dans la formulation d'une nouvelle requête, tandis que la figure de droite inclut uniquement les éléments positifs. Les performances du système s'améliorent avec le nombre de plans vus par l'utilisateur. Ainsi un gain de plus de 5% est réalisé entre la visualisation de 5 et de 30 plans. Par contre l'analyse latente perd sensiblement son intérêt. En effet, l'augmentation de sa dureté ne permet qu'une très légère amélioration de la précision moyenne qui s'amenuise avec l'augmentation de v_n. Au final le système atteint un gain de plus de 10% lorsque 15 plans sont visualisés soit en utilisant les retours positifs et négatifs soit en utilisant uniquement les retours positifs. Une légère amélioration est notable lorsque seulement les éléments positifs sont inclus dans la boucle d'asservissement ce qui apporte un gain d'environ 18% par rapport à une approche sans boucle d'asservissement.



(a) Utilisation des positifs et des négatifs. (b) Utilisation des positifs seulement.

FIG. 2.15 – Utilisation d'une boucle d'asservissement avec une itération sur la série de dessins animés.

Nous avons approfondi l'étude de la boucle d'asservissement en considérant que l'utilisateur prend le temps de sélectionner les régions pertinentes. Cela permet de s'affranchir de l'information contenue dans l'arrière plan et de concentrer la recherche sur les objets. En effet l'inclusion de l'arrière plan biaise les résultats, notamment dans le cas présent où les objets apparaissent majoritairement sur un fond similaire. La figure 2.16 à la page 46 présentent les résultats de deux expériences. Dans un premier temps, la précision moyenne est étudiée pour différentes valeurs de n_v et différentes duretés de la LSA (figure de gauche 2.16(a)). Dans un second temps nous avons étudié l'évolution de la précision moyenne en fonction du nombre d'itérations effectuées pour une dureté de 6% (figure de droite 2.16(b)). Nous en avons profité pour rappeler les performances initiales sans la boucle d'asservissement. Contrairement au cas précédent, la LSA offre une amélioration significative des performances. Toutefois elle décroît lorsque v_n croît. Ainsi entre 5 et 30, le gain de la LSA passe de 10% à 5%. De la comparaison des deux figures, nous concluons que le facteur principal dans l'amélioration des performances à l'aide de la boucle d'asservissement est le nombre de plans positifs sélectionnés. En effet les performances sont très proches lorsque deux itérations sont effectuées en visualisant 5 plans ou lorsqu'une itération est effectuée en visualisant 10 plans.

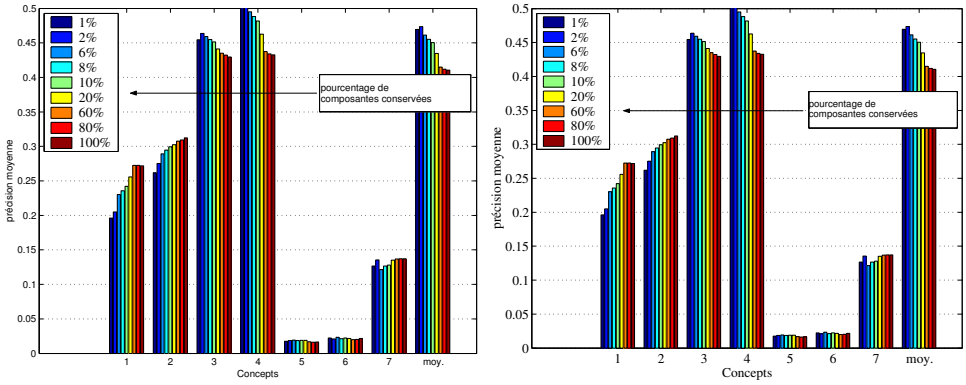


(a) Avec une itération.

(b) Evolution en fonction du nombre d'itérations pour plans vus.

FIG. 2.16 – Utilisation d'une boucle d'asservissement et sélection des objets dans les images clefs retournées sur la série de dessins animés.

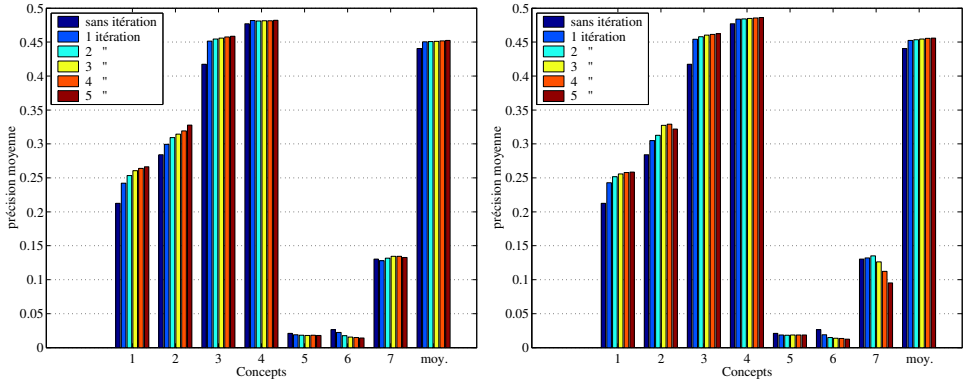
La boucle d'asservissement a également été utilisée pour réaliser des recherches sur l'ensemble de TRECVID (figure 2.17 de la page 47) avec une seule itération en premier lieu. La figure de gauche fournit les résultats obtenus lorsque les plans pertinents et non pertinents sont inclus dans la nouvelle requête. Tandis que la figure de droite fournit les résultats obtenus lorsque uniquement les plans pertinents sont inclus dans la nouvelle requête. Une comparaison relative indique que l'introduction des plans non pertinents ne change pas réellement les effets de la boucle d'asservissement. La diversité du contenu des nouvelles télévisées est telle que la suppression d'un type de plans non pertinents laisse la place à une autre catégorie de plans non pertinents. L'annexe C à la page 119 fournit des exemples de requêtes avec et sans boucle d'asservissement sur l'ensemble de TRECVID. En particulier un exemple est fourni lorsqu'une personne est sélectionnée comme requête. Les figues 2.18(a) et 2.18(a) de la page 47 synthétisent les performances obtenues lorsque plusieurs



(a) Utilisation des positifs et des négatifs. (b) Utilisation des positifs seulement.

FIG. 2.17 – Utilisation d'une boucle d'asservissement avec 10 plans visualisés et une itération sur les plans de journaux télévisés.

itérations sont mises en oeuvres. D'autre part, les résultats obtenus sans boucle d'asservissement sont également représentés pour mieux souligner les effets de l'asservissement. La dureté de la LSA est fixée à 10%. Cette valeur fournit des performances raisonnables sur l'ensemble des classes. Le gain majeur est réalisé dès la première itération que 10 ou 20 plans soient visualisés. Ensuite la précision moyenne se stabilise pour les autres itérations. Les deux premiers concepts *basket-ball* et *studio* échappent à cette règle et trois nouvelles itérations permettent d'atteindre le même gain que la première. La visualisation de 20 plans par itération ne semble pas fournir un intérêt majeur, au contraire il semble préférable d'effectuer plus d'itérations avec un plus petit nombre de plans visualisés.

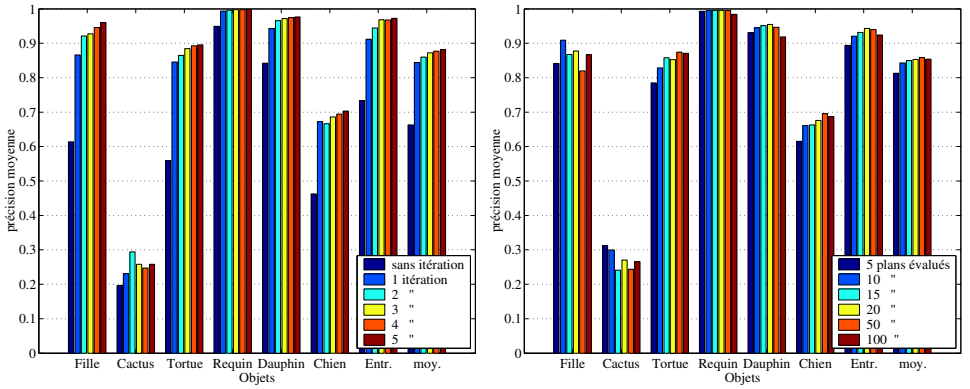


(a) 10 plans vus par itération. (b) 20 plans vus par itération.

FIG. 2.18 – Evolution des performances en fonction d'une nombre d'itérations pour 10% de composantes conservées par la LSA sur les plans de journaux télévisés.

Dans notre modélisation de la boucle d'asservissement et la simulation d'un utilisateur, nous

avons considéré uniquement les premiers plans retournés par le système. Ces derniers sont ensuite évalués par l'utilisateur puis utilisés par la boucle d'asservissement. Un décalage progressif est introduit entre chaque itération pour éviter uniquement l'évaluation des premiers plans au sens strict, qui sont souvent similaires d'une itération à l'autre. Malgré cela, cette procédure ne modélise pas parfaitement le comportement de l'utilisateur et ce n'est pas la meilleure approche pour optimiser l'asservissement. L'utilisateur réel est beaucoup plus souple sur le nombre de plans à évaluer. Il ne prendra pas forcément les premiers ni les derniers proposés. Cet aspect est important pour le système d'asservissement puisque cette sélection permet d'enrichir la requête avec des contenus différents. Une nouvelle évaluation est proposée qui permet d'annoter aléatoirement des sous-ensembles de taille v_n jusqu'à obtenir l'évaluation de v_n plans pertinents. Le tirage aléatoire des ensembles est réalisé dans la limite du raisonnable et s'arrête dès lors que v_n plans pertinents sont choisis. Une nette amélioration des performances est ainsi obtenue (figure 2.19 de la page 48). Sur la série de dessins animés, la barre des 80% de précision moyenne sur l'ensemble des objets est dépassée et une nette amélioration par rapport à une approche sans boucle d'asservissement (c'est à dire sans itération) est observée. Des expériences supplémentaires ont été conduites pour connaître les limites de la boucle d'asservissement. Quelque soit la dureté de la LSA, il en ressortait que la boucle d'asservissement dans sa forme actuelle ne permettait pas d'obtenir une requête parfaite, c'est à dire qui permettait au système d'avoir une précision moyenne de 1. Nous pouvons voir sur la figure 2.19(b) que malgré des nombres élevés de plans pertinents sélectionnés, uniquement le personnage *requin* atteint les performances maximales. Pourtant lorsque 100 plans sont sélectionnés la requête générée est composée de tous les plans pertinents à la requête originale. Nous avons tiré les mêmes conclusions quelque soit la dureté de la LSA.



(a) Evolution en fonction du nombre d'itérations pour 10 plans pertinents sélectionnés.

(b) Evolution en fonction du nombre de plans pertinents sélectionnés.

FIG. 2.19 – Sélection aléatoire des plans visualisés sur la série de dessins animés.

2.8 Conclusion

Ce chapitre était dédié à la représentation du contenu des plans vidéo. Pour cela une adaptation de l'analyse de la sémantique latente (LSA) a été proposée. La LSA a été introduite à l'origine pour l'analyse et l'indexation de document écrits. En raison de son succès et de ses caractéristiques, nous en avons proposé une adaptation à la description du contenu visuel des plans vidéo. Pour cela, nous avons d'abord introduit les espaces vectoriels d'images puis la LSA. Contrairement aux approches générales, la LSA permet de capturer le contenu local des plans en opérant au niveau des régions. Le système a ensuite été évalué sur le problème de la recherche d'information où la requête est un exemple. Les espaces vectoriels d'image et la LSA ont été comparés, puis ils ont été comparés à une méthode qui conserve la totalité de l'information sur les caractéristiques des régions. La LSA a alors montré ses capacités à indexer efficacement les plans vidéos à la fois dans le cadre d'une recherche d'objets et d'une recherche de plans sur une grande quantité de vidéos de nouvelles.

Dans la suite du chapitre, nous avons proposé différentes solutions pour améliorer la représentation par la LSA et la recherche d'information. Pour cela, nous avons présenté la quantification floue, l'analyse locale de la sémantique latente, l'indexation multi échelles et multi images et la boucle d'asservissement. Nous retiendrons en particulier l'amélioration significative des performances qui est obtenue grâce à la boucle d'asservissement. Toutefois, elle ne suffit pas pour générer la requête parfaite qui fournira une précision moyenne avoisinant 1. La difficulté est de gérer la diversité des contenus visuels associés à une classe sémantique.

Pour cela, le prochain chapitre aborde le problème délicat de l'estimation automatiques de concepts sémantiques. L'objectif est d'attribuer automatiquement des classes ou concepts comme studio ou basket-ball aux plans vidéo pour améliorer encore les résultats de la recherche d'information.

Bibliographie

[1] H.D. Cheng, X.H. Jiang, Y. Sun, and J. Wang. Color image segmentation : advances and prospects. *Pattern Recognition*, 34(12) :2259–2281, 2001.

[2] D. Comanicu and P. Meer. Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 :603–619, may 2002.

[3] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6) :391–407, 1990.

[4] P. Felzenszwalb and D. Huttenlocher. Efficiently computing a good segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–104, 1998.

[5] Thomas Hofmann. Probabilistic latent semantic indexing. In *ACM SIGIR*, 1999.

[6] Mikko Kurimo. Indexing audio documents by using latent semantic analysis and som. In Erkki Oja and Samuel Kaski, editors, *Kohonen Maps*, pages 363–374. Elsevier, 1999.

[7] Mingkun Li, Dongge Li, Nevenka Dimitrova, and Ishwar Sethi. Audio-visual talking face detection. In *Proceedings of the International Conference on Multimedia and Expo*, 2003.

[8] Joo-Hwee Lim. Learnable visual keywords for image classification. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 139–145, 1999.

[9] L. Lucchese and S.K. Mitra. Color image segmentation : A state-of-the-art survey. In *Proceedings of the Indian National Science Academy*, pages 207–221, 2001.

[10] Wei-Ying Ma and Hong Jiang Zhang. Benchmarking of image features for content-based image retrieval. In *Thirty-second Asilomar Conference on Signals, System and Computers*, volume 1, pages 253–257, 1998.

[11] Florent Monay and Daniel Gatica-Perez. On image auto-annotation with latent space models. In *Proceedings of the ACM International Conference on Multimedia*, pages 275–278, 2003.

[12] J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System : Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, 1971.

[13] M. Rochery, I.H. Jermyn, and J. Zerubia. Higher order active contours and their application to the detection of line networks in satellite imagery. In *Proceedings of the IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision In Conjunction*, oct 2003.

[14] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 59–66, january 1998.

[15] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :888–905, 2000.

[16] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Video content modeling with latent semantic analysis. In *Third International Workshop on Content-Based Multimedia Indexing*, 2003.

[17] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Latent semantic analysis for an effective region-based video shot retrieval system. In *Proceedings of the ACM International Workshop on Multimedia Information Retrieval*, 2004.

[18] Rong Zhao and William I Grosky. From features to semantics : Some preliminary results. In *Proceedings of the International Conference on Multimedia and Expo*, 2000.

Chapitre 3

Analyse du contenu sémantique des plans vidéo

La recherche par le contenu visuel a rapidement atteint ses limites. En effet l'utilisateur est intéressé essentiellement par le contenu sémantique qui regroupe des documents pouvant être visuellement très différents. Les moteurs de recherche futurs devront répondre à cette exigence des utilisateurs et avoir la capacité de proposer une grande diversité de documents dont le contenu sémantique est proche. Pour cela, la description manuelle du contenu est une approche initiale. Comme nous pouvons l'imaginer, cette tâche est longue et fastidieuse. Des systèmes automatiques sont donc fortement attendus pour résoudre et rendre possible l'analyse du contenu sémantique. Google a mis en place un moteur de recherche d'images par mots clefs. « Pour déterminer le contenu graphique d'une image, Google analyse le texte de la page qui entoure l'image, le titre de l'image et de nombreux autres critères. Google applique également des algorithmes performants pour éliminer les doublons (images identiques) et pour garantir que les résultats portent sur les images de la plus haute qualité possible. » [1] *Le monde de la recherche est actuellement très actif dans ce domaine. Notamment NIST (« National Institute of Standards and Technology ») a débuté en 2001 un vaste projet international d'évaluation des méthodes de recherche d'information audiovisuelle sur de grandes quantités de vidéos (TRECVID [11]). L'idée générale est de construire des modèles qui vont utiliser les similarités visuelles pour estimer le contenu sémantique. Ces estimations vont ensuite permettre d'effectuer des recherches au niveau sémantique et de retrouver des contenus visuellement différents tout en ayant le même contenu. Par exemple un coucher du soleil peut contenir une palette de couleur très différente selon le lieu, l'heure et le climat. En utilisant une recherche uniquement sur les caractéristiques brutes de l'image, il sera impossible de retrouver certaines images. Par contre, un modèle statistique est parfaitement apte à déterminer une catégorie sous différentes conditions.*

Dans ce chapitre nous proposons d'appliquer différents modèles pour estimer le contenu sémantique de plans vidéo. Pour cela, nous travaillons sur l'ensemble des données fournies par NIST pour TRECVID. Nous avons porté notre attention sur des systèmes généraux dont l'entraînement doit permettre de s'adapter à une grande variété de contenus et de classes. Les systèmes de classification sont regroupés principalement en trois classes : par mémoire, étude des frontières, et

[1] Principe de base décrit sur le site français de Google.

probabilistes. Nous avons sélectionné un système représentant chaque catégorie.

3.1 La base de données TRECVID

Nos travaux sont réalisés sur l'ensemble des vidéos fourni par TRECVID en 2003. A l'occasion de ce partage international d'une relativement grande masse de vidéos, un effort commun d'annotation a été fourni par l'ensemble des participants à l'aide de l'outil de Lin et al. [6]. Il consistait à décrire les plans vidéo par des mot choisis dans un vocabulaire prédéfini. Au total le vocabulaire comptabilise 133 concepts ou classes regroupés dans trois catégories : évènement, scène et objets. L'application utilisée pour cette tâche permettait également de sélectionner les régions correspondant aux mots clefs attribués, mais nous n'avons pas eu l'occasion d'utiliser cette information. Deux avantages majeurs sont à retenir de ce partage des annotations. Tout d'abord elle permet d'obtenir une grande quantité de vidéos annotées. Ensuite elle permet de comparer objectivement les différents systèmes proposés puisqu'ils sont entraînés sur la même référence. Cette comparaison n'est pas effectuée dans ce mémoire puisque nous avons préféré effectuer une évaluation identique des systèmes présentés dans ce chapitre et le prochain. Notons que nos systèmes ont obtenu des performances moyennes aux trois dernières évaluations (Souvannavong et al. [8, 9, 10]) sachant que nous avons proposé des méthodes génériques à chaque fois.

Au total, la base de données regroupe 60 heures de nouvelles américaines (canaux de CNN et ABC) avec leurs annotations. L'unité de traitement et d'évaluation est le plan. Le découpage des vidéos en plan est fourni par TRECVID ainsi que leur images clefs. Pour cela le système proposé par Quénot [7] est utilisé. Dans le cadre particulier de notre étude de différents systèmes, nous avons choisi de répartir les 60 heures annotées en trois ensembles de taille égale qui seront désignés par l'ensemble d'entraînement, l'ensemble de validation et l'ensemble de test. Comme nous le verrons, l'ensemble d'entraînement est utilisé pour créer les modèles de classification. L'ensemble de validation est utilisé pour sélectionner les meilleurs paramètres pour ces modèles. Et l'ensemble de test permet d'effectuer l'évaluation finale. Il est important de noter que nous avons respecté l'ordre chronologique des nouvelles lors du découpage afin de ne pas introduire de biais. Si l'ordre des plans avait subit un ensemble de permutation aléatoire, alors des plans très similaires se trouveraient dans les trois ensembles. Le problème serait alors simplifié mais il ne correspondrait pas à la réalité.

Parmi tous les concepts du vocabulaire, nous en avons retenus une vingtaine pour notre étude : studio, présentateur du journal, présentateur masculin, présentateur féminin, Bill Clinton, basket-ball, hockey sur glace, nature et végétation, fleur, arbre, forêt, verdure, désert, montagne, plage, route, gens, avion, incrustation de texte, bâtiment, paysage urbain, autre. La classe autre est active lorsque aucun des concepts choisis n'est présent. Nous avons sélectionné des concepts aux propriétés différentes pour pouvoir étudier avec précision les modèles de classification. Sur 46 600 plans, les concepts apparaissent de 123 à 19 600 fois comme nous pouvons le voir dans le tableau récapitulatif 3.1. Ils décrivent d'une part des concepts présents au niveau de la scène comme studio et basket-ball ; et d'autre part des concepts plus orientés vers les objets composants la scène comme route et avion.

L'objectif que nous fixons est de construire des modèles permettant d'estimer les concepts choisis. Nous avons choisi d'utiliser des méthodes génériques qui sont présentées dans les sections suivantes.

Id	Concepts	test	entraînement	validation	total
1	studio	961 (6.1%)	1113	916	2990
2	présentateur du journal	859 (5.5%)	881	868	2608
3	-présentateur masculin	465 (2.9%)	423	434	1322
4	-présentateur féminin	413 (2.6%)	479	442	1334
5	Bill Clinton	107 (0.7%)	68	143	318
6	basketball	144 (0.9%)	244	235	623
7	hockey sur glace	121 (0.7%)	45	34	200
8	nature et végétation	957 (6.1%)	849	1281	3087
9	-fleur	35 (0.2%)	101	128	264
10	-arbre	354 (2.3%)	233	417	1004
11	-forêt	119 (0.8%)	102	109	330
12	-verdure	480 (3%)	328	745	1553
13	désert	65 (0.4%)	18	40	123
14	montagne	107 (0.7%)	132	122	361
15	plage	81 (0.5%)	83	65	229
16	route	369 (2.4%)	377	373	1119
17	gens	1768 (11.3%)	1836	2185	5789
18	avion	135 (0.9%)	89	72	296
19	incrustation de texte	6903 (44.4%)	6532	6188	19623
20	bâtiment	616 (3.9%)	592	819	2027
21	paysage urbain	195 (1.3%)	211	192	598
22	autre	5709 (36.7%)	5925	5479	17113
	total	15531	15531	15533	46595

TAB. 3.1 – Répartition des plans vidéo dans les différents ensembles par concept sémantique. *Le quantité relative de chaque classe de l'ensemble de test dans l'ensemble de test est précisée pour donner une idée de la borne inférieure des performances à obtenir (précision moyenne avec un ordonnancement aléatoire).*

3.2 Classification

Cette première section présente le problème de la classification. La première partie décrit la tâche de la classification puis elle intoduit le cadre théorique. La seconde partie présente les vecteurs que nous allons classer, et introduit la notion de fusion. Finalement cette section se termine par la présentation de la méthode d'évaluation des systèmes de classification dans le cadre de la recherche d'information.

3.2.1 Objectif

La classification est le procédé qui permet d'estimer la ou les classes auxquelles un élément donné appartient. Elle peut être menée de manière aveugle ou supervisée. Les méthodes aveugles permettent de déterminer automatiquement des classes au sein d'un ensemble puis de classer les éléments dans ces classes. L'algorithme de k-means est un exemple de système de classification en aveugle. Les méthodes dites supervisées nécessitent un ensemble d'entraînement qui est constitué de couples formés par un élément et ses classes associées. L'entraînement permet alors de calculer les paramètres des modèles sélectionnés (mélange de Gaussiennes, machine à vecteurs de support, ...) afin de minimiser les erreurs de classification. Le modèle entraîné permet ensuite de classer de nouveaux points (3.1). La qualité des modèles dépend alors de leur capacité de généralisation, c'est à dire la capacité de classer correctement de nouveaux éléments qui sont différents de ceux présents dans l'ensemble d'entraînement. Toutefois l'estimation cette faculté de généralisation est délicate et souvent un ensemble de validation est utilisé pour valider le modèle. Ce chapitre se place dans le cadre d'une classification supervisée.

Plus formellement, définissons par $\mathcal{E}$ l'espace des éléments ou des échantillons x_i à classer et par C l'ensemble des classes possibles. Dans de nombreux cas, la convention consiste à poser $C = [-1, 1]^{n_c}$. Une classe est présente lorsque la dimension correspondante est positive et absente lorsqu'elle est négative. C'est la convention que nous adopterons au cours de notre étude. L'ensemble d'entraînement est ensuite défini par $\mathcal{L} = \{(x_i, c_i) \in \mathcal{E} \times C, i \in [1..n_{\mathcal{L}}]\}$. En théorie la fonction de classification est définie par une hypothèse $\mathcal{H}_{\mathcal{L}} : \mathcal{E} \rightarrow \mathfrak{R}^{n_c}$ qui minimise l'erreur moyenne généralisée :

$$e_{\mathcal{L}} = \int_{\mathcal{E}} E_{C|X}[C(\mathcal{H}_{\mathcal{L}}(x), c)]p(x)dx \tag{3.1}$$

Dans cette expression intervient les termes $p(x)$ et $C : \mathfrak{R}^{n_c} \times C \rightarrow \mathfrak{R}^+$ qui sont respectivement la den-

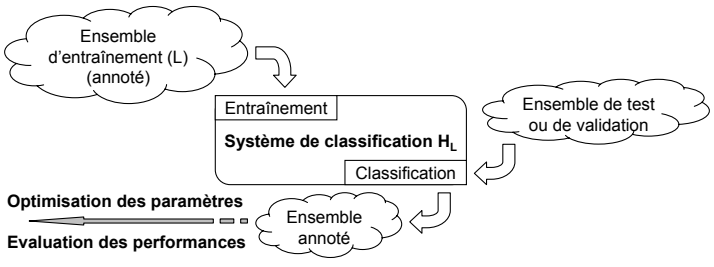


FIG. 3.1 – Principe de la classification supervisée.

sité de probabilité de X et la fonction de coût des erreurs de classification. Souvent les algorithmes d'entraînement restreindront la minimisation sur $\mathcal{L}$ en supposant que la distribution des éléments est représentative de $\mathcal{E}$. Un ensemble de test est alors utilisé pour estimer les capacités de généralisation des hypothèses construites. La capacité de généralisation est un aspect important des hypothèses. Une bonne généralisation permet de garantir une estimation correcte lorsque les échantillons présentés diffèrent de ceux présents dans $\mathcal{L}$. Dans le problème de classification qui nous intéresse nous supposons que les classes sont indépendantes et les hypothèses sont construites par classe. Cette logique est la plus courante puisque la plupart des systèmes de classification sont élaborés pour résoudre des problèmes binaires. Par ailleurs Allwein et al. [1] ont montré qu'il était possible de transformer un problème de classification multi classes en plusieurs problèmes de classification binaire. Ils proposent entre autre la méthode un-contre-tous (« one-against-all ») qui consiste à construire un système de classification binaire par classe. Chaque système binaire classifie les échantillons dans une classe ou dans une autre qui comprend toutes les classes restantes. Cette méthode est adoptée dans le reste du chapitre.

3.2.2 Un aperçu de la fusion

La classification est réalisée sur les caractéristiques de couleur et de texture des plans vidéo. L'analyse de la sémantique latente introduite dans le chapitre précédent est employée pour caractériser efficacement le contenu. Au total trois images clefs sont sélectionnées par plan. Nous adoptons la technique présentée dans le chapitre précédent pour combiner l'information issue des ces trois images. L'objectif est de pouvoir capturer le contenu de plans variés ou incorrectement segmentés dans le temps. Pour cela l'image clef fournie par TRECVID est traitée ainsi que les images centrées dans les deux intervalles voisins : entre le début du plan et l'image clef et entre l'image clef et la fin du plan. Finalement nous laisserons les systèmes de classification sélectionner la dureté de la LSA en utilisant une recherche exhaustive du meilleure pourcentage de composantes à conserver. Le choix se fera en fonction des performances réalisées sur un ensemble de validation.

Les modèles sont construits par classe et par caractéristique visuelle (couleur et texture). Chaque modèle est alors évalué indépendamment puis les hypothèses de couleur et de texture sont combinées pour profiter des propriétés des deux caractéristiques visuelles. Comme nous le verrons dans le prochain chapitre, la fusion est un problème délicat que nous abordons simplement dans les expériences suivantes. Les scores de classification fourni par chaque système doivent être fusionnés afin d'obtenir une valeur unique de détection d'une classe. Les scores fournis par les hypothèses des deux caractéristiques sont soit additionnés, soit leur maximum est conservé. Nous proposons également de réaliser la fusion en amont de la classification. Pour cela, les vecteurs de couleur et de texture sont simplement concaténés pour former un unique vecteur. La classification est alors réalisée sur ces vecteurs hybrides.

3.2.3 Evaluation des systèmes de classification

L'évaluation des systèmes de classification s'effectue généralement dans un contexte de détection ou de reconnaissance. Les critères permettant d'évaluer la qualité des estimations sont alors le taux d'erreur, la sensibilité et la spécificité. Le taux d'erreur est défini par le pourcentage d'erreurs réalisées sur un ensemble. La sensibilité et la spécificité permettent d'affiner l'évaluation en

quantifiant le taux de vrais positifs et le taux de vrais négatifs.

Cependant nous plaçons notre étude dans le contexte de la recherche d'information et les mesures appropriées sont donc mises en oeuvre. Ainsi nous affranchissons les systèmes de classification d'établir une décision binaire sur l'appartenance à une classe comme nous le préciserons dans chaque cas. Un score d'appartenance à une classe suffit pour ordonner les plans et calculer les mesures de précision et de rappel. La précision moyenne sur 2 000 plans est retenue comme mesure d'évaluation afin de faciliter les comparaisons entre les différents systèmes. Notons que le 24^{e} concepts qui apparaît sur l'ensemble des tracés de ce chapitre, correspond en fait à la performance moyenne sur l'ensemble des classes.

3.3 Modèles utilisés

Les systèmes de classification sont regroupés principalement en trois catégories, les systèmes probabilistes, les systèmes à mémoire et les systèmes d'études des frontières. Ces catégories sont complémentaires comme l'indique leur dénomination, et pour cette raison un algorithme de chacune d'entre elles est étudié. Les algorithmes les plus usités sont choisis, tout d'abord les mélanges de gaussiennes pour représenter la première catégorie, la classification par les k plus proches voisins pour représenter la deuxième catégorie, et finalement les machines à vecteurs de support pour représenter la dernière catégorie.

3.3.1 Mélange de Gaussiennes

Principe

Le premier modèle de classification que nous proposons utilise les mélanges de Gaussiennes pour modéliser la distribution des classes. Un mélange de Gaussiennes m_G est une fonction de densité de probabilité définie par la somme pondérée de plusieurs densités de probabilité qui suivent une loi Gaussienne. Elle permet de prendre en compte la diversité des classes et la présence de sous-classes par l'intermédiaire des mélanges et la variabilité par l'intermédiaire des densités de probabilité Gaussiennes comme le montre la figure 3.2.

$$m_G(x) = \sum_{i=1}^{n_G} \alpha_i \mathcal{N}(\mu_i, \Sigma_i)(x) \tag{3.2}$$

$$\text{avec } \mathcal{N}(\mu_i, \Sigma_i)(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right) \tag{3.3}$$

Les paramètres α_i, μ_i et Σ_i sont respectivement les probabilités a priori, les moyennes et les matrices de co-variances des distributions Gaussiennes. Ils sont souvent estimés en utilisant l'algorithme itératif EM en alternant le calcul de l'espérance (étape E) et la maximisation de la vraisemblance (étape M) (Dempster et al. [4]). Dans le cas d'un mélange de Gaussiennes, il est possible de montrer (Bilmes [2]) que les étapes E et M sont conjointement réalisables par les formules suivantes de

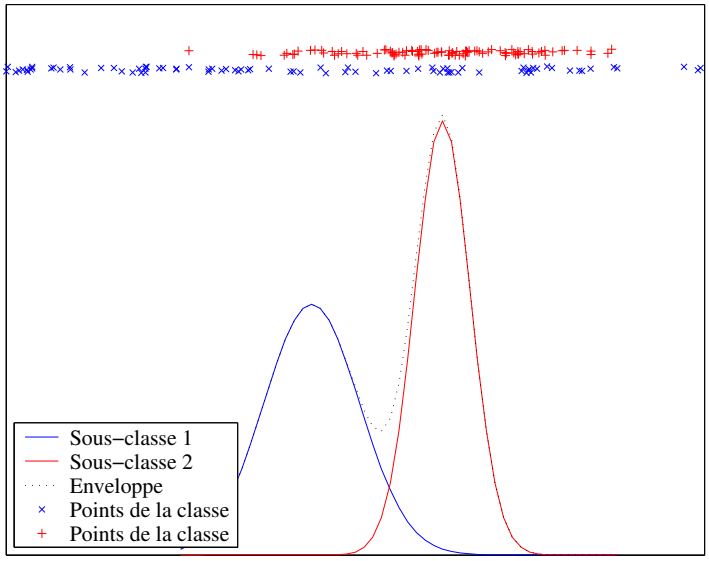


FIG. 3.2 – Modélisation d'une classe par un mélange de gaussiennes à deux composantes. *Illustration en une dimension. Les points en haut de la figure sont les points de la classe à modéliser. La courbe en pointillés représente la densité de probabilité estimée. Les deux autres courbes sont les composantes gaussiennes de cette estimation. L'ordanncement des plans est effectué en fonction de la valeur de la densité de probabilité au point correspondant.*

mise à jour des paramètres :

$$\alpha'_i = \frac{1}{n_{\mathcal{L}}} \sum_{k=1}^{n_{\mathcal{L}}} p(i|x_k, \mu_k, \Sigma_k) \tag{3.4}$$

$$\mu'_i = \frac{\sum_{k=1}^{n_{\mathcal{L}}} x_k p(i|x_k, \mu_k, \Sigma_k)}{\sum_{k=1}^{n_{\mathcal{L}}} p(i|x_k, \mu_k, \Sigma_k)} \tag{3.5}$$

$$\Sigma'_i = \frac{\sum_{k=1}^{n_{\mathcal{L}}} (x_k - \mu'_i)(x_k - \mu'_i)^T p(i|x_k, \mu_k, \Sigma_k)}{\sum_{k=1}^{n_{\mathcal{L}}} p(i|x_k, \mu_k, \Sigma_k)} \tag{3.6}$$

$$\text{avec } p(i|x_k, \mu_k, \Sigma_k) = \frac{\alpha_i \mathcal{N}(\mu_i, \Sigma_i)(x_k)}{\sum_{j=1}^{n_G} \alpha_j \mathcal{N}(\mu_j, \Sigma_j)(x_k)} \tag{3.7}$$

L'estimation des paramètres présentée est un cas particulier du cadre général présenté en première partie 3.2.1. En effet l'optimisation est effectuée uniquement sur les échantillons positifs, la fonction de coût est la fonction produit et l'hypothèse de classification est la densité de probabilité d'une classe i.

$$\mathcal{H}_{G_i} = m_{G_i}(x) \tag{3.8}$$

Cette hypothèse a la particularité de ne pas avoir de seuil défini par défaut permettant de conclure à l'appartenance à une classe. Toutefois la définition d'un seuil n'est pas le choix le plus pertinent

puisque l'hypothèse est construite uniquement sur les échantillons positifs. Il est judicieux d'utiliser le critère de maximum a posteriori (MAP) pour sélectionner une classe parmi les autres. Il est défini comme suit :

$$MAP(x) = arg\max_{c_i} p(c_i|x) \tag{3.9}$$

$$\text{avec } p(c_i|x) = \frac{p(x,c_i)}{p(x)} \tag{3.10}$$

$$\text{soit } p(c_i|x) = \frac{p(x|c_i)p(c|i)}{p(x)} \tag{3.11}$$

p(x) est une constante et nous supposons que toutes les classes sont équiprobables. Le critère se met alors sous la forme :

$$MAP(x) = arg\max_{c_i} p(x|c_i) = arg\max_{c_i} m_G(x) \tag{3.12}$$

Bien que cette décision ne soit pas nécessaire dans le cadre qui a été spécifié auparavant, elle permet de définir une nouvelle hypothèse qui prend en compte uniquement le mélange le plus adéquat en respectant la logique d'une classification par les GMM.

$$\mathcal{H}'_{G_i} = \begin{cases} m_{G_i}(x) & \text{si } i = MAP(x) \\ -\infty & \text{sinon} \end{cases} \tag{3.13}$$

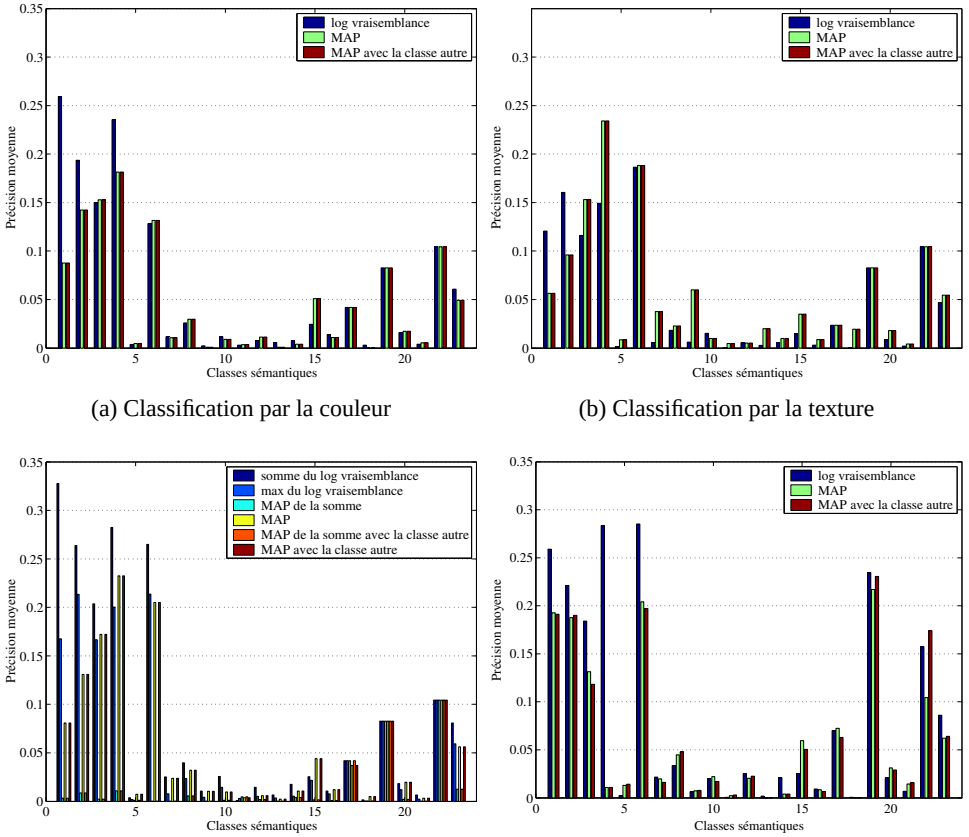
Mise en oeuvre

Les mélanges de Gaussiennes sont utilisées pour modéliser les classes sémantiques. Les matrices de co-variance sont supposées diagonales. Cette simplification du modèle permet d'obtenir un entraînement plus rapide qui produit une hypothèse plus générale. De plus cette simplification évite la dégénérescence de la matrice de covariance. Deux paramètres ne sont pas estimés durant l'algorithme précédemment proposé. Le premier est la dureté de la LSA qui peut être ajustée afin de favoriser la détection d'information implicite. Le second est le nombre de Gaussiennes qui vont permettre de modéliser au mieux la distribution des classes en détectant automatiquement les sous-ensembles. Un ensemble de validation est utilisé pour effectuer les bons choix selon un critère défini au préalable. Plusieurs critères sont à notre disposition. Tout d'abord les critères statistiques : le critère bayesien d'information (« the Bayesian Information Criterion »), la vraisemblance intégrée et complète (« the Integrated Completed Likelihood ») et le critère d'entropie normalisé (« the Normalized Entropy Criterion ») ; puis les critères propres au problème comme la précision moyenne qui sera retenue dans notre étude.

La figure 3.3 montre l'ensemble des résultats obtenus par la classification à l'aide des mélanges de Gaussiennes. Globalement les performances sont faibles, ce qui est compréhensible étant donné la difficulté du problème. Mais procédons à une étude par étape. Tout d'abord l'ensemble des résultats montre qu'il n'est pas nécessaire d'utiliser le critère du MAP. Bien au contraire la vraisemblance suffit. Le MAP fait intervenir la qualité de la classification de l'ensemble des classes ce qui pénalise fortement les systèmes bien construits. D'autre part la fusion joue un rôle primordial dans la classification et elle semble plus efficace en aval des systèmes de classification. Le tableau 3.2 récapitule les valeurs du nombre de composantes conservées par la LSA. Nous constatons qu'uniquement 10% des composantes sont conservées par la plus grande partie des systèmes de classification.

Id	Concept	couleur	texture	couleur,texture
1	studio	10	10	10
2	présentateur du journal	10	10	10
3	-présentateur masculin	10	10	10
4	-présentateur féminin	10	10	20
5	Bill Clinton	10	10	10
6	basketball	10	10	10
7	hockey sur glace	10	10	10
8	nature et végétation	10	10	10
9	-fleur	10	10	10
10	-arbre	10	10	10
11	-forêt	10	40	10
12	-verdure	10	10	10
13	désert	10	10	10
14	montagne	30	10	20
15	plage	10	10	10
16	route	10	10	10
17	gens	10	10	10
18	avion	10	10	20
19	incrustation de texte	10	10	10
20	bâtiment	10	10	10
21	paysage urbain	10	10	10
22	autre	10	10	10

TAB. 3.2 – Pourcentage de composantes conservées par concept sémantique. Classification par les mélanges de gaussiennes.



(a) Classification par la couleur

(b) Classification par la texture

(c) Classification indépendante par la couleur et la texture, puis fusion

(d) Classification conjointe par la couleur et la texture

FIG. 3.3 – Classification par les mélanges de Gaussiennes

3.3.2 Classification par mémoire

Principe

Cette catégorie de système de classification repose sur l'utilisation d'une mémoire afin d'estimer les classes des nouveaux échantillons. La mémoire est en faite composée d'exemples, c'est à dire d'éléments dont les classes sont connues. Les classes d'un nouvel échantillon sont alors obtenues par héritage des classes des exemples les plus proches comme cela est illustré sur la figure 3.4. La classification utilisant la mémoire à la particularité de ne faire aucune hypothèse sur la distribution des données contrairement aux mélanges de Gaussiennes par exemple ; de plus la mémoire peut facilement être étendue de manière à être constamment à jour et suivre les évolutions des données. Toutefois elle souffre de deux inconvénients majeurs qui sont la complexité des calculs pour trouver les plus proches voisins et son manque de généralisation. Le problème de la généralisation se pose dès lors que les échantillons à classer sont trop éloignés des exemples car la notion de proche voisin perd tout son sens.

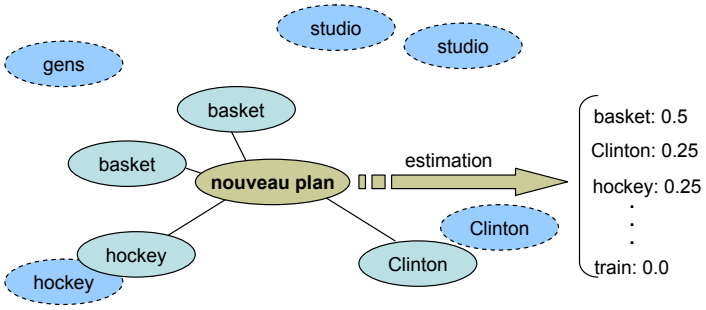


FIG. 3.4 – Classification par les quatre plus proches voisins. *La classification d'un nouvau plan se fait par la recherche de ces quatres plus proches voisins, puis par héritage de leurs classes. Les valeurs obtenues permettent ensuite d'ordonner les plans.*

Désignons par V le voisinage d'un point et par $K_j : \mathcal{E} \times \mathcal{E} \rightarrow \Re$ une fonction noyau attachée à la classe j. D'une manière générale l'hypothèse de classification s'écrit alors sous la forme :

$$\mathcal{H}_{KNN_j}(x) = \frac{\sum_{y_i \in V(x)} c_j K_j(x, y_i)}{\sum_{y_i \in V(x)} K_j(x, y_i)} \tag{3.14}$$

$$ou \quad \mathcal{H}'_{KNN_j}(x) = \sum_{y_i \in V(x)} c_j K_j(x, y_i) \tag{3.15}$$

L'absence de normalisation dans la seconde formulation est particulièrement intéressante dans notre situation puisque les éléments sont ordonnés en fonction de la valeur de l'hypothèse. Ainsi les éléments à classer trop distants de la mémoire sont pénalisés. La section suivante présente l'impact de cette normalisation et étudie les performances des systèmes à mémoire.

Mise en oeuvre

Dans les expériences, le noyau est défini par la similarité du plan avec son voisin. Il se met sous la forme :

$$K_j(x_i, y_i) = cos(x_i, y_i) \tag{3.16}$$

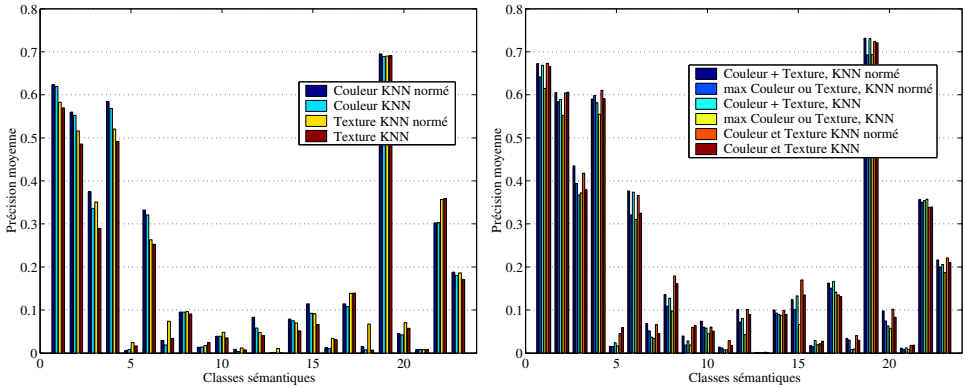
$$\tag{3.17}$$

Ainsi l'hypothèse de classification effectue une somme pondérée des classes des voisins, avec une pondération qui dépend de la similarité entre le plan et son voisin correspondant.

Afin de tirer profit du système de classification, les noyaux sont adaptés pour effectuer une mesure de similarité sur un nombre de composantes optimal par classe. La valeur de la précision moyenne sur l'ensemble de validation est alors utilisé pour sélectionner la meilleure taille du voisinage par classe. Par ailleurs nous profitons de cette validation croisée pour conjointement optimiser la taille du voisinage. La figure 3.5 montre une nette amélioration des performances avec cette méthode de classification. Même si les différences ne sont pas flagrantes, l'approche normée est légèrement plus performante. Ceux sont toujours les mêmes classes dont les performances prédominent. La fusion des données révèle de nouveau son importance tout en étant plus discrète. Le

Id	Concept	couleur	texture	couleur,texture
1	studio	40 - 40	30 - 10	30 - 30
2	présentateur du journal	40 - 10	40 - 20	40 - 30
3	-présentateur masculin	40 - 20	35 - 10	40 - 20
4	-présentateur féminin	40 - 100	40 - 10	40 - 30
5	Bill Clinton	35 - 10	30 - 70	5 - 30
6	basketball	35 - 20	25 - 10	40 - 20
7	hockey sur glace	20 - 30	35 - 20	40 - 10
8	nature et végétation	30 - 10	5 - 10	35 - 10
9	-fleur	40 - 10	20 - 20	40 - 50
10	-arbre	10 - 20	10 - 10	40 - 10
11	-forêt	5 - 70	40 - 30	25 - 20
12	-verdure	40 - 40	20 - 90	40 - 20
13	désert	5 - 100	15 - 100	15 - 80
14	montagne	5- 60	5 - 40	5 - 30
15	plage	40 - 20	5 - 70	15 - 40
16	route	5 - 30	40 - 30	40 - 10
17	gens	40 - 60	35 - 10	35 - 60
18	avion	5 - 60	30 - 20	5 - 40
19	incrustation de texte	40 - 10	40 - 10	40 - 10
20	bâtiment	40 - 10	15 - 20	40 - 50
21	paysage urbain	5 - 10	5 - 20	5 - 20
22	autre	40 - 30	35 - 30	40 - 10

TAB. 3.3 – Taille du voisinage suivi du pourcentage de composantes conservées par concept sémantique. Classification normée par les k plus proches voisins.



(a) Classification individuelle (b) Classification combinée

FIG. 3.5 – Classification par les k-plus proches voisins.

tableau 3.3 montre que les voisinages nécessaires sont souvent grand et le nombre de composantes conservées est de l'ordre de 30% en moyenne.

3.3.3 Machine à vecteurs de support

Principe

Les machines à vecteurs de support permettent d'étendre la notion de neurone à des configurations plus complexes des données. Notamment elles permettent la classification de classes non linéairement séparables et bruitées. Introduites par Vapnik [12] en 1995 pour la reconnaissance de texte, elles connaissent actuellement un essor important. L'idée de base est celle d'un neurone, c'est à dire la recherche d'un hyperplan paramétré par (w, b) qui sépare deux classes linéairement :

$$c_i(w.x_i + b) \geq 1 \tag{3.18}$$

Afin de maximiser la marge (c'est à dire la plus petite distance entre l'hyperplan et les points de l'entraînement), il est nécessaire de minimiser $\|w\|$. L'hyperplan paramétré par (w, b) est alors la solution du problème suivant :

$$\text{minimiser} \quad \|w\| \tag{3.19}$$

$$\text{avec} \quad c_i(w.x_i + b) \geq 1 \tag{3.20}$$

$$\forall (x_i, c_i) \in \mathcal{L} \tag{3.21}$$

L'utilisation des multiplicateurs de Lagrange permet de montrer que :

$$w_{opt} = \sum_{i=1}^{N_{\mathcal{L}}} \alpha_i c_i x_i \tag{3.22}$$

$$\sum_{i=1}^{N_{\mathcal{L}}} \alpha_i c_i = 0 \tag{3.23}$$

$$\alpha_i \geq 0 \tag{3.24}$$

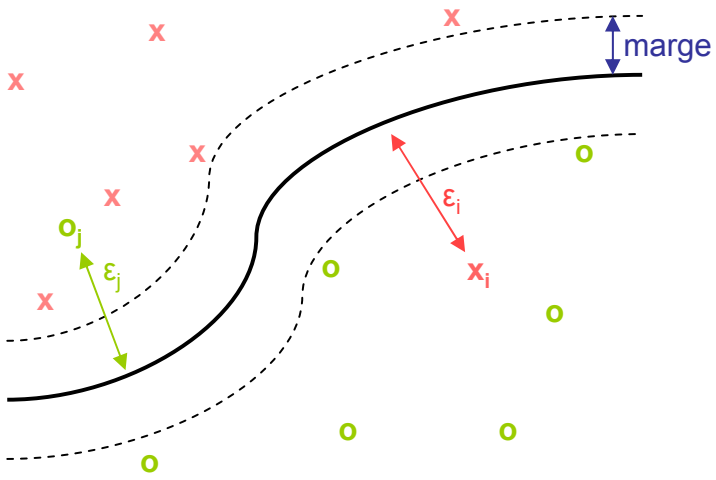


FIG. 3.6 – Machines à vecteurs de support. *L'entraînement consiste à identifier « l'hyperplan » qui sépare au mieux les deux classes en maximisant la marge et en minimisant les erreurs ϵ. Un nouveau plan est classé en fonction de sa distance signée à « l'hyperplan ». Cette dernière permet ensuite de les ordonner.*

L'ensemble des $\alpha_i \neq 0$ définit les vecteurs de support v_i qui sont utilisés pour définir l'hyperplan séparant les deux classes. L'hypothèse construite est :

$$\mathcal{H}_{SVM}(x) = b_{opt} + \sum_{i=1}^{N_v} \alpha_i v_i.x \tag{3.25}$$

Cette étude peut très élégamment être généralisée au cas non linéaire en remplaçant le produit scalaire par une fonction noyau $K(x,y)$ qui respecte les conditions de Mercer (Cristianini and Shawe-Taylor [3]). Les conditions de Mercer garantissent qu'il existe un espace $\mathcal{F}$ et une fonction $\Phi : \mathcal{E} \rightarrow \mathcal{F}$ tel que $K(x,y) = \Phi(x).\Phi(y)$. Cette formulation est alors équivalente à rechercher une solution dans l'espace transformé $\mathcal{F}$. Le fait de transformer les données est connu dans le domaine des systèmes intelligents pour permettre d'améliorer les conditions d'apprentissage et les performances. Toutefois, il est très délicat de définir la transformation et l'espace transformé qui peut être de très grande dimension. L'utilisation des noyaux est un moyen efficace de définir de nombreuses fonctions sans explicitement définir la transformée. Les plus communément utilisés actuellement sont les noyaux polynomiaux et gaussiens.

Pour permettre de calculer une hypothèse dans le cas où les données sont bruitées et qu'il n'est pas possible de trouver un noyau permettant de parfaitement séparer les classes, une formulation plus générale est proposée. L'hyperplan doit satisfaire la condition suivante :

$$c_i(w.x_i + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0 \tag{3.26}$$

Dans ce cas ε_i mesure à quel point un échantillon est mal classifié et $\sum \varepsilon_i$ est une borne supérieure

des erreurs de classification. L'hyperplan est alors la solution du problème suivant :

$$\text{minimiser} \quad \|w\| + C \sum_{i=1}^{N_{\mathcal{L}}} \varepsilon_i^2 \tag{3.27}$$

$$\text{avec} \quad c_i(w.x_i + b) \geq 1 - \varepsilon_i \tag{3.28}$$

$$\forall (x_i, c_i) \in \mathcal{L} \tag{3.29}$$

Cette formulation permet de tolérer les erreurs de classification et d'éviter ainsi le sur entraînement tout en favorisant une meilleure généralisation. Le terme C est alors souvent déterminé en utilisant un ensemble de validation.

Mise en oeuvre

Nous avons utilisé la librairie SVMLight (Joachims [5]) avec un noyau gaussien pour réaliser nos expériences. Ce noyau possède un paramètre de lissage qui doit être prédéfini. L'ensemble de validation est alors utilisé pour estimer au mieux les trois paramètres intervenant dans le modèle, c'est à dire le lissage, le nombre de composantes et le coût des erreurs. Après une recherche exhaustive des meilleurs paramètres, les valeurs de lissage et le coût des erreurs obtenues sont très proches d'un problème de classification à l'autre.

La figure 3.7 résume les résultats fournis par l'ensemble des expériences. Une amélioration sensible est observée en particulier en utilisant la somme comme procédé de fusion. Le tableau 3.4

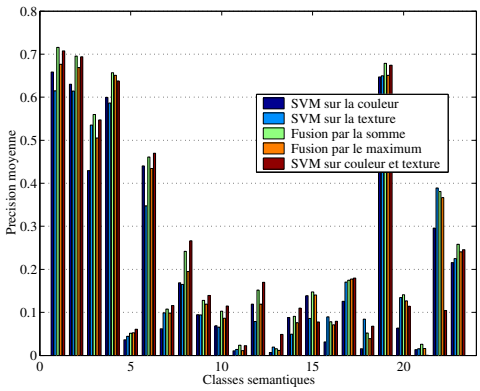


FIG. 3.7 – Classification par les machines à vecteurs de support.

montre que les propriétés de l'analyse de la sémantique latente sont toujours mises à profit pour tirer le meilleur parti de cette représentation.

3.4 Analyse du problème de classification

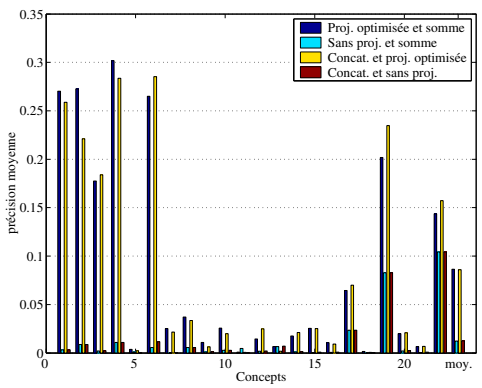
Cette partie apporte une analyse du problème de la classification au vu des résultats fournis dans la section précédente. Tout d'abord, l'intérêt de la LSA est étudié puis le problème de la classification des plans de nouvelles télévisées.

Id	Concept	couleur	texture	couleur,texture
1	studio	40	90	20
2	présentateur du journal	20	100	10
3	-présentateur masculin	20	90	30
4	-présentateur féminin	30	10	20
5	Bill Clinton	10	10	10
6	basketball	70	20	20
7	hockey sur glace	20	20	20
8	nature et végétation	50	20	10
9	-fleur	80	10	80
10	-arbre	30	20	30
11	-forêt	10	50	20
12	-verdure	20	100	50
13	désert	10	40	40
14	montagne	20	80	50
15	plage	10	90	90
16	route	30	30	30
17	gens	40	20	30
18	avion	10	20	30
19	incrustation de texte	100	30	30
20	bâtiment	30	20	100
21	paysage urbain	20	20	10
22	autre	90	60	70

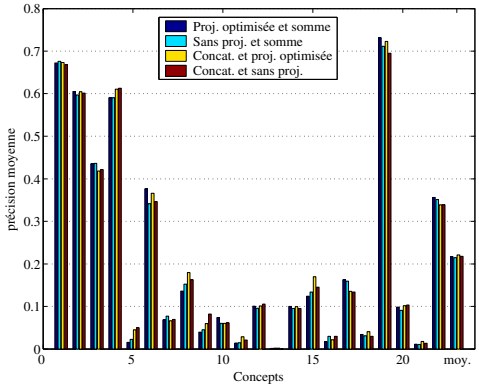
TAB. 3.4 – Pourcentage de composantes conservées par concept sémantique. Classification par les SVM.

3.4.1 Intérêts de la LSA

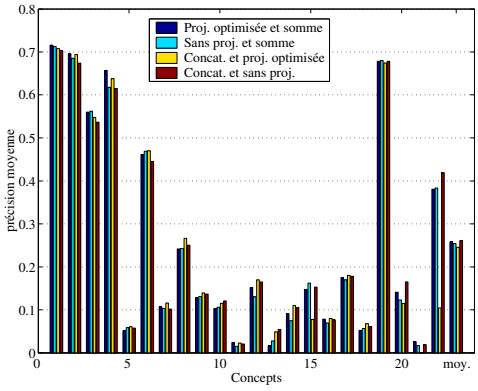
Cette section traite de l'intérêt de l'analyse de la sémantique latente dans le cadre de l'estimation du contenu sémantique des plans. Nous avons vu dans la partie précédente que la LSA était relativement dure en moyenne puisque plus de la moitiée des composantes sont supprimées dans pratiquement tous les cas. Toutefois nous n'avons pas comparé les résultats obtenus avec ou sans projection. Nous allons remédier à cela et montrer tous les bénéfices apportés par une représentation des régions d'un plan par la LSA.



(a) Classification par les mélanges de gaussiennes.



(b) Classification par les k plus proches voisins.



(c) Classification par les machines à vecteurs de support.

FIG. 3.8 – Impact de l'optimisation de la dureté de la LSA.

Les figures 3.8(a), 3.8(b) et 3.8(c) montrent la précision moyenne obtenue par concepts pour plusieurs configurations des systèmes de classification étudiés. Nous trouvons tout d'abord les performances fournies lorsque la fusion est effectuée par la somme puis lorsque la fusion est effectuée avant la classification par concaténation des vecteurs de couleur et de texture. Dans chaque cas, les performances sont tout d'abord fournies lorsque la dureté de la LSA est optimisée puis losrqu'elle est nulle ; c'est à dire que toutes les composantes sont conservées et donc que la LSA n'est pas

effective. La différence de performances est flagrante lors d'une classification par les GMM. Le grand nombre de dimensions à analyser et la faible quantité des données d'entraînement sont les causes de cette différence comme le montre les classes 19 et 22. Ces deux classes sont disponibles en grande quantité et leur performance sans LSA est bien meilleure comparée aux autres classes. La réduction de la dimension des vecteurs est donc particulièrement importante puisqu'elle permet de construire de meilleurs modèles. Par contre il est difficile de savoir si la LSA participe également à l'amélioration des performances comme c'était le cas pour la recherche d'information. En effet, les classifications par les KNN et les SVM donnent une vue différente. En moyenne une légère amélioration est obtenue lorsque la taille de la projection est optimisée. Toutefois cette amélioration qui n'est pas générale à toutes les classes et qui reste faible ne permet pas de justifier l'emploi de la LSA. Par contre la réduction de la dimension qui en découle est un très fort atout.

3.4.2 La classification de plans

La classification de plans de nouvelles télévisées est un problème particulièrement délicat. Tout d'abord les caractéristiques extraites doivent correctement représenter le contenu et sa sémantique. Une grande difficulté est de capturer à la fois le contenu global, le contenu local et plus précisement les objets. Ensuite les systèmes de classification doivent faire face à une grande diversitée des contenus au sein des classes et à un besoin de généralisation très marqué, tout en ayant des données d'entraînement en faibles quantités.

Nous avons proposé d'utiliser l'analyse de la sémantique latente pour décrire le contenu des plans dans leur intégralité, c'est à dire au niveau de l'image et au niveau de la région. Nous avons concentré notre étude sur des systèmes génériques et n'avons pas abordé la description des objets qui nécessite souvent une connaissance a priori. Trois systèmes de classification aux caractéristiques complémentaires ont été utilisés pour conduire la classification sémantique. Au vu des résulats présentés dans la section précédente nous observons une étroite corrélation entre la quantité des données d'entraînement et les performances de la classification. Toutefois cette corrélation est plus complexe puisque d'autres paramètres interviennent. Le premier paramètre est la notion d'objet. Les classes qui sont exprimées par un élément visuel local sont plus difficile à détecter. En effet notre signature décrit l'ensemble de la scène et le système de classification doit faire la part des choses. Mais pourquoi les classes 2, 3 et 4 qui correspondent à des présentateur de journal masculin et/ou féminin sont si bien détectées ? L'explication est fournie par le contexte où ses personnes apparaissent. Les chaînes CNN et ABC ont des fonds différents. Or la majorité des présentateurs de la chaînes CNN sont féminins et inversement sur la chaîne ABC ils sont majoritairement masculins. La classe 19 qui correspond au texte est également bien détectée, ce qui est expliqué par sa fréquence dans les journaux télévisés. Les autres classes qui font intervenir la notion d'objet sont difficilement classées. Outre les contextes très hétérogènes qui accompagnent ces notions, les données ont également des défauts. D'un côté, l'annotation des classes sémantiques orientées objets est délicate et particulièrement subjective comme nous avons pu nous en rendre compte lors de l'annotation des vidéos pour TRECVID. L'annexe B à la page 117 montre un échantillon des classes étudiées et elle met en valeur les difficultés de la définition d'une classe. Ainsi dans la classe avion comprend aussi bien la vue d'un avion dans son ensemble qu'un gros plan sur le réacteur. La plage est parfois réduite à une zone de sable et les arbres sont simplement des troncs. D'un autre côté, la faible qualité des vidéos accentue la difficultés de la détection des objets.

3.5 Conclusion

Ce chapitre a présenté trois systèmes de classification. Ils ont été choisis pour représenter les trois familles principales de systèmes de classification. Ils ont été évalués sur le problème délicat de la classification sémantique de plans vidéo. Les meilleures performances sont détenues par les machines à vecteurs de support suivi de près par une classification par les k plus proches voisins. Les mélanges de gaussiennes ne semblent pas adaptées au vu des résultats obtenus. Nous avons conclu sur un résumé du problème de la classification de nouvelles télévisées et avons apporté des éléments de réponse qui expliquent la diversité des résultats et leurs origines.

Nous avons observé que la classification est grandement améliorée par la fusion de plusieurs caractéristiques. Fort de cette observation, nous poursuivons dans cette direction et proposons dans le prochain chapitre une méthode de fusion utilisant des opérateurs simples.

Bibliographie

[1] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary : A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1 :113–141, 2000.

[2] Jeff A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. ICSI-TR, 1997.

[3] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines*, chapter Kernel-Induced Feature Spaces. Cambridge University Press, 2000.

[4] P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data using the em algorithm. *Journal of the Royal Society of Statistics*, 39(1) :1–38, 1977.

[5] T. Joachims. *Advances in Kernel Methods - Support Vector Learning*, chapter 11 (Making large-Scale SVM Learning Practical). MIT Press, 1999.

[6] Ching-Yung Lin, Belle L. Tseng, and John R. Smith. Video collaborative annotation forum : Establishing ground-truth labels on large multimedia datasets. In *Proceedings of the TRECVID 2003 Workshop*, 2003.

[7] Georges M. Quénot. Trec-10 shot boundary detection task : CLIPS system description and evaluation. In *The 10th Text REtrieval Conference (TREC)*, 2000.

[8] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Semantic feature extraction using mpeg macro-block classification. In *The 11th Text REtrieval Conference (TREC)*, 2002.

[9] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Latent semantic indexing for video content modeling and analysis. In *The 12th Text REtrieval Conference (TREC)*, 2003.

[10] Fabrice Souvannavong, Bernard Merialdo, and Benoit Huet. Eurecom at trecvid 2004 : Feature extraction task. In *The 13th Text REtrieval Conference (TREC)*, 2004.

[11] TRECVID. Digital video retrieval at NIST. http ://www-nlpir.nist.gov/projects/trecvid/.

[12] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

Chapitre 4

Fusion de systèmes de classification

Dans le chapitre précédent nous avons utilisé les caractéristiques de couleur et de texture pour déterminer le contenu sémantique de plans vidéo. Nous avons vu que la combinaison de ces deux caractéristiques permettait d'accroître les performances de la classification. A présent nous allons poursuivre dans cette direction en étudiant le problème de la fusion et en introduisant de nouvelles caractéristiques.

Les sources d'information qui nous permettent de déduire la présence ou l'absence d'une classe dans un plan vidéo sont immenses. Malheureusement elles se comportent de manières très différentes. Certaines sources sont systématiquement associées à une classe, d'autres ne sont que des indices d'une éventuelle présence ou absence. La fusion doit donc faire la part des choses et accorder aux différentes sources l'importance qu'elles méritent. Selon les concepts les sources pertinentes sont différentes et le système de fusion doit s'adapter à chaque problème. Comme nous avons pu le voir au cours des chapitres précédents, la fusion peut être réalisée à différentes étapes du processus. Dans ce chapitre nous concentrons notre étude sur les systèmes de fusion de systèmes de classification.

La première section traite du problème de la fusion. Dans un premier temps un état de l'art est conduit et nous en profitons pour justifier notre choix d'étudier la fusion de systèmes de classification. Un système de classification de référence est détaillé dans un second temps. Dans un troisième temps, nous présentons une nouvelle méthode de fusion qui modélise l'opération de fusion par un arbre binaire. Finalement les deux méthodes sont comparées. La deuxième section introduit de nouvelles modalités pour améliorer les résultats de la classification. Les caractéristiques de texte et de mouvement sont présentées. Une nouvelle étude des systèmes de fusion est alors conduite pour conclure sur une extension de la méthode initiale.

4.1 La fusion

La fusion consiste à réunir l'information provenant de diverses sources pour en former une seule. Dans la section précédente, la fusion était réalisée naturellement de façon simple pour combiner la couleur et la texture. La première partie de cette section introduit les difficultés inhérentes à la fusion au cours d'un état de l'art. Ensuite nous proposons de résoudre le problème de la fusion avec les algorithmes génétiques. Pour cela un système de référence utilisant les machines à vecteurs

de support est proposé. Ensuite notre algorithme de fusion est décrit puis comparé au système de référence.

4.1.1 Les difficultés de la fusion

Pour l'instant seules les caractéristiques de couleur et de texture étaient prises en compte pour la classification de plans vidéos en classes sémantiques. Et nous avons déjà pu observer la difficulté de combiner ces deux types de caractéristiques. La fusion peut s'opérer à différentes étapes du processus de classification et de différentes manières. Cette multitude de combinaisons rend le choix d'un système de fusion particulièrement difficile. D'après les résultats des expériences précédentes, la fusion semble plus appropriée en aval de la chaîne de la classification. Pourtant la logique voudrait que la fusion se fasse en amont afin de tirer profit de la corrélation entre les modalités. En pratique la nécessité d'analyser ces relations de corrélation requiert des modèles plus complexes et les données disponibles lors de l'entraînement sont rarement suffisantes pour permettre d'estimer correctement ces derniers. De plus la complexité des modèles rend la tâche d'apprentissage beaucoup plus lourde et plus délicate avec un risque accru de sur-entraînement (systèmes trop spécialisés). Malgré les progrès réalisés avec les machines à vecteurs de support, l'approche conventionnelle pour réaliser la classification incluant la sélection des caractéristiques, la classification par classe et la fusion restent donc d'actualité (Kittler [4]).

La fusion de systèmes de classification est actuellement un domaine très actif de la recherche (Kuncheva and Jain [7], Kuncheva [6], Tseng et al. [13], Shi and Manduchi [12], Ruta and Gabrys [11]). Les premières approches portent sur la sélection du ou des meilleurs systèmes pour réaliser la fusion (Ruta and Gabrys [11]). Les scores obtenus par la classification peuvent également être combinés en utilisant des opérateurs mathématiques de base (Kuncheva [6], Tseng et al. [13]) comme la somme, le produit, le maximum, ... Le problème de la fusion peut également être vu comme un problème de classification (Verlinde et al. [14]), notamment en effectuant une classification bayesienne (Shi and Manduchi [12]) ou en utilisant les machines à vecteurs de support (Iyengar et al. [3], Wu et al. [16]). Des systèmes plus complexes réalisent conjointement la classification et la fusion. Les algorithmes de Boosting (Freund and Schapire [1]) par exemple permettent de construire des systèmes de classification dits faibles dont les sorties sont combinées. Cette idée est ensuite reprise et développée avec les algorithmes génétiques (Kuncheva and Jain [7]).

4.1.2 La fusion par les SVMs

L'approche naturelle pour mettre au point un système de fusion élaboré est d'utiliser un système de classification. Toutefois la fusion est un problème qui diffère de la classification dans le sens où les données ont déjà une signification particulière : une valeur élevée implique la présence d'une classe tandis qu'une valeur faible implique son absence. Le choix s'est porté sur les machines à vecteurs de support puisqu'elles permettent de séparer les données par un « hyperplan » ce qui semble le plus approprié pour des scores de classification. Soit $\mathcal{H}_{s,c,m}$ les hypothèses construites avec le modèle s, sur la classe c, et la modalité m, définissons alors le vecteur $h_{c,i} = \{\mathcal{H}_{s,c,m}(x_i),\ m \in$ [couleur,texture]$\}$ qui représente les scores de détection d'une classe sur l'ensemble des modalités et des systèmes de classification pour un élément $x_i \in \mathcal{E}$. Par la suite, ce vecteur sera noté $e = \{e_i\}$ pour simplifier les notations. L'entraînement d'une machine par classe permet alors de construire

les hypothèses de fusion $\mathcal{F}_{SVM,c}$ en utilisant un noyau gaussien (se référer à la section 3.3.3 à la page 65).

4.1.3 La fusion par les algorithmes génétiques

Les systèmes de classification ne permettent pas d'effectuer des opérations simples comme le minimum, le maximum ou d'appliquer des fonctions de normalisation sur les données de manière automatique. Pourtant ces opérations et fonctions sont tout indiquées pour la fusion. Afin de pouvoir utiliser ces opérations et pouvoir librement choisir le type de normalisation à effectuer sur les données, nous proposons d'utiliser les algorithmes génétiques pour sélectionner la meilleure formule menant à la fusion. La partie suivante décrit la représentation des combinaisons possibles pour effectuer la fusion. Ensuite nous verrons comment les algorithmes génétiques génèrent et sélectionnent les meilleures combinaisons.

Les arbres binaires pour la fusion

Les opérations (op) que nous avons retenues en premier lieu sont la somme, le produit, le maximum et le minimum. Elles ont un sens particulier dans un cadre probabiliste qui convient parfaitement au problème de la fusion. La somme permet de mettre l'accent sur la présence d'un concept dans une des modalités au moins. Le produit valorise les concepts présents dans toutes les modalités. Le maximum et le minimum permettent d'effectuer une sélection par vote. Pour que ces opérations soient cohérentes, les scores issus des hypothèses sont normalisés pour représenter des probabilités p_i. En poursuivant dans un cadre probabiliste, nous faisons apparaître la notion de probabilité a priori en attachant un poids entre 0 et 1 à chaque entrée e_i et également une fonction complément $\overline{p_i}$. La normalisation peut être effectuée de plusieurs façons : min-max (mM), gaussienne (Gv2 : variance 2, Gv3 : variance 3), sigmoïde (IGv2 : variance 2, IGv3 : variance 3). Pour chaque entrée i, posons :

$$\begin{aligned} m_i &= \min_j(e_{ij}) \\ M_i &= \max_j(e_{ij}) \\ \mu_i &= \frac{\sum_j e_{ij}}{N_j} \\ \sigma_i &= \frac{\sum_j (e_{ij} - \mu_i)}{N_j - 1} \end{aligned} \tag{4.1}$$

Les fonctions de normalisation sont alors définies par :

$$\begin{aligned} mM(e_i) &= \frac{e_i - m_i}{M_i - m_i} \\ GvK(e_i) &= \frac{e_i - \mu_i}{2\sigma_i K} + 0.5 \\ IGvK(e_i) &= \frac{1}{1 + \exp\left(-\frac{10\cdot(e_i-\mu_i)}{\sigma_i K}\right)} \end{aligned} \tag{4.2}$$

Nous souhaitons une représentation qui puisse modéliser l'ensemble des combinaisons possibles avec les opérations citées sur n entrées. Toutes les opérations utilisées sont associatives et peuvent donc être représentées par une suite d'opérations sur deux opérandes. Les arbres binaires sont alors une bonne solution pour représenter l'ensemble des fonctions possibles. Les feuilles correspondent aux réponses aux différentes hypothèses tandis que les noeuds identifient les opérations à mener. La structure hiérarchique implique un ordre dans le déroulement des opérations ; ce qui permet d'effectuer certaines opérations en priorité (une somme avant un produit par exemple). La figure 4.1 illustre la représentation d'une fonction par un arbre binaire. En fait, il existe une bijection qui permet de passer d'un arbre binaire complet à $n-1$ noeuds internes à l'écriture d'une formule sur n opérandes composées d'opérateur sur deux opérandes.

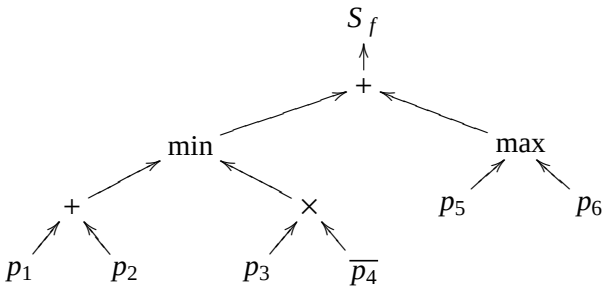


FIG. 4.1 – Exemple d'une fonction du fusion basée sur des opérations simples.

Le nombre de configurations possibles d'un arbre binaire complet à n noeuds internes est égale au n-ème nombre de Catalan C_n, c'est à dire :

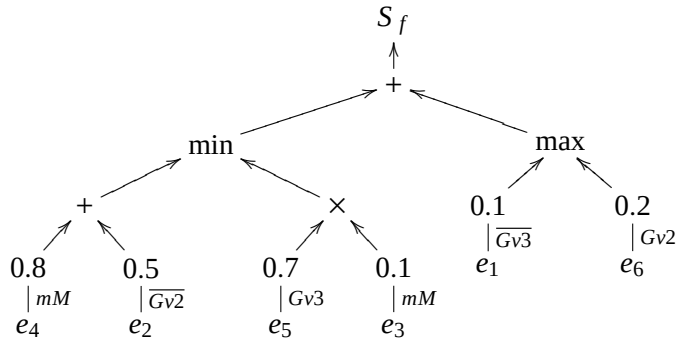
$$C_n = \frac{1}{n+1} C_{2n}^n \tag{4.3}$$

Sachant que pour chaque noeud quatre opérations sont proposées, le nombre de formules possibles est alors de $4^n * C_n$, ce qui donne pour les premières valeurs : 4, 32, 256, 3 584, 43 008, 540 672, ... Le nombre de combinaisons croit rapidement sachant que nous n'avons pas tenu compte du choix des probabilités a priori, des fonctions de normalisation, de la fonction complément et de l'ordre des entrées (ajout d'un facteur de $(n+1)^1 0 (n+1)^5 (n+1)^2 (n+1)!$). La figure 4.2 illustre l'opération de fusion dans sa totalité.

Etant donné le nombre de combinaisons possibles, il n'est pas concevable d'effectuer une recherche exhaustive de la solution et les algorithmes génétiques sont une méthode adaptée pour trouver efficacement la meilleure combinaison rapidement.

Les algorithmes génétiques

Les algorithmes génétiques sont des procédures qui s'inspirent des mécanismes de sélection naturelle et des phénomènes génétiques. Le principe de base consiste à simuler le processus d'évo-



$$S_f = \min(0.8mM(e_4) + 0.5\overline{Gv2(e_2)}, 0.7Gv3(e_5) \times 0.1mM(e_3)) + \max(0.1\overline{Gv3(e_1)}, 0.2Gv2(e_6))$$

FIG. 4.2 – Exemple d'une fonction de fusion décrite par un chromosome.

lution naturelle dans un environnement hostile. Ces algorithmes utilisent un vocabulaire similaire à celui de la génétique.

Pour un problème d'optimisation donné, un individu représente un point de l'espace des états. On lui associe la valeur du critère à optimiser. L'algorithme génère ensuite de façon itérative des populations d'individus sur lesquelles on applique des processus de sélection, de croisement et de mutation. La sélection a pour but de favoriser les meilleurs éléments de la population, tandis que le croisement et la mutation assurent une exploration efficace de l'espace des états.

Le mécanisme consiste à faire évoluer, à partir d'un tirage initial, un ensemble de points de l'espace vers le ou les optima d'un problème d'optimisation. L'ensemble du processus s'effectue à taille de population constante, que nous notons n_P.

Afin de faire évoluer ces populations de la génération k à la génération k+1, trois opérations sont effectuées pour tous les individus de la génération k :

- Une sélection d'individus de la génération k est effectuée en fonction du critère à optimiser ou plus généralement du critère d'adaptation au problème, on cherche ainsi à privilégier la reproduction des bons éléments au détriment des mauvais,
- Des opérateurs d'exploration de l'espace sont ensuite utilisés pour « élargir » la population et introduire de la nouveauté d'une génération à l'autre :
- L'opérateur de croisement est appliqué avec une probabilité Pc à deux éléments de la génération k (parents) qui sont alors transformés en deux nouveaux éléments (les enfants) destinés à les remplacer dans la génération k+1,
- Certaines composantes (les gènes) de ces individus peuvent ensuite être modifiés avec une probabilité Pm par l'opérateur de mutation. Cette procédure vise à introduire de la nouveauté dans la population.

L'évolution de la population permet de trouver des solutions qui se raffinent avec le temps. Par nature, les algorithmes génétiques peuvent évoluer indéfiniment et mettre au jour de nouvelles solutions. Toutefois il est préférable de définir des critères d'arrêt. Deux critères sont souvent utilisés : le premier détecte la stabilité du critère à optimiser au cours du temps ;le second impose un nombre

d'itérations limité.

Au final la théorie des algorithmes génétiques est particulièrement simple et sa mise en oeuvre est facile. Seules les fonctions de mutation et de fusion sont délicates à définir et spécifiques à chaque situation. Nous proposons d'étudier les solutions qui nous sont offertes sur les arbres binaires complets.

Opérateurs génétiques sur des arbres binaires

Nous rappelons que nous recherchons la meilleur formule de fusion qui puisse être générée à partir d'un lot d'opérateurs sur deux opérandes. Pour cela, nous avons vu qu'il y avait une bijection entre les arbres binaires complets à $n-1$ noeuds internes et les formules à n opérandes. Nous allons travailler sur les arbres binaires complets afin d'étudier les possibilités qui nous sont offertes pour manipuler les formules de fusion dans l'objectif de créer une population initiale, d'effectuer des mutations et des fusions.

La première question qui se pose porte sur les méthodes de génération aléatoire des arbres binaires complets à n feuilles. Cette étape est primordiale pour construire la population intiale. La plupart des algorithmes pour générer des arbres binaires complets utilisent une représentation de l'arbre par des chaînes. Ensuite un ensemble de règles et de grammaires permettent de construire aléatoirement des arbres valides (Mäkinen [8]). Une méthode particulière se détache de ce formalisme tout en permettant de construire des arbres équiprobables. Cette méthode s'appuie sur l'identité de Rémy (Remy [10]) pour itérativement construire l'arbre de taille souhaitée. Une illustration de cette identité est proposée sur la figure 4.3. L'algorithme procède comme suit :

1. Supposons que nous ayons un arbre binaire complet avec k noeuds internes et $k+1$ feuilles,
2. Sélectionnons aléatoirement un noeud ($\diamond$) parmi les $2k+1$ noeuds existants,
3. Remplaçons ($\diamond$) par ($\star$) et choisissons aléatoirement si ($\diamond$) sera le fils gauche ou droit du nouveau noeud. Le fils créé par cette insertion est alors une nouvelle feuille ($\circ$) (figure 4.3),
4. Répétons la procédure jusqu'à obtenir un arbre composé de n noeuds internes,

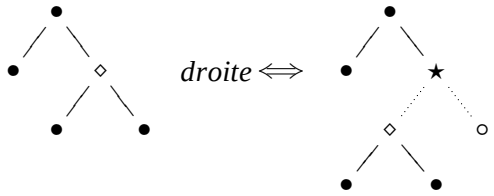


FIG. 4.3 – L'identité de Rémy pour ajouter ou supprimer une feuille.

Cet algorithme est d'autant plus intéressant puisqu'il suggère aussi de quelle manière nous allons pouvoir mettre en oeuvre les opérateurs génétiques de mutation et de fusion. Commençons par le cas plus simple de la mutation d'un individu. Il s'agit de transformer localement ce dernier pour renouveler la population et découvrir de nouveaux horizons. Nous proposons d'utiliser l'identité de Rémy pour « tailler » l'arbre puis le « régénérer ». Pour cela, un sous-arbre aléatoire est supprimé dans l'arbre. Ensuite l'arbre est aléatoirement régénéré en rajoutant aléatoirement de

nouvelles feuilles avec de nouveaux opérateurs. Un procédé similaire est employé pour réaliser la fusion de deux individus. Un sous-arbre de la mère est aléatoirement sélectionné. Suffisamment de feuilles sont ensuite supprimées du père pour y placer le sous-arbre sélectionné. Ces interventions modifient la structure de l'arbre et directement la manière dont la fusion est réalisée.

Les fonctions de normalisation et les probabilités a priori ne sont pas modifiées par ces interventions. Elles sont donc traitées indépendamment de manière plus traditionnelle. La mutation renouvelle aléatoirement une partie des fonctions et des probabilités. La fusion croise les valeurs associées aux entrées de la mère à celle du père. L'ordre des entrées est soumis à un traitement particulier puisqu'il faut garantir l'unicité et la présence de chacune d'elles. Pour cela la mutation est obtenue par inversion des éléments d'un sous-ensemble aléatoire et la fusion est obtenue par la copie d'un sous-ensemble de la mère puis le remplissage des parties vides avec les éléments non présents du père (figures 4.4).

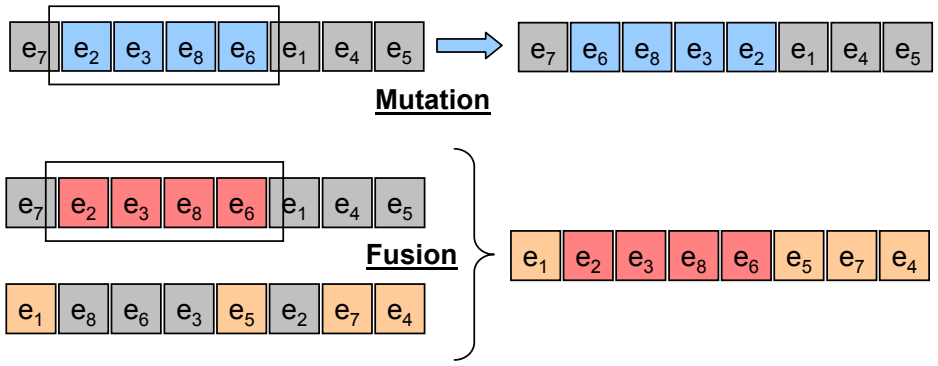
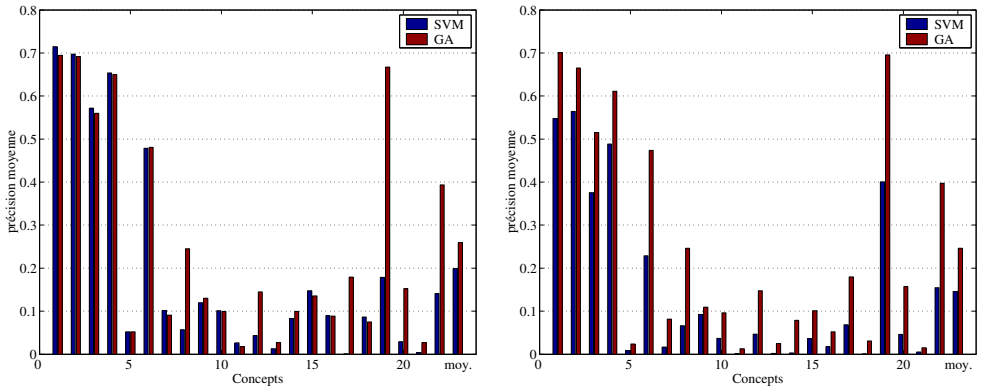


FIG. 4.4 – Opérateurs génétiques conservant l'existence et l'unicité des valeurs.

4.1.4 Les résultats

Cette partie compare les deux systèmes de fusion que nous avons présentés. La précision moyenne par concept est donnée sur les figures 4.5(a) et 4.5(b). La figure de droite présente les résultats de la fusion des informations de couleur et de texture issues de la classification par les machines à vecteurs de support. La figure de gauche présente les résultats de la fusion des informations de couleur et de texture issues de l'ensembles des systèmes de classifications (GMM, KNN et SVM). Dans le premier cas, nous observons dans l'ensemble des performances similaires entre une méthode de fusion par les machines à vecteurs de support ou les algorithmes génétiques. Toutefois pour 5 classes les algorithmes génétiques obtiennent des scores bien meilleurs et en moyenne un gain de 5% est alors obtenu. Dans le deuxième cas, une nette chute des performances est observée lors d'une fusion par les machines à vecteurs de support tandis que les algorithmes génétiques maintiennent leurs performances. Les machines à vecteurs de support ne parviennent pas à correctement fusionner les nombreuses sorties des systèmes de classification. Apparemment ils sont plus sensibles aux entrées bruitées ou erronées (par exemple celles fournies par les GMM). Comme nous pouvons l'observer sur le figure 4.6, la fonction de fusion sélectionne les fonctions de normalisation adaptées

au problème et aux caractéristiques. De plus l'ordre des opérations, les opérandes et les probabilités a priori sélectionnés par l'algorithme génétique favorisent certaines caractéristiques pour obtenir une meilleure classification. Ainsi la classification de la classe *présentateur du journal* est réalisée principalement par les SVM qui se trouvent au sommet de la pyramide. Le système de fusion pour la classe *végétation* est plus complexe, mais nous remarquons que les caractéristiques obtenues par les GMM sont toujours en bas de la pyramide et leur impact est donc plus faible. A présent que



(a) Classification sur les SVM (b) Classification sur l'ensemble des systèmes

FIG. 4.5 – Fusion des informations de couleur et de texture.

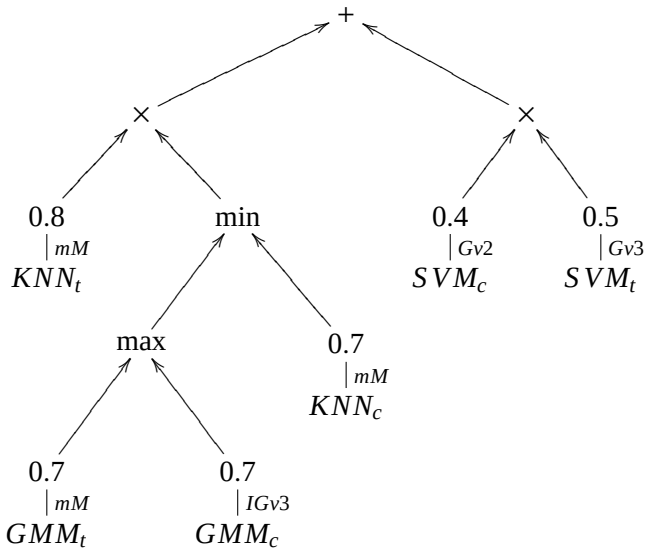
nous avons introduit un système de fusion, de nouvelles modalités sont extraites de la vidéo pour parfaire la détection des classes.

4.2 Introduction du texte et du mouvement

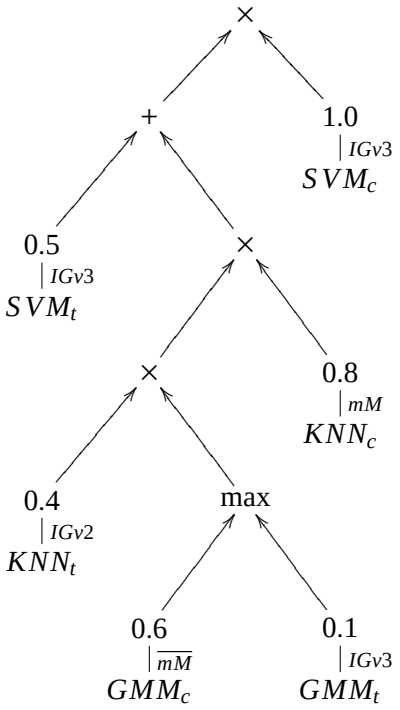
La vidéo offre d'innombrables sources d'information que nous n'avons pas exploitées. Jusqu'à présent nous avons considéré uniquement les caractéristiques visuelles. Or les méthodes de fusion offrent la possibilité d'en inclure d'autres comme le texte et le mouvement. Les parties suivantes présentent ces nouvelles caractéristiques.

4.2.1 Texte

Le texte ou les voix associés à une vidéo sont d'une importance capitale. En effet, ils contiennent explicitement une information sémantique utilisable. Pour capturer cette information, le texte est soit obtenu par le télétexte ou par un système automatique de reconnaissance vocale. Toutefois l'obtention du texte ne résout pas tous les problèmes. Contrairement aux documents écrits, le texte (ou les voix) présent dans un plan est court et contient peu de mots. De plus les contenus visuels et textuels sont souvent désynchronisés. Par exemple une personne interpellée par son prénom apparaîtra souvent dans les plans suivants, voire pas du tout. De même les nouvelles sont introduites par le présentateur avant le reportage.



(a) Fusion pour la classe *présentateur du journal*(2)



(b) Fusion pour la classe *végétation*(8)

FIG. 4.6 – Structure de la fonction de fusion pour deux classes.

Extraction

Dans le cas présent le texte est fourni par un système automatique de reconnaissance de la parole (Gauvain et al. [2]). Il est ensuite analysé avec les techniques habituelles employées pour le traitement des documents écrits. Tout d'abord les racines des mots sont extraites en utilisant l'algorithme très répandu de Porter [9]. Elles sont ensuite filtrées pour retirer les mots apportant peu de sens comme les articles, les pronoms, les conjonctions de coordinations, ... 12 000 mots ont été recensés sur l'ensemble des vidéos de TRECVID dont 16% apparaissaient très rarement, c'est à dire moins de cinq fois. Après suppression des mots rares, un dictionnaire de 2 000 mots est obtenu et retenu pour des vidéos en anglais.

Synchronisation

Malheureusement un plan n'est pas une unité sémantique et les thèmes abordés par le texte se trouvent souvent dans les plans voisins. Pour palier à ce problème de synchronisation, le texte d'un plan est étendu au texte de ses voisins. Cette opération est équivalente à définir une scène comme étant le plan courant et ses voisins. La signature textuelle est alors obtenue sur la scène. L'approche la plus logique pour représenter le texte est l'utilisation des modèles vectoriels. La signature est alors le vecteur d'occurrence des mots du dictionnaire dans la scène. L'ensemble des

signatures est finalement sujet à la classification en classes sémantiques. Pour cela, la classification par des mélanges de gaussiennes et par les machines à vecteurs de support sont utilisés (figure 4.7). Dans l'ensemble les performances sont plutôt faibles comparées à une classification utilisant les caractéristiques visuelles. Nous conservons pour la suite, les résultats de la classification par les machines à vecteurs de support sur les « scènes ».

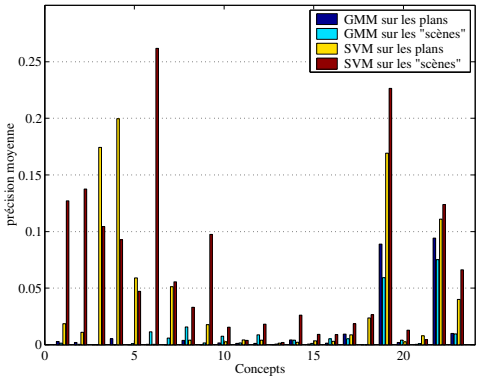


FIG. 4.7 – Classification du texte.

Détection des mots clefs

La classification de signatures créées sur la totalité du texte n'a pas fourni les performances escomptées. Et une méthode plus intuitive de détection des concepts a été développée. L'idée est d'identifier les mots les plus pertinents de chaque classe puis de calculer un score de classification en fonction du nombre de mots pertinents présents dans la scène. Une méthode semi-automatique est proposée pour réaliser cette identification. Une première liste est générée automatiquement par classe à partir de l'ensemble d'entraînement. Les mots sont ordonnés en fonction de leur occurrence dans les classes. Les mots effectivement pertinents pour chaque classe sont alors sélectionnés manuellement. Cependant certaines classes ne sont pas exprimées par le texte et aucun mot ne permet leur description et identification. Par exemple la classe *nature et végétation* est difficilement reconnaissable à partir du texte. Le tableau 4.1 montre la répartition moyenne normalisée des mots clefs sélectionnés autour du plan t contenant les concepts associés. Les lignes nulles correspondent aux concepts pour lesquels aucun mot clef n'a pu être identifié pour les décrire. Les valeures obtenues mettent en évidence l'importance de prendre en compte le décalage entre l'expression visuelle et textuelle d'un concept. Pour cela nous procédons comme dans le cas précédent. Les mots détectés sur un plan participent aux scores de détection des plans voisins. Cette participation peut être constante et d'une valeur unitaire ou elle peut être pondérée en utilisant les poids calculés dans le tableau 4.1.

La figure 4.8 montre les résultats de la classification lorsque les mots clefs sont extraits sur le plan, sur la « scène » et lorsque leur diffusion est pondérée. Pour toutes les classes les meilleures performances sont réalisées par une signature pondérée calculée sur la « scène ». Nous conservons donc cette caractéristique pour la fusion.

Id	Concept	t-3	t-2	t-1	t	t+1	t+2	t+3
1	studio	0.43	0.48	0.64	1	0.79	0.57	0.46
2	présentateur du journal	0.40	0.46	0.58	1	0.70	0.48	0.38
3	-présentateur masculin	0.34	0.44	0.52	1	0.70	0.45	0.34
4	-présentateur féminin	0.48	0.48	0.67	1	0.73	0.53	0.43
5	Bill Clinton	0.68	0.64	0.71	1	0.99	0.82	0.76
6	basketball	0.80	0.86	0.82	1	0.87	0.80	0.79
7	hockey sur glace	0.50	0.60	0.69	0.89	0.91	1	0.83
8	nature et végétation	0	0	0	0	0	0	0
9	-fleur	0.25	0.6	0.7	1	0.55	0.4	0.4
10	-arbre	0	0	0	0	0	0	0
11	-forêt	0	0	0	0	0	0	0
12	-verdure	0	0	0	0	0	0	0
13	désert	0	0	0	0	0	0	0
14	montagne	0.33	0.33	0.66	1	1	0.33	0.33
15	plage	0.6	0.46	0.33	1	0.86	0.33	0.73
16	route	0.52	0.61	0.76	0.97	0.85	1	0.79
17	gens	0.83	0.83	0.90	0.74	1	0.97	0.91
18	avion	0.43	0.61	0.71	1	0.90	0.75	0.51
19	incrustation de texte	0	0	0	0	0	0	0
20	bâtiment	0.72	0.72	0.68	1	1	0.66	0.50
21	paysage urbain	0	0	0	0	0	0	0
22	autre	0	0	0	0	0	0	0

TAB. 4.1 – Occurrence des mots clefs autour du plan courant.

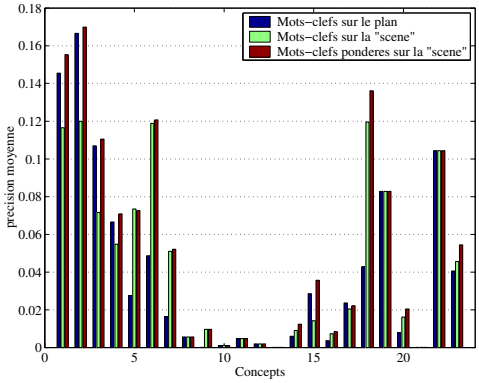


FIG. 4.8 – Détection des mots clefs.

4.2.2 Mouvement

Le mouvement permet de capturer l'activité du plan et ainsi d'établir une discrimination sur l'ambiance des plans, c'est à dire agité, calme, zoom, ... Une étude approfondie permet d'avoir une information plus précise comme le mouvement de la caméra et la trajectoire des objets. Nous avons décidé d'adopter une approche intermédiaire qui capture le mouvement de la caméra et l'activité générale des objets. L'analyse du mouvement débute systématiquement par une estimation du mouvement des pixels entre deux images. Cette estimation est communément appelée calcul du flux optique. Toutefois les caractéristiques devant être calculées pour toutes les images, une approche dans le domaine compressé est la bienvenue puisque les vecteurs de mouvement des macro blocs sont directement disponibles pour de nombreuses images.

Mouvement de la caméra

L'algorithme présenté par Wang and Huang [15] estime les paramètres du mouvement de la caméra à partir du flux MPEG. Le mouvement est modélisé par un modèle affine à quatre paramètres mesurant l'intensité de la translation, de la rotation et du zoom :

$$\begin{aligned} v_x &= z((x-x_0)cos\alpha + (y-y_0)sin\alpha) + t_x - x \\ v_y &= z(-(x-x_0)sin\alpha + (y-y_0)cos\alpha) + t_y - y \end{aligned} \tag{4.4}$$

$\vec{v} = (v_x, v_y)$ est le vecteur du mouvement d'un point (x, y) de l'image. (x_0, y_0) est le centre de l'image. $\vec{t} = (t_x, t_y)$ est le mouvement de la caméra en translation. α est l'angle de la rotation et z est l'intensité du zoom. L'équation 4.4 se met aussi sous la forme :

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} . \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{4.5}$$

Les paramètres de la rotation et du zoom ne sont plus explicites, mais cette forme linéaire est plus appropriée pour une minimisation aux moindres carrées sur l'ensemble des observations disponibles. Ces dernières sont fournies par le codage MPEG. Comme nous l'avons vu dans la section 1.2 à

la page 8, les macro blocs des image de type P et B sont obtenus par reconstruction à partir des images de type I ou P les plus proches. Ces dernières sont désignées par images de référence. Dans le cas des images de type P, un unique vecteur mouvement est disponible par rapport à l'image de référence précédente. Par contre, les images de type B peuvent avoir deux vecteurs mouvement, le premier par rapport à l'image de référence précédente et le second par rapport à l'image de référence suivante. Afin d'utiliser tous les vecteurs disponibles, les vecteurs mouvements sont redéfinis de la manière suivante :

$$\vec{v}_P(t) = \frac{\overrightarrow{mvt}_P(t, t-\delta_p)}{\delta_p}$$
$$\vec{v}_B(t) = \frac{1}{2}\left(\frac{\overrightarrow{mvt}_B(t, t-\delta_p)}{\delta_p} - \frac{\overrightarrow{mvt}_B(t, t+\delta_s)}{\delta_s}\right) \tag{4.6}$$

$\vec{mvt}_P(t,t')$ et $\vec{mvt}_B(t,t')$ correspondent aux vecteurs de mouvement fournis par MPEG entre une image de type B ou P à l'instant t et leur référence à l'instant t'. δ_p et δ_s correspondent au nombre d'images entre l'image à l'instant t et les images de référence précédente et suivante. Comparer à la méthode proposée par Wang and Huang [15], nous avons inclus un facteur de normalisation supplémentaire pour les images de type B et nous ne traitons pas les images de type I. L'estimation est finalement obtenue à partir des méthodes classiques d'algèbre linéaire. L'observation de N réalisations fournit les $2N$ équations suivantes :

$$Y = H.X \tag{4.7}$$

$$Y = \begin{bmatrix} v_{x1} \\ v_{y1} \\ v_{x2} \\ v_{y2} \\ \vdots \\ v_{xN} \\ v_{yN} \end{bmatrix}, H = \begin{bmatrix} x_1 & y_1 & 1 & 0 \\ y_1 & -x_1 & 0 & 1 \\ x_1 & y_1 & 1 & 0 \\ y_1 & -x_1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & 1 & 0 \\ y_N & -x_N & 0 & 1 \end{bmatrix}, X = \begin{bmatrix} a \\ b \\ t_x \\ t_y \end{bmatrix}$$

Y est le vecteur des observations. H est la matrice de localisation des observation (les macro blocs dans le cas présent). X est le vecteur des paramètres à estimer. La solution aux moindres carrées $\hat{X}$ est donnée par :

$$\hat{X} = (H^t H)^{-1} H^t Y \tag{4.8}$$

Malheureusement cette estimation du mouvement de la caméra est perturbée par la présence d'objets mobiles dans la scène. Une approche itérative par élimination des macro blocs permet de créer un masque distinguant l'arrière plan de l'avant plan. L'estimation des paramètres est alors faite uniquement sur les observations du masque. Le masque est mis à jour en fonction des erreurs de reconstruction sur l'ensemble des observations. La méthode de minimisation aux moindres carrées employée suppose que l'erreur suit une loi gaussienne. Il est donc naturel de rejeter les macro blocs dont l'erreur ne satisfait pas cette condition. Pour cela, la densité de probabilité des erreurs sur le masque est calculée. Puis l'ensemble des vecteurs dont l'erreur n'est pas incluse dans l'intervalle

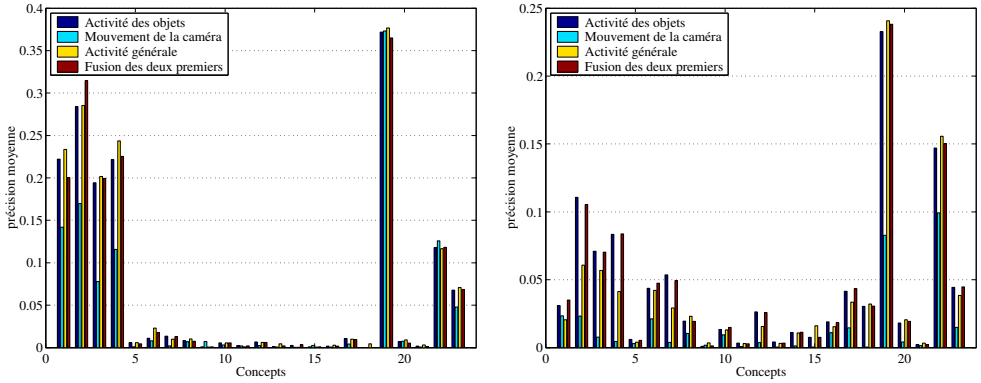
de confiance à 90% forme le nouveau masque. Le procédé est répété jusqu'à obtenir la stabilisation de l'estimation. Pour représenter le mouvement de la caméra, nous avons choisi les paramètres de translation selon l'horizontale et la verticale, le facteur de zoom et l'angle de rotation dans l'intervalle $]-\pi,\pi]$, comme ils sont exprimés dans l'équation 4.4.

Activité du plan

L'activité du plan est principalement caractérisée par l'ensemble du mouvement présent. Les histogrammes sont alors une représentation adaptée. Afin de distinguer le mouvement relatif du mouvement absolu, deux histogrammes sont définis. Le premier caractérise le mouvement des macro blocs sans compensation du mouvement. Le mouvement de la caméra et des objets est alors capturé. Le second caractérise le mouvement des objets. Pour cela le mouvement de la caméra est estimé puis il est déduit pour obtenir le mouvement propre des objets. Une analyse préliminaire de l'ensemble d'entraînement a montré que l'amplitude des vecteurs de mouvement fournis par MPEG est souvent comprise dans l'intervalle $[-20, 20]$. La quantification est donc restreinte à ce dernier avec huit valeurs de quantification par direction. Afin de caractériser l'activité sur l'ensemble du plan, les histogrammes moyens sont calculés sur l'ensemble des images P et B composant le plan.

Classification

Le mouvement n'est pas une caractéristique particulièrement discriminante. Il ne permet pas à lui seul de déterminer les concepts sémantiques attachés à un plan. Trop souvent le mouvement est similaire dans des plans sans aucune relation. Pour ces raisons, nous n'avons pas trouvé nécessaire d'effectuer la classification avec les k plus proches voisins, mais avec les deux autres systèmes de classification proposés (GMM et SVM). Les scores obtenus sont alors fusionnés avec les scores précédemment étudiés pour confirmer, valider ou accentuer les décisions lors de la classification. La figure 4.9 donne la précision moyenne par classe lors d'une classification par le mouvement. D'après les valeurs obtenues, le mouvement propre de la caméra n'est pas une caractéristique particulièrement efficace pour les classes étudiées et elle ne sera pas conservée pour la fusion.



(a) Classification par des GMM (b) Classification par les SVM

FIG. 4.9 – Classification du mouvement des plans.

4.3 Etude de la fusion

Cette section débute par une présentation des performances fournies par les deux systèmes de fusion présentés auparavant. Nous verrons en particulier l'impact des différentes modalités sur le résultat de la fusion. Ensuite de nouveaux opérateurs de fusion sont présentés pour mieux explorer l'espace des formules de fusion. Nous recherchons à améliorer l'effet des caractéristiques intéressantes et supprimer les caractéristiques néfastes. Finalement l'évaluation du système d'indexation et de recherche est faite dans son ensemble, c'est à dire lors d'une recherche mixte à l'aide d'une image exemple et de mots clefs.

4.3.1 Comparaison des systèmes de fusion sur plusieurs modalités

L'utilisation des arbres binaires pour modéliser une fonction de fusion a donné des résultats forts probants comme le montre la figure 4.10. Cette figure présente la précision moyenne des systèmes de fusion par classe. Les deux systèmes présentés auparavant sont étudiés, c'est à dire les machines à vecteurs de support et les algorithmes génétiques avec des arbres binaires pour modéliser la fonction de fusion. Afin d'évaluer l'importance de chaque caractéristique, les expériences sont réalisées dans le cadre de la fusion des informations visuelles, puis des informations visuelles et textuelles, ensuite des informations visuelles et de mouvement et finalement de l'ensemble. Les algorithmes génétiques et la fonction de fusion proposée parviennent à mieux tirer profit des différentes entrées qu'une fusion par les machines à vecteurs de support. Le texte apporte une grande amélioration de la précision moyenne pour de nombreuses classes et cette amélioration est particulièrement forte avec une fusion par les SVM. D'ailleurs les SVM fournissent des performances comparables aux GA uniquement lorsque le texte est combiné à la couleur et à la texture. La combinaison de toutes les caractéristiques ne permet pas une fusion par les SVM convaincante contrairement aux GA. Une comparaison des performances obtenues sur l'ensemble et sur les autres combinaisons par paire de modalités montre que les GA tirent correctement profit de l'ensemble de l'information fournie. Cependant le modèle de fusion par les GA choisi impose l'utilisation de toutes les caractéristiques y compris les plus néfastes et ne permet pas l'utilisation multiple des caractéristiques intéressantes. Ces défauts expliquent en partie les légères pertes de performances qui sont notables entre une fusion de l'ensemble des modalités et les combinaisons par paire de modalités. La prochaine section propose une solution à ce problème.

4.3.2 Fonctions de fusion de taille variable

Nous avons vu dans la section précédente qu'une fusion sur l'ensemble des modalités n'offrait pas systématiquement des performances similaires ou meilleures à une fusion sur des modalités présélectionnées. En effet, la méthode proposée ne permet pas de supprimer entièrement les effets néfastes de certaines d'entre elles. Pour remédier à ce problème, nous proposons d'employer des arbres de taille variable combinés à une sélection aléatoire des entrées actives pour représenter la solution. Cette représentation s'apparente de plus en plus à celle employée par la programmation génétique (Koza [5]) dont les solutions sont des programmes ou des fonctions. L'identité de Rémy que nous avons utilisé dans les opérateurs génétiques précédents nous offre la possibilité d'étendre ces opérateurs à des arbres de taille variable. La « taille » ou la « régénération » se font de la même

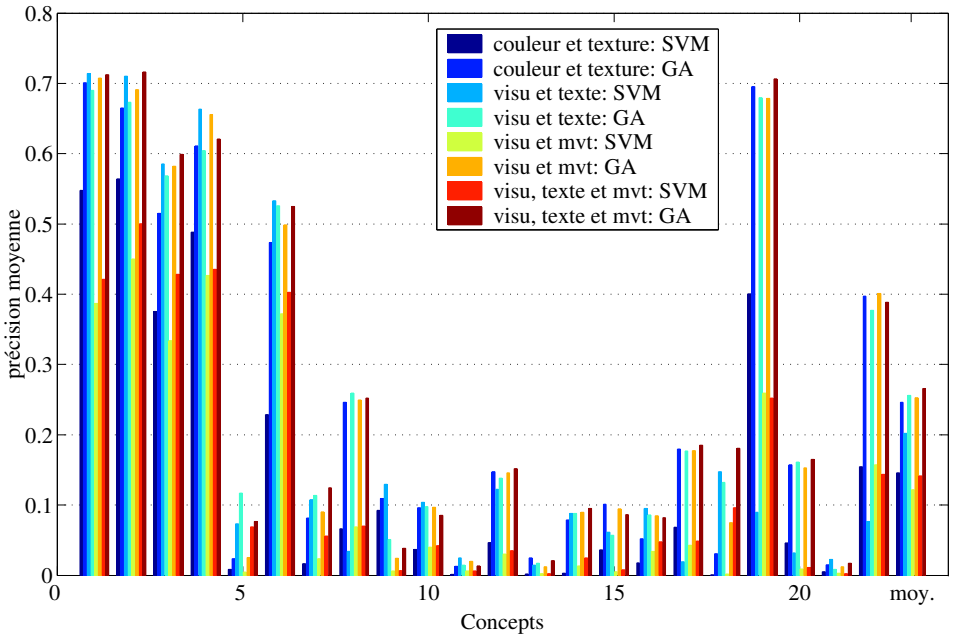


FIG. 4.10 – Fusion des différents systèmes de classification et des différentes modalités et caractéristiques.

manière mais cette fois la taille des arbres est choisi aléatoirement. Les entrées sont également tirées aléatoirement avec la possibilité d'avoir des redondances et des entrées non présentes. Cette approche présente deux avantages majeurs : les entrées superflues ne sont pas utilisées pour la fusion et les opérations superflues ne sont pas incluses dans le procédé de fusion.

La figure 4.11 compare l'approche d'origine utilisant un arbre de taille fixe et son extension utilisant un arbre de taille variable. L'amélioration est significative sur trois classes et négligeable sur le reste. L'augmentation de la complexité de l'arbre et des structures tolérées augmente également le nombre des configurations équivalentes. La recherche par les algorithmes génétiques devient moins efficace avec une perte de rapidité et un espace moins bien exploré.

4.3.3 Recherche mixte

Cette partie a pour objectif de comparer les systèmes automatiques d'indexation et de recherche utilisant différents modes de recherche : la requête est construite à partir d'un exemple, d'une classe ou des deux. La méthode présentée dans la section 2.3.3 à la page 33 est utilisée comme système nécessitant un exemple pour requête. Les signatures utilisées sont décrites dans la section 2.5 à la page 38. Lorsque le bouclage de pertinence est actif (section 2.7.2 à la page 44), une itération est réalisée et l'utilisateur sélectionne uniquement les 20 premiers plans pertinents. La recherche par classe est effectuée en utilisant les scores obtenus par le système de fusion décrit dans la section 4.1.3 sur l'ensemble des caractéristiques présentées dans la section 4.1.3. La requête est uniquement consituée de la classe en cours d'évaluation. Finalement la recherche mixte combine les scores obtenus par une recherche par exemple et les scores de la fusion. Dans l'état actuel, la combinaison des deux

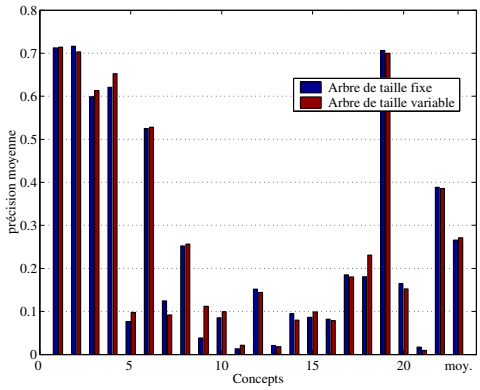


FIG. 4.11 – Comparaison des performances de la fusion sur des arbres de taille fixe et de taille variable.

scores est effectuée par une somme non pondérée. La figure 4.12 montre les performances moyennes par classe pour chaque mode de recherche. Nous observons qu'une recherche par classe surpasse de loin les autres modes. Cette situation s'explique par la méthode d'évaluation qui utilise l'information sémantique pour juger si un plan est pertinent ou non. De plus les classes sont estimées pour répondre à une classification sémantique qui concorde avec l'évaluation. Notons que l'évaluation est cohérente avec l'attente des utilisateurs qui sont essentiellement intéressés par le contenu sémantique ; nous voyons que l'étape de classification ou d'annotation est efficace et nécessaire pour une recherche sur le contenu sémantique. La recherche mixte apporte également un gain de performance significatif par rapport à une recherche uniquement à partir d'un exemple (avec ou sans asservissement). Ce mode offre la possibilité de retrouver en premier lieu les plans visuellement et sémantiquement similaires, puis les plans sémantiquement similaires.

4.4 Conclusion

Ce chapitre a abordé le thème de la fusion d'information. En particulier nous avons étudié le problème de la fusion de systèmes de classification. L'objectif est d'utiliser au mieux l'information issue des différentes modalités, caractéristiques et systèmes de classification. Un système de référence utilisant les machines à vecteurs de support a été présenté, puis nous avons proposé une nouvelle méthode de fusion qui utilise les arbres binaires pour représenter les fonctions de fusion possibles. Cette méthode repose sur des opérations simples qui se sont révélées très performantes. Pour conclure nous avons comparé les performances obtenues lors d'une requête par une image exemple, d'une requête par mots clefs et d'une requête mixte.

L'ensemble de validation a été utilisé pour effectuer l'entraînement des systèmes de fusion. Toutefois plus de plans vidéo annotés pour entraîner les systèmes de classification et de fusion ne seraient pas superflus. Le prochain chapitre étudie les méthodes d'apprentissage actif pour permettre une annotation rapide et efficace des données nécessaires pour entraîner les modèles utilisés.

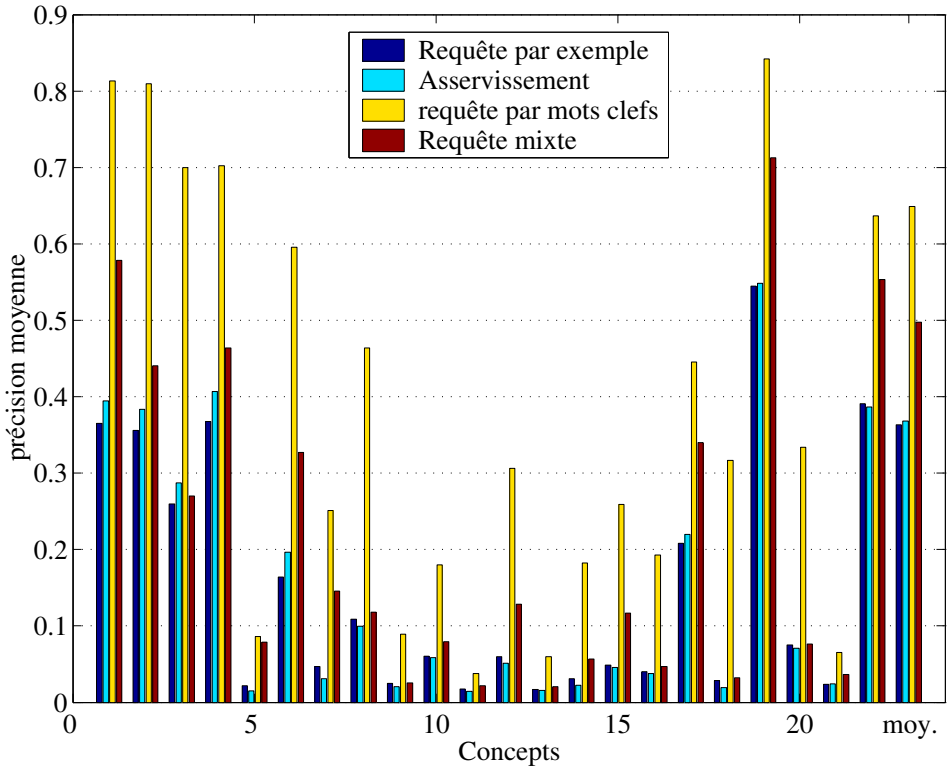


FIG. 4.12 – Comparaison des différents modes de recherche. *Recherche à partir d'une image exemple sans et avec une boucle d'asservissement. Recherche uniquement sur la classe. Recherche mixte à l'aide d'un exemple et d'une classe.*

Bibliographie

[1] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the International Conference on Machine Learning*, pages 148–156, 1996.

[2] J.L. Gauvain, L. Lamel, and G. Adda. The LIMSI broadcast news transcription system. *Speech Communication*, 37(1-2) :89–108, 2002.

[3] G. Iyengar, H. Nock, C. Neti, and M. Franz. Semantic indexing of multimedia using audio, text and visual cues. In *Proceedings of the International Conference on Multimedia and Expo*, 2002.

[4] Josef Kittler. A framework for classifier fusion : Is it still needed ? In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 45–56. Springer-Verlag, 2000. ISBN 3-540-67946-4.

[5] John R. Koza. *Genetic Programming : On the Programming of Computers by Means of Natural Selection*. MA : The MIT Press, 1992.

[6] Ludmila I. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 24(2) :281–286, february 2002.

[7] Ludmila I. Kuncheva and Lakhmi C. Jain. Designing classifier fusion systems by genetic algorithms. *IEEE Transactions On Evolutionary Computation*, 4(4) :327–336, september 2000.

[8] Erkki Mäkinen. Generating random binary trees : a survey. *Inf. Sci. Inf. Comput. Sci.*, 115 (1-4) :123–136, 1999. ISSN 0020-0255.

[9] Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3) :130–137, 1980.

[10] Jean-Luc Remy. Un procédé itératif de dénombrement d'arbres binaires et son application á leur génération aléatoire. In *Informatique Théorique et Applications*, volume 19, pages 179–195, 1985.

[11] Dymitr Ruta and Bogdan Gabrys. Classifier selection for majority voting. *Special issue of the journal of INFORMATION FUSION on Diversity in Multiple Classifier Systems*, 2004.

[12] X. Shi and R. Manduchi. A study on bayes feature fusion for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 8, june 2003.

[13] Belle L. Tseng, Ching-Yung Lin, Milind Naphade, Apostol Natsev, and John R. Smith. Normalized classifier fusion for semantic visual concept detection. In *Proceedings of the IEEE International Conference on Image Processing*, 2003.

[14] Patrick Verlinde, Gerard Chollet, and Marc Acheroy. Multi-modal identity verification using expert fusion. *Information Fusion*, 1(1) :17–33, 2000.

[15] Roy Wang and Thomas Huang. Fast camera motion analysis from MPEG domain. In *Proceedings of the IEEE International Conference on Image Processing*, pages 691–694, 1999.

[16] Y. Wu, E. Y. Chang, K. C.-C. Chang, and John R. Smith. Optimal multimodal fusion for multimedia data analysis. In *Proceedings of the ACM International Conference on Multimedia*, pages 572–579, 2004.

Chapitre 5

Apprentissage actif

Ce chapitre s'attaque au problème de l'annotation du contenu. Nous avons vu au cours des chapitres précédents qu'il était important et nécessaire d'avoir des vidéos annotées afin de pouvoir construire des systèmes de classification et de fusion efficaces. Cette tâche fastidieuse et ingrate est donc nécessaire pour effectuer une indexation par le contenu réellement efficace. Elle peut être réalisée entièrement manuellement au prix d'un investissement humain conséquent ou en partie en utilisant les méthodes d'apprentissage actif qui permettront d'effectuer une grande partie de l'annotation de manière automatique. Nous portons notre intérêt sur ces méthodes qui permettent de construire des modèles en minimisant le nombre d'annotation à effectuer et donc en limitant l'effort humain à apporter. Comme nous le verrons, les systèmes existants d'apprentissage actif présentent un inconvénient majeur qui est leur baisse significative de performance lorsque plusieurs échantillons doivent être annotés par l'utilisateur. Nous allons donc présenter une approche qui permet de réduire cette chute de performance. Dans un premier temps nous portons notre attention sur l'estimation d'un concept à la fois. Puis toujours dans l'esprit de réduire le nombre d'annotation, nous étudierons les possibilités de généraliser la méthode à plusieurs concepts.

5.1 L'apprentissage actif

5.1.1 Présentation

L'annotation du contenu est une tâche longue, délicate et sujette aux erreurs. Cependant elle est nécessaire pour de nombreuses applications allant de l'indexation à la construction de modèles statistiques. La réduction de l'effort fourni pour l'annotation a stimulé un fort intérêt dans la communauté scientifique de l'apprentissage des machines (« Machine learning »). Principalement deux approches ont été proposées pour résoudre ce problème : l'apprentissage semi supervisé et l'apprentissage actif. La première catégorie d'algorithmes utilise conjointement un petit ensemble annoté avec un ensemble plus grand non annoté (Nigam et al. [8]). Ce dernier n'apporte pas une information directe mais l'information sur la distribution de ces éléments est largement utilisé pour améliorer les modèles. L'idée principale est d'alterner l'estimation des classes des échantillons non annotés et le raffinement des modèles. La deuxième catégorie d'algorithmes procède différemment. Un algorithme d'apprentissage actif commence par un petit ensemble d'entraînement puis en faisant

intervenir l'utilisateur il augmente intelligemment et de manière progressive le nombre d'éléments annotés. Au fur et à mesure l'hypothèse de classification se raffine et accroît la précision de son estimation. De plus l'utilisation d'un ensemble d'entraînement réduit permet en théorie un entraînement facilité accompagné d'une meilleure généralisation de l'hypothèse de classification. Récemment une approche hybride appelée CO-EMT (Muslea et al. [7]) a été introduite. Elle combine une approche semi supervisé avec une approche active pour tirer profit de leur complémentarité. Dans la suite, nous portons notre attention uniquement sur les systèmes d'apprentissage actif.

La principale difficulté dans les systèmes d'apprentissage actif est de déterminer la suite optimale d'échantillons à annoter pour arriver au meilleur modèle. Deux approches se distinguent selon l'origine des échantillons à annoter. Dans le premier cas les échantillons sont générés par le système lui-même. D'après les données d'apprentissage initiales, le système génère de nouveaux échantillons qui une fois annotés vont lui permettre de grandement améliorer son hypothèse de classification. Malheureusement cette approche est difficile à réaliser en pratique et elle ne garantit pas que les éléments générés aient réellement en sens. Par exemple, si nous prenons le cas d'un système de reconnaissance de chiffre ; il pourrait très bien être emmené à générer l'image résultante d'une superposition du chiffre cinq et du chiffre six. Dans le deuxième cas, qui est le plus commun, les échantillons sont issue d'un ensemble très grand qui n'est pas annoté. Ces méthodes sont désignés sous le nom d'échantillonnage sélectif (ou « selective sampling »). L'important est alors de définir une stratégie pour sélectionner les éléments dans l'ensemble non annoté qui permettront à chaque itération d'améliorer l'hypothèse de classification. L'ensemble des stratégies proposées dans la littérature se décompose en deux sous-ensembles. Les premières méthodes, regroupées sous la désignation de sélection par comité (ou « query by committee »), font appel au désaccord régnant au sein d'un comité de systèmes de classification pour sélectionner les meilleurs éléments à soumettre à l'annotation (McCallum and Nigam [6], Freund et al. [4], Seung et al. [10]). Ces méthodes ont l'avantage d'utiliser les caractéristiques de généralisation de différents systèmes d'apprentissage et de réduire itérativement leur désaccord. Toutefois elles nécessitent de construire plusieurs modèles à chaque itération, ce qui peut devenir particulièrement lent. Les deuxièmes méthodes, regroupées sous la désignation de sélection par incertitude (ou « uncertainty sampling »), évaluent la qualité des estimations afin de choisir les éléments dont l'estimation est la moins fiable (Cohn et al. [2], Lindenbaum et al. [5]). Elles reposent uniquement sur un procédé de classification et une estimation de la qualité de l'estimation.

Les applications de l'apprentissage actif apparaissent actuellement dans le domaine de l'annotation de contenu multimédia (Tong and Chang [11], Zhang and Chen [12], Rong Yan [9]). La prochaine section introduit le cadre mathématique de l'apprentissage actif avec une stratégie de sélection par incertitude.

5.1.2 Sélection par incertitude

Dans ce chapitre, les notations introduites dans la section 3.2.1 à la page 56 sont conservées. Pour mémoire nous rappelons que $\mathcal{E}$ est l'espace des éléments à classifier $\{x_i\}$. $C = [-1,1]^{n_C}$ est l'ensemble des classes possibles. $\mathcal{L} = \{(x_i, c_i) \in \mathcal{E}x\mathcal{L}, i \in [1..n_\mathcal{L}]\}$ définit l'ensemble d'entraînement. Et C est la fonction de coût des erreurs. En premier lieu, nous plaçons notre étude dans le cas particulier où $n_C = 1$, qui est le plus répandu et le plus simple à traiter.

Le schéma de la figure 5.1 illustre le processus d'entraînement actif par échantillonnage sélectif.

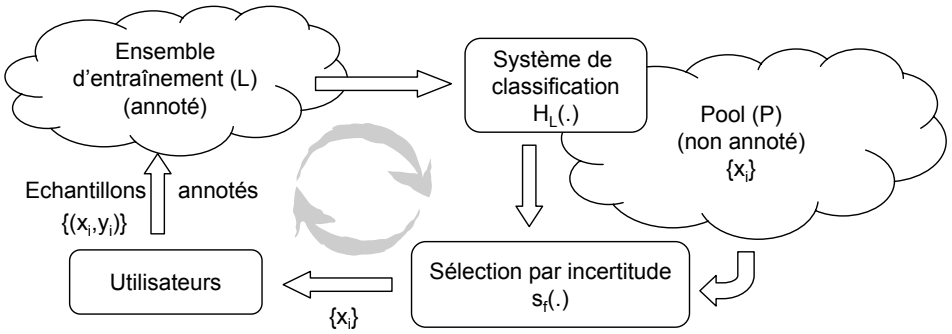


FIG. 5.1 – Principe de l'apprentissage actif par échantillonnage sélectif : *Méthode d'annotation itérative qui permet de réduire le nombre d'annotation à effectuer. Elle demande uniquement l'annotation des échantillons qui sont estimés les plus informatifs.*

L'algorithme débute à partir d'un petit ensemble d'entraînement $\mathcal{L}_0$. Puis il accroît progressivement cet ensemble initial de manière à construire la meilleure hypothèse. A chaque itération le système construit une hypothèse de classification $\mathcal{H}_{\mathcal{L}_i}$ en utilisant les données annotées $\mathcal{L}_i$. Désignons le grand ensemble (idéalement infini) non annoté par $\mathcal{P}$ pour pool. L'hypothèse permet alors d'évaluer la qualité des éléments de $\mathcal{P}$ et de sélectionner ceux qui devront être annotés. Les éléments sélectionnés par la fonction de sélection $s_{\mathcal{H}}(.)$ devront permettre d'améliorer au mieux l'hypothèse de classification qui sera calculé sur le nouvel ensemble d'entraînement. La difficulté réside dans le choix de la fonction $s_{\mathcal{H}}(.)$.

En théorie, le problème de l'apprentissage actif est de trouver à chaque itération l'ensemble à annoter $\mathcal{S}$ qui fournira la meilleure réduction de l'erreur généralisée, $R_{\mathcal{S}}$. C'est à dire :

$$\mathcal{S} = s_{\mathcal{H}}(P) = arg\max_{\mathcal{S}} R_{\mathcal{S}} \tag{5.1}$$

$$R_{\mathcal{S}} = \int_X (E_{Y|X}[C(\mathcal{H}_{\mathcal{L}}(x), y)] - E_{Y|X}[C(\mathcal{H}_{\mathcal{L}\cup\mathcal{S}}(x), y)])\, P(X)dx \tag{5.2}$$

Malheureusement, la pratique ne permet pas de résoudre le problème comme il est formulé et deux difficultés majeures sont identifiables. Tout d'abord les distributions $P(X)$ et $P(Y|X)$ ne sont pas connues. Ensuite il est impossible d'énumérer toutes les combinaisons possibles pour former S et encore moins de calculer la réduction de l'erreur pour tous les ensembles possibles. De nombreuses hypothèses devront être faites qui aboutiront à différentes stratégies de sélection.

Une première approximation consiste à transformer l'intégrale sur $\mathcal{E}$ en somme finie sur $\mathcal{P}$. $\mathcal{P}$ n'est pas annoté et nous pouvons donc supposer qu'il est choisi avec une taille suffisante pour représenter convenablement la distribution initiale. La réduction de l'erreur généralisée s'écrit alors :

$$\hat{R}_S = \sum_{\mathcal{P}} E_{Y|X}[C(\mathcal{H}_{\mathcal{L}}(x), y)] - E_{Y|X}[C(\mathcal{H}_{\mathcal{L}\cup\mathcal{S}}(x), y)] \tag{5.3}$$

Construire l'hypothèse pour tous les sous-ensembles de $\mathcal{P}$ devient le problème principal. Dans (Rong Yan [9]), les auteurs supposent tout d'abord que toutes les erreurs effectués sur $\mathcal{P} \setminus \mathcal{S}$ ont une

influence égale. La somme est alors réduite sur $\mathcal{S}$. Ensuite ils remarquent que le coût des erreurs réalisées par $\mathcal{H}_{\mathcal{L}\cup\mathcal{S}}$ est négligeable devant celles réalisées par $\mathcal{H}_{\mathcal{L}}$. Puis ils notent qu'il est possible de choisir un seul élément à chaque fois puisque le procédé est itératif. Finalement le modèle du meilleur cas est utilisé pour estimer y, ce qui donne :

$$s_{\mathcal{H}}(\mathcal{P}) = arg\max_{x\in\mathcal{P}} C(\mathcal{H}_{\mathcal{L}}(x), \hat{y}) \tag{5.4}$$

$$\hat{y} = arg\min_{y\in C=[-1,1]} C(\mathcal{H}_{\mathcal{L}}(x), y) \tag{5.5}$$

L'idée derrière cette fonction de sélection est de choisir les éléments les plus ambiguës à chaque itération. Ainsi les estimations sont de plus en plus tranchées. La plupart des système d'apprentissage actif utilisant une sélection par incertitude effectuent implicitement les approximations précédentes pour proposer une fonction de sélection similaire à l'équation 5.4.

5.1.3 Système de classification

Nous avons concentré cette étude sur le système de classification par les k plus proches voisins. Il a la particularité de subir uniquement des changements locaux avec l'évolution de l'ensemble d'entraînement. La mise à jour des estimations à chaque itération sera donc restreinte à un petit sous-ensemble de $\mathcal{P}$. Nous rappelons que dans ce cas l'hypothèse $\mathcal{H}_{\mathcal{L}}$ est définie par :

$$\mathcal{H}_{\mathcal{L}}(x) = \frac{\sum_{v_i\in V(x)} c_j K(x, v_i)}{\sum_{v_i\in V(x)} K(x, v_i)} \tag{5.6}$$

avec $V(x)$ le voisinage du point x dans $\mathcal{L}$

et $K(x, v_i) = cos(x, v_i)$

Et la fonction de coût des erreurs de classification est définie par :

$$C(x, y) = \|x - y\| \tag{5.7}$$

La partie suivante présente les données qui seront utilisées et les résultats préliminaires.

5.2 Résultats préliminaires

Cette section a pour objectif d'évaluer l'intérêt de l'apprentissage actif. Dans cette optique, les données utilisées pour démontrer l'efficacité des algorithmes sont présentées. Ensuite les résultats préliminaire seront détaillés.

5.2.1 Les données

Nous allons continuer d'utiliser la base de données fournie par TRECVID et son annotation. Les plans des vidéo sont représentés par leur couleur et leur texture en suivant la méthode proposée dans la section 2.2 à la page 27, c'est à dire l'analyse de la sémantique latente. Nous avons fait le choix de fusionner par concaténation les caractéristiques de couleur et de texture avant la classification. Cette solution permet de s'affranchir du délicat problème poser par la fusion. Les concepts « studio » et

« climat » ont été choisis dans les expériences pour deux raisons. La première est leur présence au niveau de l'ensemble de l'image. Le problème de la classification est déjà ardu et il est inutile de la compliquer en essayant de détecter des objets. La seconde est leur répartition dans l'ensemble d'entraînement. Le concept « climat » apparaît 262 fois contre 2 990 pour le concept « studio ». Ce qui permet d'étudier le comportement de concepts fréquents et plus rares.

Toutefois la tâche de classification sur des données réelles est très délicate et elle ne permet pas d'analyser objectivement le comportement des systèmes de classification actifs. Pour cela, nous avons introduit deux ensembles synthétiques composés de deux classes. Les ensembles sont construits à partir de deux cent groupes dont la position est tirée aléatoirement. Les vecteurs à deux dimensions de ces groupe sont ensuite tirés aléatoirement selon une distribution normale. La figure 5.2 illustre la structure des deux ensembles générés. Comme nous pouvons le voir, ils se distinguent par la séparabilité des classes. L'avantage d'utiliser des ensembles synthétiques réside dans



(a) 1er ensemble (b) 2ème ensemble

FIG. 5.2 – Ensembles synthétiques générés pour étudier le comportement de l'apprentissage actif. *Le premier ensemble est composé de classes bien distinctes alors qu'elles sont plus entrelacées dans le second.*

la connaissance de leur structure. Ainsi, nous savons qu'il y a dans le pire des cas, cent groupes d'une classe et cent groupe de l'autre classe en alternance. Intuitivement six échantillons connus par groupe permettent déjà d'obtenir un très bon système de classification puisque le voisinage est composé de dix voisins.

5.2.2 Evaluation des systèmes d'apprentissage actif

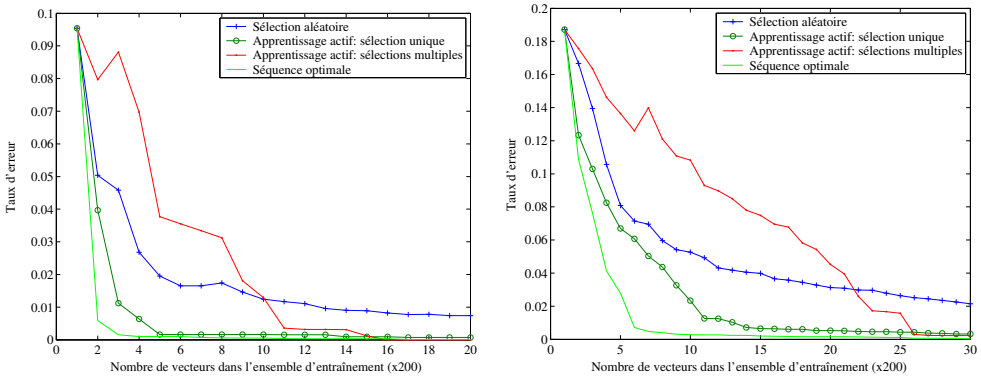
L'évaluation des systèmes d'apprentissage actifs consiste à étudier l'évolution du taux d'erreur en fonction de la taille de l'ensemble d'entraînement. A chaque itération, ce dernier croît de manière à fournir une information supplémentaire au système de classification. L'amélioration obtenue grâce à cette nouvelle information permet d'évaluer la qualité de la fonction de sélection. Un système actif performant atteint rapidement des taux d'erreur faibles en effectuant à chaque itération la meilleure sélection.

Dans l'absolu, il est impossible d'évaluer les performances sans point de repère. Pour cela les bornes inférieure et supérieure sont estimées pour servir de référence. La borne inférieure est fournie par une approche sélectionnant aléatoirement les échantillons qui seront annotés. Le principe de sélection intelligente des échantillons est alors supprimé et les hypothèses construites à chaque itération sont indépendantes. La borne supérieure peut être approchée en connaissant les annotations du pool. Dans ce cas, une maximisation gloutonne (« greedy maximization ») est appliquée. Malheureusement, le calcul de la borne supérieure est très long et il ne pourra pas être conduit à terme sur la base de données fournie par TRECVID.

Le calcul du taux d'erreur requière la connaissance des classes de tous les échantillons. Pour cette raison, les expériences sont effectuées sur la base de données de TRECVID et deux ensembles synthétiques. L'intervention des utilisateurs n'est heureusement pas nécessaire pour évaluer les systèmes présentés. L'apprentissage actif débute avec un ensemble d'entraînement initial qui servira à construire la première hypothèse. A chaque itération des échantillons du pool sont sélectionnés en fonction de la stratégie employée. Un utilisateur virtuel, c'est à dire le système lui-même, attribue les classes correctes à chaque échantillons. Ces derniers sont finalement ajoutés à l'ensemble d'entraînement avec leur nouvel annotation. Dans une application réelle, l'utilisateur virtuel est remplacé par un ou plusieurs utilisateurs qui auront la pénible tâche d'effectuer l'annotation.

5.2.3 Expériences

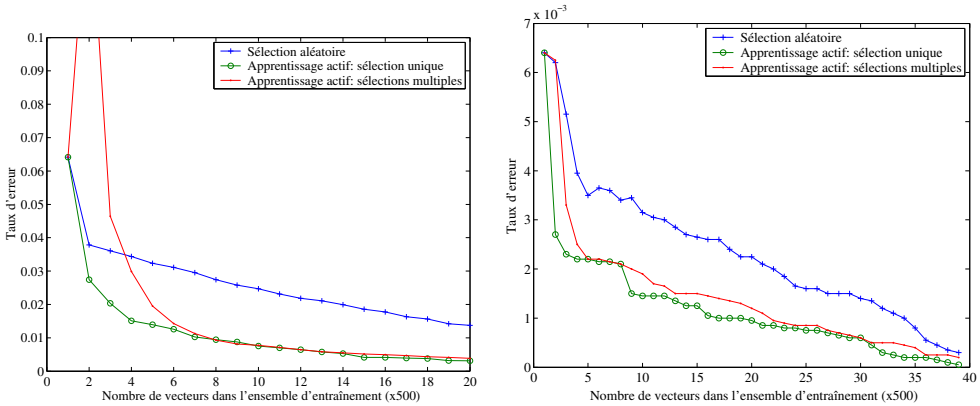
Les expériences préliminaires présentées dans cette partie ont deux objectifs. Le premier est de mettre en valeur l'apport de l'apprentissage actif : les hypothèses peuvent être efficacement construites à partir d'un ensemble d'entraînement réduit. Le second est de mettre en évidence le point faible des approches classiques qui supportent mal la sélection de plusieurs échantillons par itération : lorsque le nombre d'échantillons sélectionnés par itération croît, les performances décroissent. Les figures 5.3(a) et 5.3(b) montrent l'évolution du taux d'erreur de quatre expériences. La première expérience est la borne inférieure, la seconde et la troisième sont le fruit de l'apprentissage actif avec respectivement un et deux cents échantillons sélectionnés par itération, la dernière est la borne supérieure. Notons que les abscisses ne correspondent pas au nombre d'itérations mais elles indiquent la taille atteinte par l'ensemble d'entraînement. Ceci permet de comparer les approches en fonction du critère le plus important qui est le nombre d'échantillons à annoter. Toutefois le nombre d'itérations est également une donnée importante puisqu'il s'agit du nombre d'interventions de l'utilisateur et d'hypothèses construites. Avant de se lancer dans une comparaison précise du comportement des systèmes, remarquons tout d'abord qu'ils sont similaires sur les deux ensembles de données. Comme nous pouvions nous y attendre, le second ensemble requière plus d'éléments annotés pour effectivement discerner les deux classes présentes. Mais relativement les comportements



(a) 1er ensemble (b) 2e ensemble

FIG. 5.3 – Apprentissage actif sur les ensembles synthétiques.

des quatre approches sont identiques à ceux obtenus sur le premier ensemble et nous présentons uniquement l'étude sur ce dernier. Commençons par comparer les deux premières approches qui sont la borne inférieure et l'apprentissage actif avec une sélection unitaire. Les deux courbes se rejoignent parfaitement lorsque l'ensemble d'entraînement atteint une taille de 10 000 échantillons, le taux d'erreur est alors de 0%. Cette partie de la courbe n'est pas illustré puisque la partie intéressante se situe en amont. Notamment lorsque la taille de l'ensemble d'entraînement atteint 1 200 échantillons. En ce point, l'apprentissage actif fournit un taux d'erreur avoisinant déjà 0% tandis que la borne inférieure fournit un taux d'erreur de 2%. Une analyse automatique et aveugle du pool permet donc d'effectuer une sélection progressive et intelligente aboutissant à un ensemble d'entraînement de taille significativement réduite. Une comparaison avec la borne supérieure montre que le système d'apprentissage actif présenté parvient à une séquence de sélections s'approchant de la séquence optimale. La borne supérieure atteint un taux d'erreur avoisinant les 0% pour 600 échantillons dans l'ensemble d'entraînement. Seulement la moitié des points nécessaire dans le pire des cas sont utilisés pour construire un système de classification efficace. Remarquons qu'en utilisant le deuxième ensemble ce nombre monte à 1 200 ce qui correspond à l'ordre de grandeur du nombre de points nécessaire dans le pire des cas. Concluant cette étude sur les ensembles synthétiques en analysant le comportement du système lorsque plusieurs échantillons sont sélectionnés simultanément à chaque itération. Il faut annoter environ deux fois plus d'échantillons pour arriver aux mêmes performances que l'apprentissage actif avec une sélection unitaire. L'intérêt de l'approche est alors fortement pénalisé par cet excès d'annotation à réaliser. Ce phénomène a déjà été observé sans être analysé par Rong Yan [9]. Intuitivement les échantillons sélectionnés ont beaucoup de chance d'être proches et similaires. Cette répétition induit la redondance d'information qui ralentit l'évolution du système. Cela explique pourquoi les performances peuvent même être inférieure à celles de la borne inférieure. Les figures 5.4(a) et 5.4(b) montrent les mêmes expériences sur des données réelles. Malheureusement la borne supérieure est trop longue à calculer et elle fera défaut. Les comparaisons seront donc faites relativement à la borne inférieure. Le concept « climat » étant rare, le taux d'erreur est par défaut très faible ce qui explique la différence d'échelle utilisée entre les deux figures. La rareté du concept explique aussi l'évolution en escalier des courbes. Les changements bruts



(a) Concept « studio » de TRECVID (b) Concept « climat » de TRECVID

FIG. 5.4 – Apprentissage actif sur deux concepts sémantique de TRECVID.

correspondent à l'introduction de plans contenant le concept. Relativement à la borne inférieure, l'apprentissage actif est modérément efficace dans les deux cas. Pour une taille donné de l'ensemble d'entraînement, les performances de la borne inférieure sont systématiquement plus faible. La différence de performance est la plus grande pour une faible taille de l'ensemble d'entraînement, ensuite elle décroît progressivement. En fonction du taux d'erreur acceptable, l'apprentissage actif apporte un avantage non négligeable notamment sur les premières itérations. Par contre la sélection simultanée de plusieurs échantillons dégrade les performances sur les premières itérations.

Les stratégies traditionnelles de sélection des échantillons ne sont pas adaptées à des sélections multiples. La prochaine partie présente une nouvelle méthode de sélection des échantillons qui offre la possibilité d'effectuer des sélections multiples sans dégradation des performances.

5.3 Sélection par partitionnement

Cette section décrit notre approche pour effectuer une sélection efficace de plusieurs échantillons. Dans un premier temps, l'approche mathématique est décrite et dans un deuxième temps l'implémentation retenue est présentée.

5.3.1 Théorie

Le problème de la sélection multiple réside principalement dans la redondance d'information. Les algorithmes de classification sont étroitement liés à la notion de voisinage et de similarité entre les échantillons. Généralement les points proches appartiennent à la même classe et leur valeurs de la fonction de coût sont proches. Par conséquent un point sélectionné pour son aptitude à apporter une valeur ajoutée forte aura probablement ses proches voisins sélectionnés également. Or il apparaît évident que cette sélection groupée doit être évité pour minimiser l'effort d'annotation et surtout pour ne pas demander l'ennuyeuse annotation de plans similaires. Il apparaît beaucoup plus intéressant de choisir des échantillons correctement distribués dans leur espace. Cette notion

de redondance d'information dans le voisinage des échantillons est implicitement utilisée dans de nombreux algorithmes d'apprentissage actif, en particulier par Zhang and Chen [12] qui proposent de pondérer les valeurs de la fonction de sélection par une estimation de la densité de probabilité au voisinage des points.

Dans l'optique d'éviter une sélection groupée des échantillons, Ferecatu et al. [3] ont présenté une solution au problème par filtrage des échantillons sélectionnés. Ainsi, l'ensemble des échantillons similaires sont retirés de la sélection pour obtenir un sous-ensemble d'échantillons plus hétérogènes.

Nous proposons ici de garantir que la plupart des points sélectionnés sont éloignés et potentiellement à grande valeur ajoutée directement dans le processus de sélection. Pour cela supposons qu'une partition $P = \cup U_i, i = 1..p$ est construite telle que $U_i \cap U_j = \emptyset$ pour $i \neq j$ et U_i connexes. Choisissons un élément représentatif de chaque partition qui sera noté m_i. Il peut être défini par l'élément moyen ou l'élément dont la valeur ajoutée potentielle est la plus élevée. L'équation 5.3 s'écrit alors :

$$\hat{R}_S = \sum_{i=1}^{p} \left(\sum_{x_i \in U_i} (E_{Y|X}[C(\mathcal{H}_{\mathcal{L}}(x_i), y_i)] - E_{Y|X}[C(\mathcal{H}_{\mathcal{L}\cup\mathcal{S}}(x_i), y_i)]) \right) \tag{5.8}$$

$$\hat{R}_S \approx \sum_{i=1}^{p} (E_{Y|X}[C(\mathcal{H}_{\mathcal{L}}(m_i), y_i)] - E_{Y|X}[C(\mathcal{H}_{\mathcal{L}\cup\mathcal{S}}(m_i), y_i)])N_i \tag{5.9}$$

Où N_i est le cardinal de U_i. Cette approximation repose sur l'hypothèse que les échantillons d'un groupe ont un comportement similaire vis à vis des hypothèses construites. En suivant un raisonnement similaire au cas précédent et en posant $M = \{m_i\}$, nous aboutissons à :

$$s'_f(P) = arg \max_{S \subset M} \sum_{x_i \in S} E_{Y|X}[C(\mathcal{H}_{\mathcal{L}}(x_i), y_i)]N_i \tag{5.10}$$

L'idée est de sélectionner les échantillons qui ont potentiellement une grande valeur ajoutée et qui sont répartis sur l'ensemble de la distribution.

5.3.2 Implémentation

En pratique, la partition du pool est construite à l'aide d'un algorithme de regroupement. Idéalement la partition doit être mise à jour à chaque itération pour prendre en compte les changements topologiques du pool. Nous verrons dans la partie expérimentale s'il est nécessaire de construire la partition à chaque itération ou si elle peut être calculée uniquement lors de l'initialisation. Nous en profiterons pour étudier l'impact de la taille de la partition. Par mesure de simplicité et de rapidité, l'algorithme de k-means est employé pour déterminer le partitionnement de manière aveugle.

Une fois la partition créée sur le pool, l'équation 5.4 est utilisée pour choisir les éléments représentatifs de chaque ensemble de la partition. Au final, $\mathcal{S}$ est composé des meilleurs représentants.

5.4 Evaluation

Les résultats expérimentaux sont présentés dans cette section. Tout d'abord la première partie traite du problème du partitionnement. La seconde partie compare les performances obtenue entre

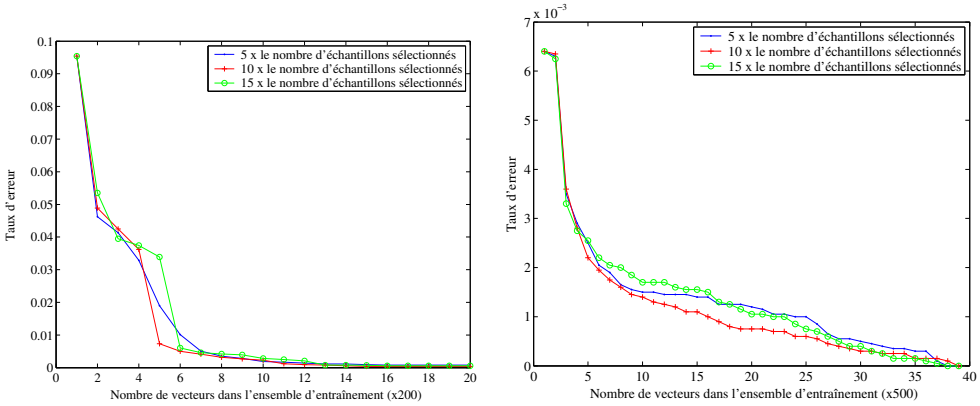
une stratégie de sélection par incertitude et une stratégie de sélection par partitionnement.

5.4.1 Evaluation du partitionnement

L'approche proposée pour déterminer le partitionnement est l'algorithme de k-means. Comme la plupart des algorithmes de regroupement, le nombre de groupe doit être spécifié. Dans les expériences qui vont suivre la taille des partitions est déterminée relativement au nombre d'échantillons à sélectionner. Pour cela, le facteur de partitionnement f est défini tel que :

$$\text{(the size of the partition)} = f \times \text{(the number of selected samples)} \tag{5.11}$$

La figure 5.5 montre l'influence de ce facteur sur les performances de l'apprentissage actif. Les



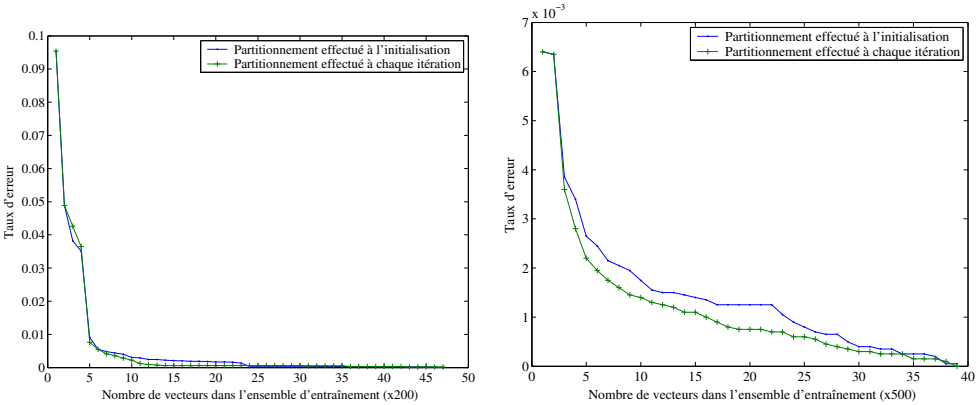
(a) 1er ensemble (b) Concept « climat » de TRECVID

FIG. 5.5 – Influence du nombre de partitions sur les performances de l'apprentissage.

variations de performances sont faibles et le facteur peut raisonnablement être fixé à dix. Une sélection empirique dans la fourchette $[5, 15]$ n'affecte pas dramatiquement les performances ce qui est primordial à noter pour effectuer le choix sur d'autres problèmes. La figure 5.6 montre la l'évolution des performances lorsque la partition est créée une fois à l'initialisation et lorsqu'elle est mise à jour à chaque itération. Sans surprise, la deuxième approche fournit de meilleures performances. Toutefois pour certain problème, et en particulier le nôtre, les temps de calculs supplémentaire à fournir ne justifient pas ce gain. Les systèmes demandant une forte précision peuvent adopter un compromis en effectuant une mise à jour de la partition que toutes les N itérations.

5.4.2 Evaluation de la sélection

Cette partie est dédiée à l'évaluation de la nouvelle stratégie de sélection par partitionnement. Les expériences préliminaires présentées dans la section 5.2.3 sont reprises et comparées avec notre nouvelle stratégie. Les figures 5.7(a) et 5.7(b) donnent l'évolution du taux d'erreur en fonction de la taille de l'ensemble d'entraînement pour chaque système. Comme nous l'attendions, la stratégie introduite permet effectivement de limiter la baisse de performance dû à une sélection de multiples



(a) 1er ensemble (b) Concept « climat » de TRECVID

FIG. 5.6 – Etude de la nécessité d'effectuer le partitionnement à chaque itération.

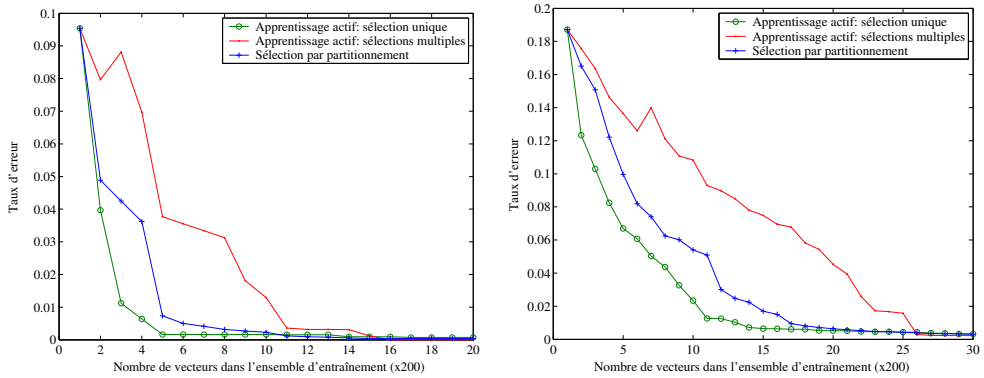
échantillons. De plus les performances obtenues sont proches de leur optimale qui est donné par un stratégie de sélection unitaire. Le cadre mathématique présenté dans la section précédente est donc bien adapté au problème et constitue une bonne solution.

Les figures 5.8(a) et 5.8(b) permettent d'aboutir aux mêmes conclusions sur les données réelles mais de manière plus modérée. L'évolution du taux d'erreur de la nouvelle stratégie se situe effectivement sous l'évolution du taux d'erreur pour la stratégie de sélection d'échantillons multiples.

5.5 Annotation de plusieurs classes

En pratique il est très rare d'effectuer l'annotation uniquement sur deux classes. Nous avons donc porté notre attention sur le problème de l'apprentissage actif lors d'une annotation multiple. Dans la littérature, nous trouvons que très peu de documents traitant de l'apprentissage actif avec plusieurs classes (Rong Yan [9], Zhang and Chen [12]). La solution de base est de calculer une mesure de sélection moyenne sur l'ensemble des systèmes de classification binaire. Allwein et al. [1] ont présenté et étudié plusieurs approches pour réduire un problème de classification multi classe en problèmes binaires. Le choix s'est porté sur des classifications binaires une-contre-tous (« once-against-all »). Un système de classification binaire est construit par classe. Cette méthode était implicitement employée dans les chapitres précédent. Les systèmes d'apprentissage actifs sur plusieurs classes souffrent également d'une réduction de leur performance lors d'une sélection multiple. Il s'avère donc particulièrement intéressant d'étudier le comportement de la nouvelle stratégie présentée dans la section 5.3 dans cette situation délicate.

Le système de classification par les k plus proches voisins présenté dans la partie 3.3.2 est facilement généralisable à l'estimation de plusieurs classes. De plus la complexité de l'algorithme est très peu affectée par cette généralisation puisque le voisinage ne dépend pas des classes en jeu. Soit K le nombre de classe et $y_i = \{y_i^k\}, k = 1, .., K \in [-1, 1]^K$ les classes associées au plan x_i. Pour



(a) First synthetic data set (b) Second synthetic data set

FIG. 5.7 – Apprentissage actif avec une stratégie par incertitude et partitionnement sur les ensembles synthétiques. *Comparison des performances.*

la classe k, l'hypothèse est définie par :

$$\mathcal{H}^k_{\mathcal{L}}(x_i) = \frac{\sum_{N_s} sim(x_i, n_j) * y^k_{n_j}}{\sum_{N_s} sim(x_i, n_j)}$$
$$\text{where } sim(x_i, n_j) = cos(x_i, n_j)$$

L'estimation des classes de x_i est alors donnée par :

$$\hat{y}^k_i = arg\min_y C(\mathcal{H}^k_{\mathcal{L}}(x_i), y^k)$$
$$C(x_i, y) = \|x_i - y\|$$

En considérant la participation de chaque système de classification binaire, l'équation 5.4 est généralisée sous la forme :

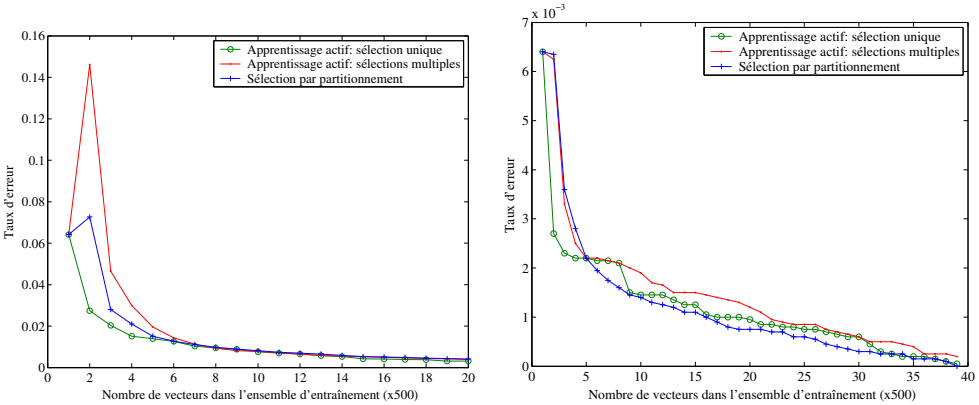
$$s_f(P) = arg\max_{x \in S} \sum_{y^k} C(f^k_L(x), \hat{y}^k) \tag{5.12}$$

Et pour l'équation 5.10, la généralisation donne :

$$s_f(P) = arg\max_{S \subset M} \sum_{S} \sum_{y^k} C(f^k_L(x_i), \hat{y}^k) N_i \tag{5.13}$$

La figure 5.9 montre le comportement du système lorsque plusieures classes interviennent dans le processus d'annotation. Ces premiers résultats sont obtenus sur l'ensemble de données artificielles. L'apprentissage actif permet de diminuer significativement la taille de l'ensemble d'entraînement et la sélection par partitionnement permet de conserver cette avantage même lors d'une sélection multiple par itération.

La figure 5.10 montre le comportement du système sur des données réelles. Nous observons en particulier une nette baisse de l'utilité de l'apprentissage actif présenté. Ce problème est principalement lié à la decorrélation entre les classes qui aboutit à une sélection proche d'un tirage



(a) Studio TRECVID feature (b) Weather TRECVID feature

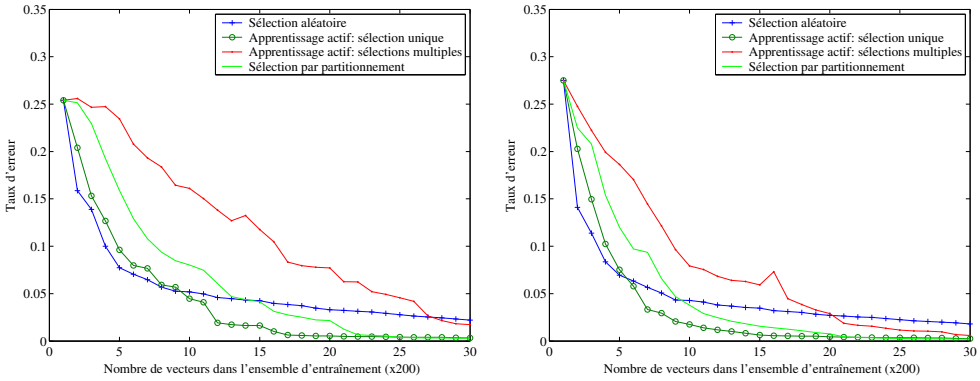
FIG. 5.8 – Apprentissage actif avec une stratégie par incertitude et partitionnement sur les vidéos de TRECVID. *Comparison des performances.*

aléatoire. En effet, l'information fournie par l'annotation d'une classe est souvent inutile aux autres classes. Ainsi l'entraînement simultané des concepts *studio* et *climat* permet encore de limiter l'effort d'annotation par apprentissage actif. Par contre l'introduction de trois classes supplémentaires *végetation*, *visage* et *violence* réduit grandement les bénéfices de ce dernier. L'annotation de presque tous les plans est nécessaire.

5.6 Conclusion

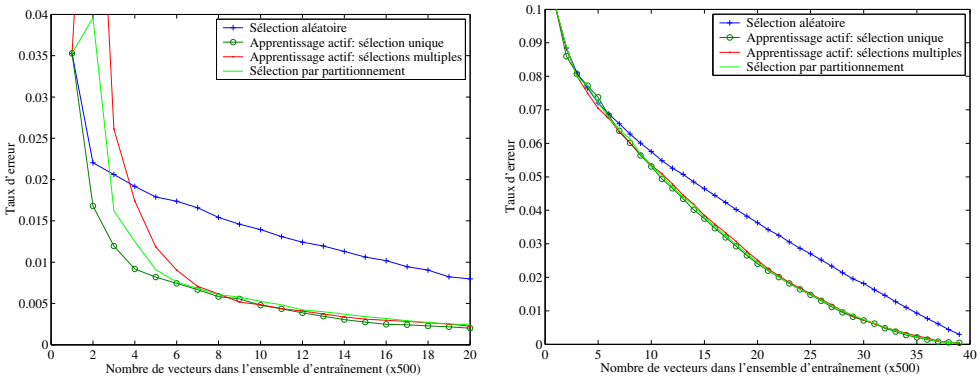
Ce chapitre a présenté les méthodes d'apprentissage actif pour réduire le nombre d'échantillons à annoter. En particulier dans le cadre d'une annotation à grande échelle, nous avons introduit une nouvelle stratégie de sélection des échantillons qui permet une sélection multiple sans une grande atteinte aux performances de l'apprentissage actif. Les expériences ont été effectuées à la fois sur des données synthétiques et sur un problème réel et délicat de classification de plans vidéos. Les données synthétiques ont permis de valider notre approche et de démontrer son intérêt. Ensuite l'étude sur un problème réel a soulevé de nombreux problèmes dus à la difficulté de réaliser la classification correctement.

Dans un second temps nous avons étendu notre approche au problème de l'annotation simultanée de plusieurs classes. En effet, il est plus judicieux de faire directement l'annotation sur plusieurs classes que de relancer la procédure pour chaque concept. Une perte notable des performances est observée. Nous pensons qu'elle est principalement liée au manque de corrélation entre les classes. Dans ce cas, la sélection se rapproche d'un tirage aléatoire. La difficulté de la tâche de classification est également responsable puisque sur l'ensemble synthétique l'apprentissage actif atteint toujours des performances correctes. Le problème peut venir de nombreuses sources dont la représentation du contenu.



(a) Problème à deux classes (b) Problème à cinq classes

FIG. 5.9 – Apprentissage actif sur un ensemble synthétique et sur un problème multi classes.



(a) Problème à deux classes (b) Problème à cinq classes

FIG. 5.10 – Apprentissage actif sur la base de données TRECVID et sur un problème multi classes.

Bibliographie

[1] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary : A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1 :113–141, 2000.

[2] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4 :129–145, 1996.

[3] M. Ferecatu, M. Crucianu, and N. Boujemaa. Reducing the redundancy in the selection of samples for SVM-based relevance feedback. Technical Report 5258, INRIA, may 2004.

[4] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28 :133–168, 1997.

[5] Michael Lindenbaum, Shaul Markovitch, and Dmitry Rusakov. Selective sampling for nearest neighbor classifiers. *Machine Learning*, 54(125-152), February 2004.

[6] Andrew McCallum and Kamal Nigam. Employing em and pool-based active learning for text classification. In *Proceedings of the International Conference on Machine Learning*, 1998.

[7] Ion Muslea, Steve Minton, and Craig Knoblock. Active + semi-supervised learning = robust multi-view learning. In *Proceedings of the International Conference on Machine Learning*, pages 435–442, 2002.

[8] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3) :103–134, 2000.

[9] Alexander Hauptmann Rong Yan, Jie Yang. Automatically labeling video data using multi-class active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2003.

[10] H.S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, 1992.

[11] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 107–118, 2001.

[12] Cha Zhang and Tsuhan Chen. An active learning framework for content-based information retrieval. In *IEEE Transactions on Multimedia*, volume 4, pages 260–268, 2002.

Chapitre 6

Conclusion

Dans ce mémoire nous avons abordé de nombreux problèmes posés par la constitution des systèmes d'indexation et de recherche de plans vidéo par le contenu. Les solutions proposées ont été évaluées dans le cadre de la recherche d'objets sur une vidéo et dans le cadre de la recherche de plans vidéo sur une grande base de données de nouvelles télévisées. Le mémoire a suivi l'ordre logique nécessaire à l'élaboration et à la compréhension d'un tel système.

La prochaine section résume les thèmes abordés et les méthodes retenues ou proposées. Ensuite nous explorerons les perspectives de recherche mises en valeur par ce travail de thèse. En particulier, des travaux préliminaires sur l'intégration de la structure des images pour l'indexation seront présentés.

6.1 Résumé

Le premier chapitre introductif a rapidement mis en place le cadre des travaux de la thèse et présenté les diverses contributions qui ont été apportées.

Le second chapitre a tout d'abord introduit le fonctionnement des systèmes d'indexation et de rechercher d'information par le contenu. Un tel système débute par une étape d'analyse du contenu visuel. L'objectif est d'extraire des signatures qui permettent de représenter le contenu visuel de manière efficace. L'efficacité est déterminée par la capacité d'une signature à être discriminante, compacte et fiable. La présentation du support numérique de la vidéo et la synthèse des techniques existantes d'indexation par le contenu visuel ont alors été faites.

Le troisième chapitre a développé le principe de l'analyse de la sémantique latente du contenu visuel. Cette méthode d'indexation destinée à l'origine aux documents écrits a été adaptée dans ce chapitre à l'indexation du contenu visuel. Elle a l'avantage de permettre une description efficace du plan vidéo par l'intermédiaire des régions le composant. Les résultats très encourageants de cette méthode nous ont conduits à approfondir le modèle en incluant le contenu présent à plusieurs échelles de segmentation mais aussi dans plusieurs images du plan. Le chapitre a finalement conclut sur une étude de la boucle d'asservissement. Elle permet au système d'améliorer ses résultats en utilisant un retour d'information de l'utilisateur sur le dernier résultat de la recherche.

Le quatrième chapitre aborde le thème délicat et récent de l'indexation automatique par des mots clefs. L'objectif est de détecter de manière automatique l'ensemble des concepts qui sont

présents dans un plan. Pour cela nous utilisons les méthodes de classification et trois méthodes représentatives ont été étudiées.

Le cinquième chapitre poursuit l'étude des systèmes de classification en se penchant sur le problème de la fusion. La fusion est essentielle pour combiner efficacement plusieurs modalités ou systèmes de classification. Nous avons profité de cette étude pour intégrer de nouvelles caractéristiques du plan comme le texte et le mouvement. L'inclusion du texte a requis un traitement particulier pour pallier à la désynchronisation sémantique qui existe entre les modalités visuelles et textuelles.

L'entraînement de systèmes de classification nécessite une grande quantité de données avec leur annotation. Ce travail manuel est pénible, long et sujet aux erreurs d'appréciation. Ce sixième chapitre a étudié les systèmes d'apprentissage actif qui permettent de réduire le nombre d'annotation à faire. L'apprentissage actif est un procédé itératif qui permet de suggérer les échantillons qui doivent être annotés pour améliorer au mieux la classification. Cependant l'ensemble de ces systèmes souffre d'une perte d'efficacité dès lors que plusieurs échantillons sont annotés simultanément et nous avons proposé une solution à ce problème à l'aide d'une sélection par partitionnement. L'approche a ensuite été étendue au cas de l'annotation de plusieurs classes.

6.2 Perspectives

Le travail de ce mémoire de thèse peut se poursuivre dans plusieurs directions. Nous retiendrons en particulier l'amélioration de la capture du contenu visuel et la fusion d'information. Nous avons vu que les caractéristiques visuelles ont une importance prédominante et dès le début il est important d'obtenir des signatures complètes et efficaces. Un défaut de la méthode que nous avons introduite est la perte de toute information sur l'agencement des régions. Nous détaillerons donc nos travaux préliminaires pour limiter cette perte. Ensuite nous présenterons des caractéristiques complémentaires aux régions sur lesquelles l'analyse de la sémantique latente peut aussi être conduite. Finalement la fusion est une étape primordiale pour combiner efficacement les sources d'information multiples et nous verrons les problèmes qui restent encore à étudier pour pleinement tirer profit de toute l'information disponible.

6.2.1 Amélioration de la capture de l'information visuelle

Dans un premier temps, une méthode d'indexation reposant sur l'analyse de l'agencement des régions est présentée, ensuite nous présentons les points remarquables comme caractéristiques complémentaires aux régions pour décrire le contenu visuel.

Analyse de l'agencement des régions

La description du contenu des plans par les régions offre l'avantage de capturer l'information locale tout comme le système visuel humain. Toutefois la représentation que nous avons adoptée élimine toutes relations entre les régions et uniquement la présence ou l'absence de régions particulières intervient. Cette approche efficace est alors limitée par la perte de l'information spatiale sur les régions. Une image d'échiquier est similaire à deux bandes de couleur noir et blanc. Ce problème n'est pas propre à notre représentation du contenu comme nous avons pu le voir dans la section 1.4

à la page 13. Et très peu de systèmes sont parvenus à exploiter efficacement l'organisation spatiale des régions dans un cadre général. L'utilisation de contraintes spatiales par l'intermédiaire d'un graphe d'adjacence des régions permet d'obtenir une description plus riche et plus discriminative du contenu. Toutefois les graphes d'adjacence augmentent significativement la complexité de l'indexation et de la recherche, et pour ces raisons peu de travaux les ont utilisés pour indexer de grandes quantités de vidéo (Shearer et al. [8], Huet and Hancock [4], Sengupta and Boyer [7], Messmer and Bunke [5]). Nous proposons tout d'abord une extension du système de quantification ou de regroupement pour opérer sur une région et ses voisins. Ensuite nous proposons une approche qui prend en compte les relations entre les paires de régions, où une paire est définie par deux régions voisines. Ces paires sont ensuite considérées comme l'entité de base à étudier. Nous allons voir comment l'espace vectoriel d'image et l'analyse de la sémantique latente sont adaptés à cette nouvelle forme de région.

Le voisinage peut être défini de plusieurs manières. Il peut s'agir du voisinage naturel, c'est à dire deux régions sont dites voisines si elles ont une frontière commune. Ce type de voisinage a l'inconvénient d'être sensible aux frontières. Le voisinage peut être défini par l'ensemble des régions dont les barycentres sont les plus proches. Ce type de voisinage est plus robuste aux changements des frontières (Huet and Hancock [4]), par contre il est cohérent seulement en présence de régions approximativement circulaires. En effet les régions de forme oblongue ou complexe perdent d'importants voisins au niveau de leurs extrémités dont les voisins naturels sont éloignés du barycentre. Dans cette étude préliminaire nous avons considéré uniquement le deuxième type de voisinage.

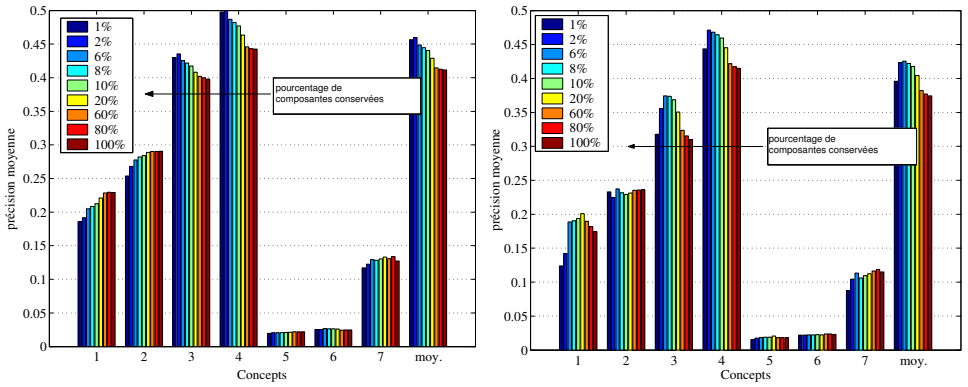
La première approche que nous explorons pour prendre en compte le voisinage effectue la quantification sur une région et ses voisins. Une nouvelle mesure de similarité est alors proposée. Soit R une région et $\phi^R = \{V_i^R\}$ ses voisins. Soit Q la deuxième région et $\phi^Q = \{V_j^Q\}$ ses voisins. Supposons que $card(\phi^R) \leq card(\phi^Q)$ (dans le cas contraire il suffit d'intervertir R et Q), la distance entre R et Q est donnée par :

$$D(R,Q) = L_2(R,Q) + \frac{1}{card(\phi^R)} \sum_{i \in \phi_R} \min_{j \in \phi_Q} L_2(V_i^R, V_i^Q) \tag{6.1}$$

Les vecteurs de quantification formés par les régions et leurs voisins forment un nouveau dictionnaire. La représentation par l'espace vectoriel d'image et l'analyse de la sémantique latente sont ensuite adoptées pour l'indexation et la recherche. Les résultats obtenus par cette approche sont décevants. La contrainte imposée par les régions et leurs voisins est trop forte.

La deuxième approche que nous avons explorée travaille sur les paires de régions. Au lieu d'étudier l'occurrence des régions dans une image, l'occurrence des paires de régions est étudiée. Le dictionnaire visuel présenté dans la section 2.1.3 à la page 26 est remplacé par toutes les paires possibles composées des mots-clefs visuels initiaux. Elles forment alors le dictionnaire relationnel. Pour un dictionnaire d'une taille initiale de k, le dictionnaire relationnel à une taille de $\frac{k \cdot (k+1)}{2}$. La taille explose donc rapidement. Cependant, nous avons remarqué que toutes les paires n'étaient pas présentes et il est possible de réduire significativement la taille du dictionnaire relationnel en retirant les paires rares.

La figure 6.1 compare l'approche initiale à gauche avec notre nouvelle approche à droite. Nous observons une baisse des performances lorsque les régions sont étudiées par paire. A présent nous devons étudier plus en détail les contraintes imposées par le voisinage des régions pour mieux



(a) Sans structure

(b) Avec les paires de régions

FIG. 6.1 – Comparaison de l'approche sans structure et de l'approche avec structure sur les vidéos de nouvelles de TRECVID

comprendre ce phénomène. En effet sur la série de dessins animés les résultats d'une indexation par la structure apporte une amélioration pour la moitié des personnages et une dégradation pour les autres. La structure est un paramètre important qu'il est difficile d'utiliser correctement.

Points remarquables

Les régions capturent le contenu par analyse des frontières entre les ensembles de points homogènes. Elles sont très sensibles aux changements de luminosité, aux mouvements et aux déformations. Un autre type de région peut être construit sur les points remarquables ou les coins d'une image. Suite à la détection de points remarquables dans l'image, les caractéristiques de ses points sont extraites (Gouet and Boujemaa [2]) ou des régions sont construites autour. Sivic and Zisserman [9] présentent un approche qui utilise les points remarquables pour construire un espace vectoriel d'image. Leur cadre entre parfaitement dans notre étude et une extension de leur méthode pourrait inclure l'analyse de la sémantique latente. Les deux types de régions par frontière et par points remarquables sont complémentaires et leur étude combinée devrait fournir de très bons résultats.

Cela nous ramène à la tâche de la fusion dont les perspectives sont traitées dans la prochaine section.

6.2.2 Fusion

Comme nous avons pu le voir la fusion est une tâche difficile et importante pour construire des systèmes de classification efficaces. Nous n'avons traité la fusion sur un petit nombre de caractéristiques mais les futurs systèmes devront en fusionner une grande quantité. La couleur et la texture sont des caractéristiques visuelles qui peuvent être décrites de nombreuses façons comme nous avons pu le voir dans la section 1.4 à la page 13. Toutes ces caractéristiques ont leurs avantages et leurs inconvénients, et il est sage de laisser le système de fusion utiliser correctement les bons descripteurs. De même uniquement le texte a été utilisé, c'est à dire la parole du flux audio. Pourtant la musique, les cris, les pleurs, ... jouent également un rôle important dans la caractérisation

du contenu. Le nombre de caractéristiques décrivant un plan ou une scène peut donc croître très facilement.

Nous avons vu que la fusion par des opérateurs simples est efficace. Dans le travail présenté, seule la meilleure fonction de fusion est conservée. Ce choix n'est peut-être pas le meilleur et une combinaison des différentes solutions pourrait améliorer la généralisation de l'algorithme. Nous avons également mis en valeur le problème inhérent à la désynchronisation du contenu visuel et du contenu textuel. La fusion pourrait très bien réduire ce problème en modélisant le décalage entre les différentes modalités.

Nous n'avons pas réellement abordé le thème important des ontologies. C'est à dire l'étude des relations entre les classes. En effet les concepts ne sont pas exprimés isolément et une forte corrélation existe entre certaines classes. Une première difficulté réside dans l'élaboration d'une ontologie (Noy and McGuinness [6]) qui décrit les relations existantes entre les concepts. La deuxième difficulté qui nous intéresse plus, réside dans l'exploitation de cette information sémantique et a priori par les systèmes de classification ou de fusion. La recherche est actuellement très active dans ce domaine, que ce soit pour l'analyse des documents distribués sur Internet (Alani et al. [1]) ou de vidéos (Wu et al. [10], Hakeem and Shah [3]). La fonction de fusion présentée dans la section 4.1.3 à la page 75 est adaptable à l'analyse conjointe de plusieurs classes. Cette adaptation fera l'objet de prochains travaux pour ensuite intégrer les ontologies à la fusion.

Bibliographie

[1] H. Alani, Kim Sanghee, D.E. Millard, M.J. Weal, W. Hall, P.H. Lewis, and N.R. Shadbolt. Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*, 18(1) :14–21, 2003.

[2] V. Gouet and N. Boujemaa. On the robustness of color points of interest for image retrieval. In *Proceedings of the IEEE International Conference on Image Processing*, volume 2, pages 377–380, 2002.

[3] A. Hakeem and M. Shah. Ontology and taxonomy collaborated framework for meeting classification. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 4, pages 219–222, 2004.

[4] B. Huet and E.R. Hancock. Line pattern retrieval using relational histograms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12) :1363–1370, December 1999.

[5] B.T. Messmer and H. Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[6] Natalya Fridman Noy and Deborah L. McGuinness. Ontology development 101 : A guide to creating your first ontology. Technical report, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, 2001.

[7] K. Sengupta and K.L. Boyer. Organizing large structural modelbases. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995.

[8] K. Shearer, S. Venkatesh, and H. Bunke. An efficient least common subgraph algorithm for video indexing. *International Conference on Pattern Recognition*, 2 :1241–1243, 1998.

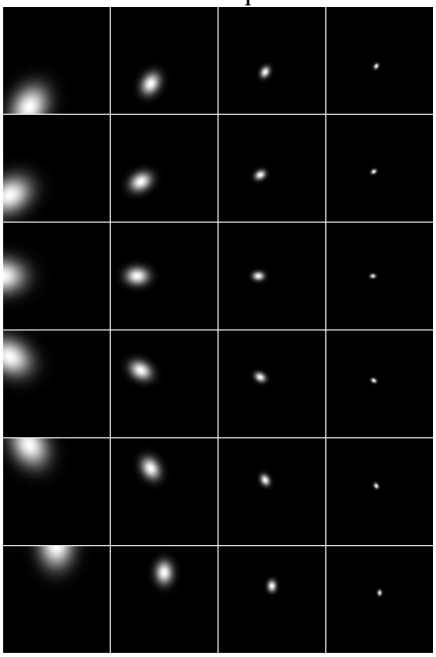
[9] Josef Sivic and Andrew Zisserman. Video google : A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2003.

[10] Y. Wu, B.L. Tseng, and J.R. Smith. Ontology-based multi-classification learning for video concept detection. In *Proceedings of the International Conference on Multimedia and Expo*, volume 2, pages 1003–1006, 2004.

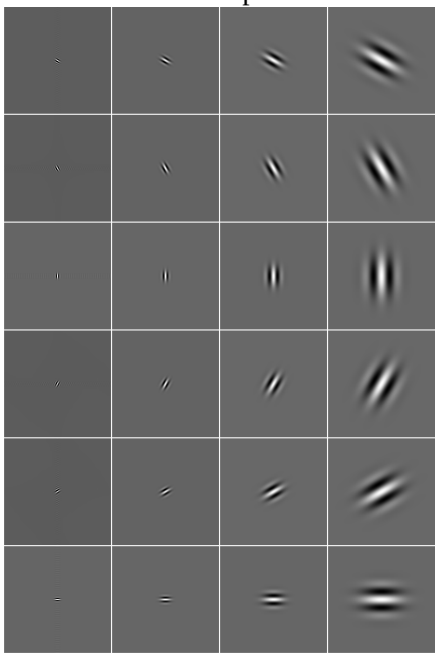
Annexe A

Filtres de Gabor

Réponse des filtres de Gabor dans le domaine fréquentiel.



Réponse des filtres de Gabor dans le domaine spatial.



Annexe B

Extraits des nouvelles télévisées

Plans en relation avec les classes *studio* et/ou au *présentateur du journal* ($id \in \{1, 2, 3, 4\}$)



Plans en relation avec la classe *Bill Clinton* ($id = 5$)



Plans en relation avec la classe *basket-ball* ($id = 6$)



Plans en relation avec la classe *hockey sur glace* ($id = 7$)



Plans en relation avec la classe *nature et végétation* ($id \in \{8, 9, 10, 11, 12\}$)



Plans en relation avec la classe *désert* (*id* = 13)



Plans en relation avec la classe *montagne* (*id* = 14)



Plans en relation avec la classe *plage* (*id* = 15)



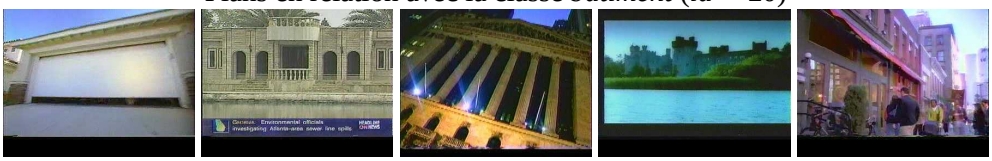
Plans en relation avec la classe *route* (*id* = 16)



Plans en relation avec la classe *avion* (*id* = 18)

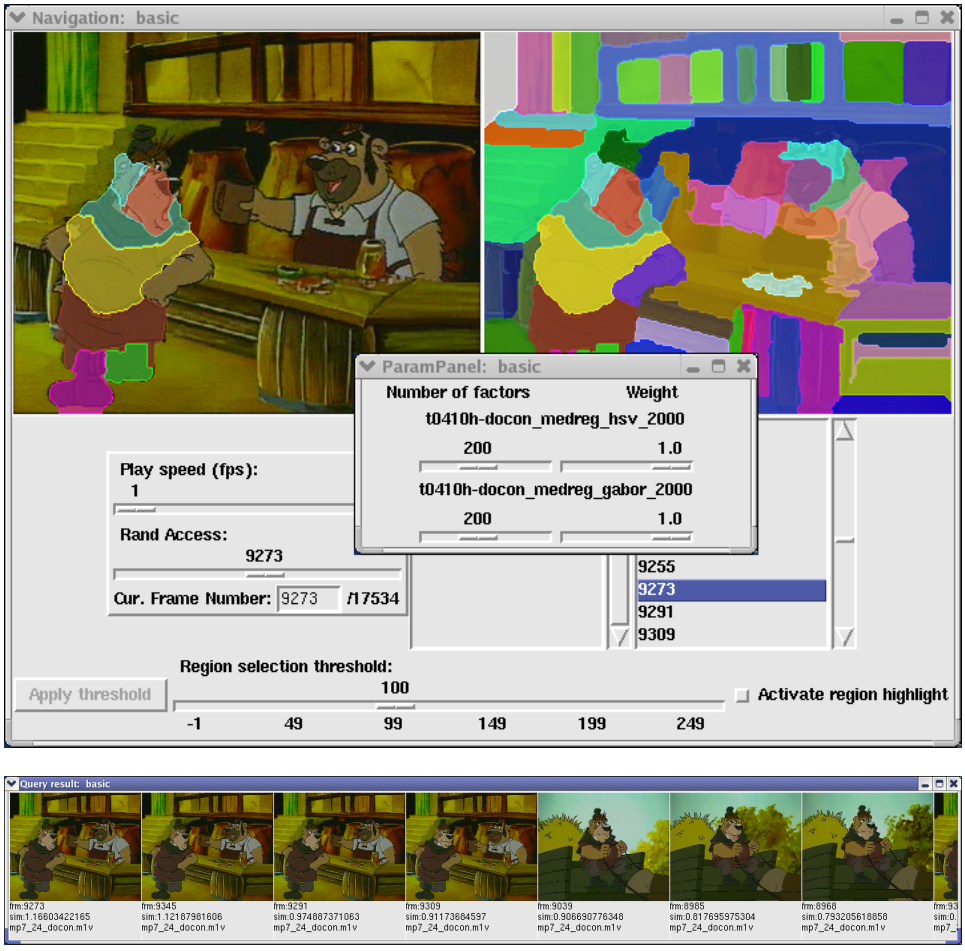


Plans en relation avec la classe *bâtiment* (*id* = 20)



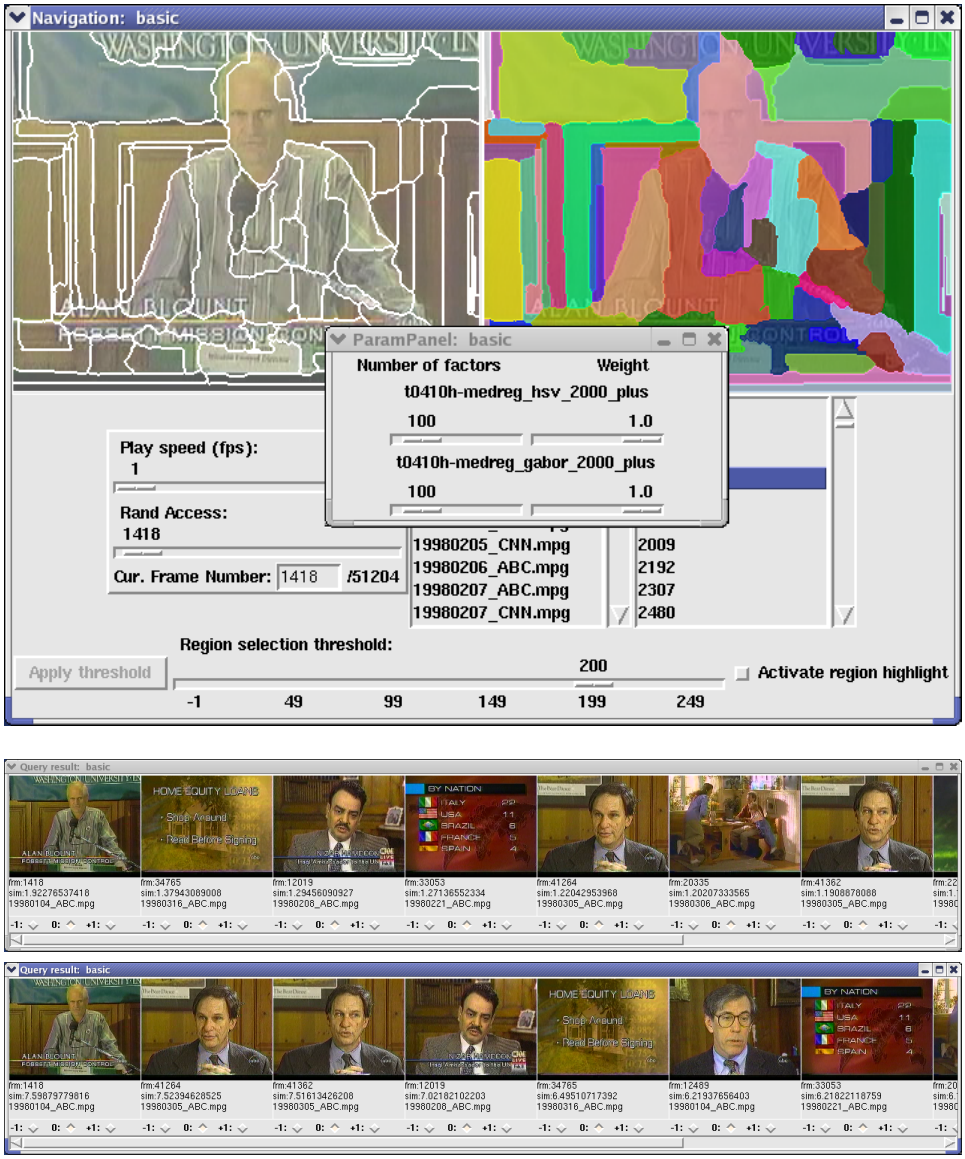
Annexe C

Logiciel de recherche de plans vidéo



Requête sur un objet dans la série de dessins animés. L'image de gauche est l'image d'origine où sont sélectionnées les régions pour faire la requête. L'image de droite est le résultat de la segmentation. La fenêtre centrale permet de fixer la dureté de la LSA et les poids des cartéristiques de couleur et de texture. La partie inférieure de la fenêtre principale permet de naviguer de vidéos

en vidéos et de plans en plans, et de jouer le flux vidéo courant. Finalement la dernière fenêtre montre le résultat de la recherche.



Requête sur l'ensemble de l'image dans la base de données des nouvelles télévisées. Réultats obtenus sans puis avec une itération dans la boucle d'asservissement.



Requête sur le présentateur dans la base de données des nouvelles télévisées. Réultats obtenus sans et avec une itération dans la boucle d'asservissement, puis résultats obtenus lors d'une requête sur l'ensemble de l'image.