Toward automatic semantic multimedia structuring and retrieval

Janvier, Bruno

UNIVERSITÉ DE GENÈVE
Département d'informatique

FACULTÉ DES SCIENCES
Professeur T. Pun
Dr. S. Marchand-Maillet

Toward Automatic Semantic Multimedia Structuring and Retrieval

THÈSE

présentée à la Faculté des sciences de l'Université de Genève
pour obtenir le grade de Docteur ès sciences, mention informatique

par

Bruno JANVIER
de
Paris (France)

Thèse n° 3882

GENÈVE
Print&Media - Umeå Universitet
2006

UNIVERSITÉ DE GENÈVE
Département d'informatique

FACULTÉ DES SCIENCES
Professeur T. Pun
Dr. S. Marchand-Maillet

Toward Automatic Semantic Multimedia Structuring and Retrieval

THÈSE

présentée à la Faculté des sciences de l'Université de Genève
pour obtenir le grade de Docteur ès sciences, mention informatique

par

Bruno JANVIER
de
Paris (France)

Thèse n° 3882

GENÈVE
Print&Media - Umeå Universitet
2006

Contents

List of Figures

List of Tables

Acronyms

CBVR Content Based Video Retrieval

CBIR Content Based Image Retrieval

DVR Digital Video Recording

OCR Optical Character Recognition

TRECVid TREC Video Retrieval Evaluation

MPEG Moving Picture Expert Group

DCT Discrete Cosine Transform

RGB Red, Green, Blue

HSV Hue,Saturation, Value

YUV Luminance, Bandwidth, Chrominance

GLCM Gray Level Co-occurrence Matrix

MSE Mean square error

MAD Mean absolute difference

SC Spectrum Centroid

BW Bandwidth

ZCR Zero Crossing Rate

HZCRR High Zero Crossing Rate Ratio

NSR Non-silence Ratio

FCVC4 Frequency Component of the Volume Contour around 4Hz

RMS Root Mean Square

LSTER Low Short-Time Energy Ratio

LEF Low Energy Fraction

SR Spectrum Rolloff point

SF Spectral flux

CF Cepstrum flux

CRRM Cepstrum Re-synthesis Residual Magnitude

MFCC Mel-Frequency Cepstral Coefficients

PSTD Pitch standard deviation

MFCC Mel-Frequency Cepstral Coefficients

STFT Short-Term Fourier Transform

ERSB Energy Ratio of a particular sub-band

VSTD Volume Standard Deviation

PSTD Pitch standard deviation

VMR Voice-or-Music Ratio

NUR Noise-or-Unvoice Ratio

KL Kullback Leiber

MDL Minimum Description Length

MML Minimum Message Length

BIC Bayesian Information Criteria

AIC Akaike Information Criteria

DPA Dynamic Programming Algorithm

SVM Support Vector Machine

HMM Hidden Markov Model

HHMM Hierarchical Hidden Markov Model

MCMC Markov Chain Monte Carlo

STG Scene Transition Graph

ML Maximum Likelihood

MAP Maximum a posteriori

EM Expectation-Maximization

PAC Probably Approximately Correct

Résumé en français

Introduction

De nos jours, les données multimédias digitales s'amassent en quantités énormes dans les bases de données autant personnelles que professionnelles et cela à une vitesse toujours croissante. Le problème lié à la surcharge d'information ne cesse de croître alors que de la même façon les technologies qui nous permettent d'acquérir, stocker et transférer ces données ne cessent de s'améliorer. Les documentalistes sont submergés par la difficulté à retrouver les informations. Il existe une grande demande pour le développement de nouveaux outils dédiés à l'indexation et à la recherche d'information multimédia.

L'architecture générale d'un système de recherche de vidéo par le contenu comporte, à l'entrée du système, un flux vidéo. Il s'agit habituellement d'un flux compressé (MPEG, MPEG2, MPEG4) contenant l'information audio et visuelle ainsi que des données de synchronisation. A la sortie du système, on trouve un utilisateur et/ou une application à qui il est offert diverses possibilités pour faire des requêtes dans le système afin de retrouver l'information vidéo recherchée. Le système en lui-même consiste en différents modules interconnectés comprenant en règle générale l'extraction de descripteurs, la structuration temporelle, la classification sémantique, l'indexation, la recherche par similarité, la visualisation et les outils d'explorations des collections et des résultats, etc. La mise au point d'un tel système est hautement pluridisciplinaire.

Cette thèse se focalise sur le problème de la structuration temporelle des vidéos. Plusieurs décompositions sont possibles à partir d'un même document:

- le niveau le plus fin correspond à la découpe de la vidéo en images (une vidéo contient fréquemment entre 24 et 30 images par seconde)
- une segmentation en segments homogènes groupe les images selon des critères de similarité
- une segmentation en plans qui tient en compte de l'édition qui a été préalablement opérée par le montage de la vidéo
- une segmentation sémantique qui groupe les plans par thématique ou sujet

Nous allons nous intéresser dans cette thèse à chacune de ces décompositions qui posent toutes des problèmes particuliers. Le but final étant d'obtenir des représentations de la vidéo selon diverses échelles et points de vues afin de faciliter leur exploration. De plus, ces décompositions sont utiles pour une analyse plus approfondie du contenu des vidéos avec des algorithmes de classification ou de détection sémantique.

L'évaluation des résultats est un point important à considérer. Souvent, les chercheurs se trouvent confrontés à la difficulté de comparer leurs résultats à cause de l'utilisation de corpora, d'annotations ou de métriques d'évaluations différents de leurs collègues. De plus, il est souvent coûteux d'obtenir un ensemble d'évaluation suffisamment large pour que l'apprentissage des algorithmes se déroule dans de bonnes conditions. Le forum TRECVID a permis de répondre à ces problèmes et les algorithmes que nous allons présenter dans ce travail ont tous été évalués grâce aux corpora, annotations et métriques provenant de ce forum. De plus, nos travaux sont comparés aux autres participants qui ont publié leurs résultats.

Extraction multimodale de descripteurs

La structuration temporelle de vidéos est fortement dépendante des descripteurs qui peuvent être extrait des documents. Il est nécessaire que les descripteurs soient d'une dimensionalité raisonnablement faible pour des raisons de stockage et de temps de calculs. Dans le même temps, il est aussi nécessaire que les descripteurs offrent des propriétés d'invariance par rapport aux changements qui ne sont pas intéressant pour notre problème. L'extraction de descripteurs cherche à résoudre ces deux problèmes

dans le même temps. C'est une tache difficile qui est loin d'être totalement résolue. Les descripteurs les plus fréquemment utilisés par la communauté des chercheurs sont encore extrêmement simples, et les descripteurs plus complexes qui ont été développés posent de sérieux problèmes dans leurs utilisations pratiques pour des raisons de temps de calcul ou de stabilité. Lorsque que l'on s'intéresse à la classification automatique, le choix des descripteurs est fondamental. Un classifieur simple peut réussir des miracles si on lui fournit des descripteurs qui sont discriminants. Dans le même temps, un classifieur complexe peut ne pas savoir résoudre le même problème si les descripteurs ne le sont pas.

On peut distinguer les descripteurs dit "bas-niveaux" qui sont souvent des statistiques extraites directement des données des descripteurs "haut-niveaux" qui se révèlent moins robuste car ils proviennent d'un processus de classification capable de commettre des erreurs. On distingue aussi les descripteurs dit "génériques" utilisables pour diverses tâches des descripteurs "ad hoc" qui sont générés pour résoudre un problème bien spécifié. Dans notre travail, nous allons principalement nous intéressé à des descripteurs "bas-niveaux" et "génériques" afin d'étudier jusqu'où ces descripteurs peuvent nous mener et maintenir notre système extensible à diverses applications.

L'information visuelle d'une séquence d'images peut être décrite à l'aide de différents descripteurs: histogrammes de couleur, flots optique, segmentation et suivi des objets. L'information audio peut être analysée dans le domaine temps-fréquence pour décrire la texture du son. La reconnaissance automatique de la parole permet d'extraire de l'information textuelle à partir de la modalité audio. L'information textuelle peut être inscrite dans les images et extraite à l'aide de technique de reconnaissance de texte. La description obtenue est dépendante de l'unité de temps choisie. Différents descripteurs peuvent viser à la description d'une image unique, d'un plan ou d'une scène.

Segmentation basée sur l'information

La segmentation temporelle des documents vidéos au niveau édition est le premier pas vers un système complet de recherche de vidéos par le contenu. Le but est de diviser le flot vidéo en éléments simples afin de les indexer. Dans la communauté

scientifique étudiant ce champ de recherche, l'élément simple le plus communément utilisé est le plan. Les plans peuvent être séparés par la détection des transitions du flux vidéo. Ces transitions peuvent être abruptes lorsqu'il y a coupure de la caméra ou graduelles lorsque les transitions sont ajoutées à l'aide de logiciels d'édition vidéo comme des effets de "dissolve", "fade-in" ou "fade-out".

Dans ce travail de thèse, nous proposons de structurer la vidéo à niveau de détail plus fin que le plan. Un document vidéo peut contenir typiquement plusieurs types d'actions dans un même plan. La caméra peut filmer un objet à un certain instant, puis filmer quelque chose d'autre plus tard sans qu'il y ait de coupure définissant un plan. Il est donc intéressant de grouper la vidéo en segments plus fins selon un critère d'homogénéité. Ce type d'algorithme a largement été ignoré par la communauté pour des raisons de temps de calcul. Nous allons montrer qu'il est possible d'utiliser ce type d'approche grâce à quelques heuristiques simples et qu'elles sont compétitives en terme de temps de calcul avec d'autres approches.

Une segmentation basée sur l'information contenue dans les vidéos prend en compte toute l'information disponible sans se focaliser sur les transitions entre plans, mais cela pose de nouveaux problèmes. Quelle est le nombre optimal de segments permettant une description complète de la vidéo ? Il faut distinguer deux aspects différents: un aspect lié au données qui prend en compte la complexité du contenu à représenter et un aspect lié à l'utilisateur qui a des besoins différents à des moments différents et qui sera intéressé par des représentations plus ou moins grossières.

Pour notre algorithme, le contenu vidéo est représenté par un profil de dissimilarité qui est une série temporelle représentant la distance entre chaque image. Nous avons focalisé nos efforts sur la modalité visuelle pour plusieurs raisons: la modalité visuelle est capable de capturer les changements qui ont un sens au niveau de la scène et des descripteurs simples et rapides à calculer sont disponibles pour cette modalité. Nous proposons un modèle linéaire basé sur la somme cumulative du profil de dissimilarité afin de capturer l'homogénéité à la fois dans le contenu, mais aussi dans la dynamique. Ensuite, il s'agit d'ajuster le profil de dissimilarité à notre modèle et d'introduire un critère de régularisation qui provient de la théorie de l'information. Le terme de régularisation rend possible d'obtenir une segmentation vérifiant l'hypothèse d'"Occam's razor": Entre tous les modèles, le modèle le plus simple qui s'ajuste aux données est celui qu'il faut choisir. Cette segmenta-

tion est régularisée par le critère MML. Les segments sont inférés afin de maximiser localement l'homogénéité de l'évolution, mais aussi essaye de minimiser la complexité de la segmentation résultante par un algorithme de Dynamic Programming. L'optimisation est globale et donc plus satisfaisante qu'une stratégie "greedy" ou "agglomérative". De plus, aucun seuil n'est requis pour cette approche. Le nombre de segments ainsi que leur localisation sont inférés automatiquement en ne prenant en compte que l'information provenant de la complexité des données.

Résumé multi-échelle de vidéos par sélection d'images clés

Lors de l'exploration de collections de vidéos, les utilisateurs sont souvent seulement intéressés par une vision grossière des documents. Les outils de navigation dans ces collections doivent permettre d'assister les utilisateurs à localiser rapidement les extraits recherchés en fournissant des résumés visuels des vidéos selon un processus interactif. Les images clés sont communément utilisées pour résumer un extrait et pour fournir un point d'accès dans le temps. En sélectionnant les bonnes images clés, le processus de recherche peut être accéléré.

La segmentation au niveau de l'information obtenue précédemment est importante pour parvenir à extraire un ensemble d'image clé qui représente au mieux le contenu, mais n'est pas suffisante pour prendre en compte les besoins qui peuvent être divers de notre utilisateur. Pour prendre cela en compte, nous allons laisser à l'utilisateur choisir un niveau de détail pour la représentation en image clé. Nous proposons d'étendre notre précédent algorithme au multi-échelle. Pour cela, un paramètre supplémentaire est introduit dans la formulation du MML et un arbre représentant toutes les segmentations pour chaque échelle est construit de façon hiérarchique.

Au final, l'algorithme de sélection des images clés respecte notre souhait d'à la fois permettre une représentation qui s'adapte aux données et à l'utilisateur.

Segmentation au niveau de l'édition

Afin de détecter les transitions entre plans, la méthode la plus simple consiste à appliquer un seuil sur un profil de dissimilarité. Le problème étant que de nombreux phénomènes vont faire faillir cette stratégie simpliste comme les objets en mouvements rapides, le mouvement violent de caméra, les changements d'illumination rapides et le bruit dû aux erreurs du flux MPEG ou de la caméra.

Il existe différents types de transition entre plans. Les effets les plus courants sont la coupe ("cut") ou les fondu-enchainés, mais bien d'autres effets sont possibles et fréquemment utilisé en télévision comme au cinéma. Nous avons choisi une approche générique basée sur la segmentation obtenue précédemment pour résoudre le problème pour tous types de transition en opposition avec les méthodes ad hoc qui utilise un type de détecteur par type de transition.

L'algorithme se base sur la segmentation basée sur l'information. Chaque point séparant nos segments homogènes est considéré comme candidat pour être une transition entre plans ce qui réduit énormément la complexité du problème. Ensuite, une comparaison fine des images avant et après le point candidat est opérée à l'aide d'un test d'hypothèse qui minimise le risque d'erreur. Nous avons procédé à une évaluation grande échelle à l'aide des données provenant de TRECVID et comparé nos performances et le coût en temps de calcul par rapport aux autres participants.

Les performances sont bonnes parce que l'algorithme de détection de transition opère sur un espace de recherche réduit grâce au groupement en segments homogènes. De plus, le test d'hypothèse vérifie efficacement nos critères de sélection. L'évaluation est favorable particulièrement pour la détection des transitions graduelles avec une complexité en temps de calcul qui reste raisonnable.

Segmentation au niveau sémantique

Une façon intéressante de proposer une représentation parlante et compacte d'une vidéo est d'opérer au niveau sémantique. Le niveau sémantique est construit par groupement des plans qui partagent des propriétés similaires au niveau du sujet, de la thématique ou de façon générale de la sémantique. Le but global est d'organiser

une collection de vidéo avec un minimum de partition tout en préservant toute l'information sémantique nécessaire à la compréhension du contenu.

Différents types de collection de vidéos impliquent une notion différente pour le critère de groupement des plans. Par exemple, des vidéos contenant des matchs de sports tels que le football ou le golf peuvent être efficacement résumés en détectant les moments forts qui peuvent être extraits par l'analyse du son provenant du public. D'autres sports tels que le tennis se prête davantage à une structuration selon les règles de ce sport: par exemple en jeu, set et match. Pour des collections de films dramatiques, la notion de structure sémantique est bien plus ambiguë et difficile à définir puisqu'elle change souvent d'un film à l'autre. Les journaux télévisés sont naturellement structurés de façon bien plus claire. Les nouvelles parlent de sujets différents qui se suivent séquentiellement avec une découpe relativement simple.

Nous avons développé un algorithme de segmentation en "histoires" adapté aux collections de journaux télévisés provenant de ABC et CNN en utilisant l'information provenant d'un ensemble de descripteurs bas-niveaux. Le problème est traité comme un problème de classification. La généricité de nos descripteurs permet de penser que la même méthodologie peut se généraliser à la structuration de collection bien différente. Nous avons présenté une approche utilisable dans des scénarios réalistes en évitant toutes procédures ad hoc.

Pourquoi est-ce que les humains sont si supérieurs aux ordinateurs lorsqu'il s'agit d'interpréter la sémantique ? Une raison est attribuée à la capacité qu'ont les êtres humains à combiner la reconnaissance des objets avec la connaissance sur les associations possibles entre concepts et observations. La connaissance du contexte aide les humains à résoudre les ambiguïtés même lorsque les observations sont ambiguës. Nous avons proposé un modèle contextuel pour la segmentation des vidéos en nouvelles. L'apport majeur est la prise en compte des interactions entre diverses modalités lors de l'apprentissage. En terme de performances, il y a deux raisons principales à l'utilisation des relations contextuelles. Premièrement, le rappel d'un classifieur est amélioré car si les descripteurs sont ambigus, on peut espérer que le contexte permettra de lever l'incertitude. Deuxièmement, la précision d'un classifieur est améliorée dans le même temps en réduisant le nombre de labels possibles à choisir connaissant le contexte. La prise en compte des relations contextuelles permet de prendre en compte davantage d'information à priori et de lever l'incertitude

lorsque les descripteurs sont peu parlant.

Nous avons réalisé une évaluation sur un ensemble de données important et comparé nos résultats sur le corpus TRECVID. Les résultats valident cette approche par rapport à une approche non-contextuelle. Malgré tout, nous sommes forcés de constater que le niveau de performance de notre algorithme comme de ceux des meilleurs participants TRECVID est encore trop bas pour une utilisation autre qu'à des fins de recherche.

Conclusion

Dans cette thèse, nous avons étudié la structuration des documents vidéos selon les différents niveaux suivants: information, édition et sémantique. Différents critères sont utilisés pour le groupement des images des documents vidéos: homogénéité visuelle, detection des transitions entre plan et homogénéité thématique.

Le niveau de segmentation lié à l'information décompose la vidéo en segments homogènes visuellement. Pour démontrer l'utilité d'une telle décomposition, nous avons développé une application d'exploration qui est adaptative à la fois aux données et aux souhaits de l'utilisateur. Cette application offre des avantages significatifs sur les approches existantes pour faciliter une exploration qui part d'un grain grossier vers un grain plus fin tout en respectant la complexité des données.

Le niveau de segmentation lié à l'édition des documents est directement relié au problème de la détection des transitions entre plans et est reconnu comme étant un problème difficile lorsqu'il s'agit de transitions graduelles. Nous avons fondé notre approche sur la segmentation basée sur l'information pour obtenir une approche générique qui repose sur un ensemble d'hypothèses aussi simple que possible en lieu et place d'un détecteur ad hoc qui varierait selon le type de transitions rencontrées. Notre approche générique a été évaluée et comparée favorablement par rapport aux participants de TRECVID particulièrement pour la détection des transitions graduelles.

Le niveau de segmentation lié au sémantique requiert de spécialiser les données à un sous-groupe précis afin de pouvoir tirer parti d'information à priori. Nous avons choisi de nous focaliser sur la segmentation en nouvelles dans une collection

de vidéos contenant des journaux télévisés. Afin de capturer autant que possible d'information à priori, nous avons proposé un modèle contextuel pour prendre en compte les interactions entre les différents labels qui sont apposés aux données bas-niveaux à notre disposition. Les performances de l'algorithme ont été favorablement comparées à l'état de l'art par rapport aux même ensemble de données. Nous avons également démontré que l'inclusion du contexte apporte de l'information utile à la résolution de tels problèmes liés à l'inférence sémantique.

Nous avons validé nos approches sur ensemble massif de documents audiovisuels grâce aux données et aux annotations provenant de l'initiative TRECVID. Cela n'aurait pas été possible il y a quelques années et cela nous a donné la possibilité de comparer la performance de nos algorithmes avec d'autres groupes de recherche à travers le monde.

Chapter 1

Introduction

1.1 Preliminaries

Digital multimedia data is more and more common in personal or professional databases. Video is the most efficient way to capture audio and visual information of the world around us in real time. All the technologies needed to easily capture, store and transfer digitized video data are now mature and used both by professionals and hobbyists. The popularity of video has gained all aspects of society: institutional, corporate or private. It now plays an important role for the propagation of information: education, entertainment as well as political issues.

Digital video documentalists are overwhelmed by the amount of multimedia documents to handle. The video industry has developed detailed procedures to index, store, edit, retrieve and display video information. However, most of these techniques are not prepared to deal with the tremendous quantity of video documents available, because searching and annotating are still done using costly manual methods at many government or film studios facilities. The burden is not likely to ease if the technology to structure, organize and search for documents does not evolve as quickly as the technology to create new documents. Ten years ago in 1996, Aigrain *et al.* [1] pointed out that even if there is some hope that some of the needed annotations will be created during the production of video documents in the future, it will never be available for many existing documents and would maybe never become as widely spread as one could wish. Today in 2006, video collections are more efficiently encoded and transmitted than ever, but annotations are still lagging behind and not considered as a priority by audio-visual professionals. Thus

there is a need for the development of innovative and automatic tools to annotate large multimedia archives.

Before entering into details, two important definitions should be given. *Information retrieval* is the field of research concerned with the systematic manipulation of information in order to easily find it again (retrieve). *Content Based Video Retrieval (CBVR)* is concerned about ways to perform information retrieval by analyzing and indexing the content of video collections. In this introduction, we will first describe a general CBVR system and overview its different modules. Then, we will explicit the problematic of our work: the temporal structuring of video documents for CBVR applications. The fact that temporal structuring plays a central part for any CBVR systems will be emphasized. We will then discuss the difficulties related to the problem of evaluation of the results which we believe should be taken into account at the very beginning of our research. With respect to these evaluation issues, we will present the TREC Video Retrieval Evaluation (TRECVid) initiative. Examples of applications deriving from our research and useful for the real world will be described. Finally the outline of the thesis will be presented as well as our contributions.

1.2 A content-based video retrieval system

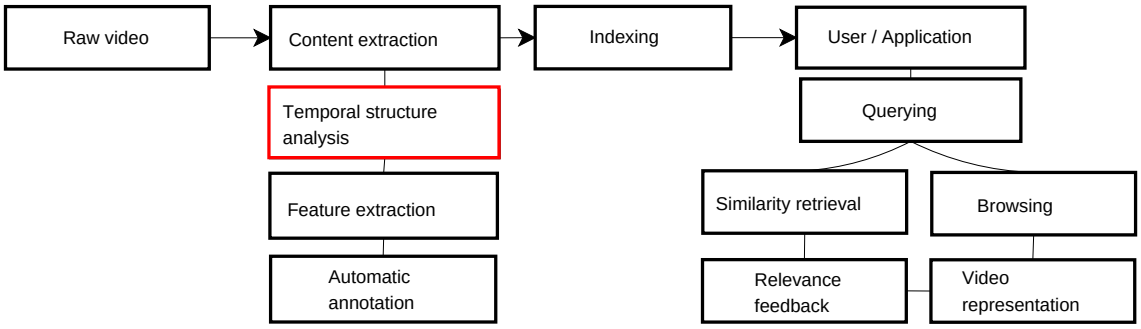


Figure 1.1: Architecture of a CBVR system.

The Figure 1.1 shows a general architecture of a CBVR system. This architecture can be described as follows. At the input of the system, we find the *raw video data*. It is usually a compressed (MPEG, MPEG2 or MPEG4) stream containing information about the audio and visual digital signals with synchronization information. At the output of the system, the *user/application* offers several possibilities to query the system to retrieve relevant video information. We will present now all the different components

of such a system by giving a general description of the content extraction, indexing and user/application modules. This will highlight the fact that the design of a CBVR system requires a combination of knowledge coming from many different research areas and that video structuring is an important building block on top of which a typical system is built.

1.2.1 Content extraction

The amount and quality of information extracted from the raw video data are determined by the *content extraction* process. Knowledge coming from computer vision, audio processing, natural language processing and machine learning is used to analyze the raw data and add features describing the content in the index. Three different processes can be distinguished: temporal structure analysis, feature extraction and automatic annotation.

Temporal structure analysis

The index of a CBVR system is usually based on temporal units: when querying the system with keywords or image-examples, the retrieval of relevant results operates on a given temporal unit. When indexing a video document, the first task is to perform a *temporal structure analysis* of the content. For a video collection of television broadcasts or movies, the most commonly used time unit is the shot, defined as an unbroken sequence of frames taken from one camera. The audio-visual industry (film-making or television) edits their data in order to communicate to their audience and uses conventional transitions between shots. From an indexing point of view, shots are an interesting time unit because:

- the conventions used for transitions between shots make their automatic detection feasible ;
- a shot represents a continuous action in time and space and therefore reduces the number of units to deal with for further analysis to a manageable size.

Automatic shot boundary detection techniques are classified in different categories: pixel based, histogram based and feature based [2]. It should be noted, though, that not all videos contain shots. In video surveillance or meeting recording applications, the video is captured continuously by the camera without editing. These video collection requires different methods for their temporal structuring. In general, different levels of segmentation are useful depending on the final application [3] to provide a coarse to fine time decomposition of video documents.

The holy grail of CBVR is to retrieve relevant documents at the semantic-level. Video structuring also plays an important role in this area. The historical approaches proposed in the literature aimed to construct a higher-level semantic segmentation by clustering visually similar shots [4]. Currently, efforts are spent on extracting the semantic video structure: scenes for movies [5] or home videos [6], significant events for sports video [7], news stories for news broadcast video [8]. Semantic video retrieval is improved instantly by incorporating such temporal structuring methods ; in [9] it was shown that the time needed to search into a video collection is dramatically reduced by using semantic video structuring methods thanks to the significant reduction of the search space and of the clear definition of the temporal support used for analysis.

In a general framework, the temporal units will define the time support on which retrieval or browsing is made and users should be allowed to select the time unit of interest accordingly for their queries from a single frame to a complex semantic unit grouping several shots. Temporal structure analysis will be our subject of interest in this thesis from the low-level to the semantic level.

Feature extraction

For each temporal unit of the video, *feature extraction* is performed to extract features describing the content in a compact and manageable way. Multimodality of video data [10] has to be considered: visual, audio and textual information are present and require specialized extraction procedures. Visual information is a sequence of image: color histograms, motion vectors, segmentation and tracking of objects are low-level features that describe visual aspects of the document. Audio information can be analyzed in the time-frequency domain to describe the texture of the sound [11]. Automatic speech recognition [12] algorithms also can be applied to extract textual information. Textual information also is found in the images of the video as captions and extracted via Optical Character Recognition (OCR) [13]. The description provided by the features is dependent on the chosen temporal unit. Different descriptors may aim at describing a single frame, a single shot or a complete semantic scene.

Automatic annotation

Automatic annotation means marking or tagging each temporal unit of the video to place it into a relevant class: the mapping from low-level features to high-level semantic concepts

is attempted here. This is a very challenging problem due to the well-known semantic gap problem: the lack of coincidence between the formative and cognitive information. It means that even if we have all the digital information needed for a human to understand the meaning of document, a computer algorithm is far from having the same cognitive capabilities to map the signal-level of the information to the meaning it does possess. Even so, automatic annotation is an important step to organize the database without using costly manual annotations ; the mapping from low-level features to high-level concepts is done using machine learning via classification/recognition algorithms [14].

The classifiers can be trained offline with training data or online by letting the users interactively indicate positive and negative examples. Classifiers are then used to recognize general semantic labels about the whole image (indoor/outdoor, sports/finance/cartoon, ...) [15–17] or to extract and recognize objects within the image (face, car, plane, ...) [18–20]. Unimodal methods focus on a single channel of the video content (for example: the visual information), whereas multimodal approaches fuse information coming from several channels (text, audio and visual information). The semantic and the meaning of the message conveyed by the video is often embedded in several modalities that are usually complementary. The possibility to use all channels altogether can improve dramatically the relevancy and the robustness of the detection/recognition of a particular event/setting in a multimedia document.

Here also, the temporal unit has to be defined for a given classification task. The classification needs a temporal support and often the temporal support has to be defined as well. As always segmentation and classification are interrelated problems. A good example of this interrelation is the application of hierarchical Hidden Markov Model (HMM) by Xie *et al.* [21] to infer a hierarchy of labels in an unsupervised manner to automatically structure and label soccer videos. This shows the importance of temporal structuring at different levels to classify video data at different time scales.

1.2.2 Indexing

The retrieval and browsing effectiveness will strongly depend on the quality of the *indexing*. By quality of an index, we mean the completeness and relevance of the information stored compared to the potential queries expected for the system. The indexing can be relative to the temporal segments: one entry per frame is too many, but one entry for the whole video is too coarse. That is why a reasonable solution is to store a feature

vector for each time unit which groups several frames with similar characteristics. Several levels of granularities should be supported by the index to support queries concerning fine or coarse temporal units. In the literature, object-based [22] and event-based [23] indexing structures are proposed in order to store the successive appearances of particular objects/events throughout the video sequence. We note that such indexing structures can be embedded into an indexing based on temporal segmentation by assigning labels to the segment marking the presence of a given object/event.

1.2.3 User/Application

Classically, there are two main directions offered to let a user interact with a CBVR system: *similarity retrieval* and *browsing* .

Similarity Retrieval

In *similarity retrieval*, we are searching for results that are supposed to be similar or relevant to the query: a sequence of keywords, a set of positive and negative audiovisual examples or the combination of the two. The research in Content Based Image Retrieval (CBIR) has proven to be very important to improve the quality of the results of a text-based retrieval system. If a keyword can refer to a concept, it still does not contain all the information included in an image or a sound. The similarities are classically defined on color, texture or shape low-level features. Different prototype systems (PictureSeek [24], GIFT [25], Blobworld [26]) as well as commercial systems (IBM's QBIC [27], Virage Image Engine [28], Virage Video Engine [29]) have been proposed in the last decade. Query by example approaches are not always suitable when the documents to retrieve appear significantly different of the example due to differences in image illuminations, scales, orientations or transformations of objects. More importantly, the naive user is more interested to search at the semantic-level than to specify features to describe its concepts.

As the user discovers the content of the corpus, it is important to let the user refines the query by a few loops of interaction: this is called *relevance feedback* [30], [31]. The feedback can be used to provide more training examples to an online classification algorithm or to redefine the parameters of the ranking function as examples. In [32], it was found that large multimodal feature spaces create too high dimensional spaces for a fast and interactive access such as relevance feedback. For the TRECVID news broadcast video collection, it was found in [33] that text search was helpful to identify interest-

ing candidate points and reduce significantly the search space so that high dimensional descriptors become usable for interactive retrieval.

Browsing

In *browsing*, we are more interested in presenting to the user a structured and summarized *representation of the video*. Also, browsing is totally complementary to searching, because there is a high probability that the content neighboring temporally the relevant item retrieved is also relevant. For example, a typical way to display the results of a query is to return a grid containing keyframes describing a set of contiguous temporal segments [34]. Browsing lets users selectively navigate in the video content via different methods.

Three different basic models have been proposed: time-line [35], hierarchical [36] or graph [37] browsers. These three approaches perform hierarchical clustering or graph construction based on visual similarities and near duplicates detection. Syntactic video structure such as dialogs or the successive appearances of a given shot are then captured. These methods reduce the number of information displayed to the user and helps improve retrieval speed. Fisheyes visualization optimizes the usage of the screen space by showing more details near the central relevant temporal segment [38]. Finally, storyboards have proven to be an efficient and effective technique to support visual browsing through temporally related shots. In [39], it was found that showing text in a storyboard gave more cues for browsing. Video summarization has a different purpose: it aims at presenting only the important content of a video document to a user. A measure of importance has then to be defined according to a criterion chosen depending on the specific video collection [40, 41].

For similarity retrieval and browsing, different time units should be supported to give to the user the ability to explore quickly (at a coarse level) and accurately (at the finest level of detail) a given video collection.

1.2.4 Summary

In this section, we have showed how the temporal structure analysis is included in a CBVR system. This is the first step toward content-based video indexing and automatic annotation.

Modern CBVR systems group all these methodologies together. For example, Columbia university in [42] combines text search, CBIR, story-based browsing, text search against

visual concepts (coming from automatic annotation) and near-duplicate detection in their system for news broadcast video retrieval. In [43], we proposed a database model to facilitate the development of content-based indexing algorithms and their benchmarking: the temporal structure plays again a central role as shown in the Figure 1.2. Descriptors and annotations are associated with the temporal segments. Therefore, the proper automatic extraction of meaningful temporal segments at various levels is a key issue for the design of CBVR systems as a whole.

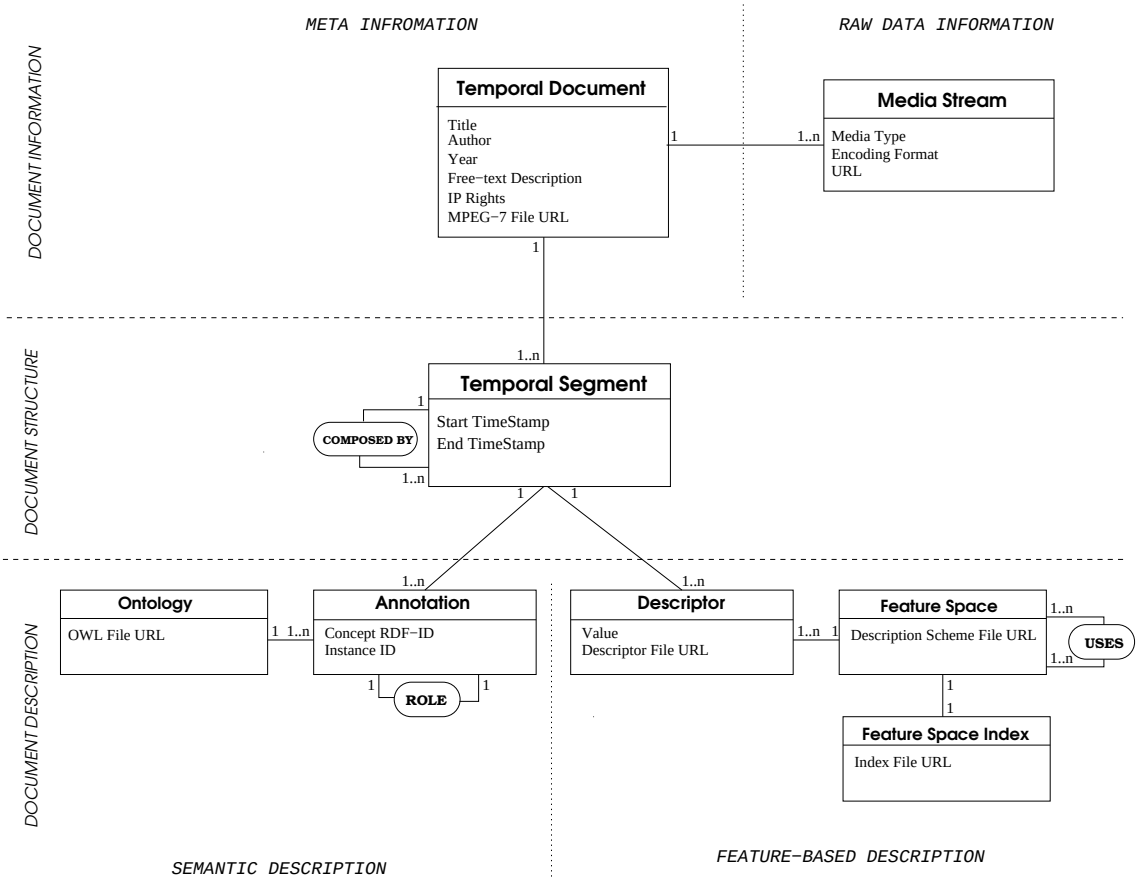


Figure 1.2: Conceptual model of a database representation of a CBVR system.

1.3 Aims and approach

The aim of our study is related to the general problem of content-based video management and is focused on algorithms dedicated to temporally structure video collections. Video structuring is obtained by temporally segmenting the document at different levels. A

temporal video segment can be defined as a group of frames which are contiguous and defined by a starting and an ending frame. The goal of our segmentation algorithms will be to define a set of temporal segments at different levels.

It is important to clearly define what will be the meaning of a given temporal segment, because this clearly depends on the application of interest. In order to illustrate this statement, we will here present two different examples. When considering the application of speech recognition, a segmentation is performed to separate different phonemes from the raw signal so that the recognition can be done phoneme per phoneme. For medical imaging applications, image segmentation can be performed to separate a given organ from the rest of a radiography image by using gray-level dissimilarities. In order to achieve these automatic segmentation tasks, some knowledge about what differentiates phonemes or body organs is obviously needed.

In our case, we are interested in applications to facilitate access to multimedia data. Therefore, it should be noted that concurrent levels of segmentation exist inside video data. In this thesis, we will focus our work on three different levels. The intermediate level is the **shot**: it is a basic unit which starts and stops with the recording of the camera. Movies and television broadcasts are usually created as a succession of different shots by the staff in charge of the video editing. 'Shot detection' is about automating the reverse engineering of the media capturing and editing processes and breaking videos into the atomic entities meant by the video producer. Transitions between shots can be abrupt or gradual (due to special effect transitions such as fade in/out, dissolve, etc). A large number of different methods exist to solve this problem. Our contribution is to define in this thesis a set of minimal rules to detect shot boundaries in a generic manner, i.e. without developing a specific detector for each different type of transitions.

Then we will be interested in a finer decomposition of the shot that we will call the **homogeneous segment**. A shot can have a quite long duration and show a succession of different contents. When trying to summarize a shot with a single keyframe, the risk of loss of information is important. Therefore, a homogeneous segment is defined as a set of contiguous frames which are similar from the visual point of view. In our case, we will focus on detecting homogeneous segment which possess a strong similarity in the color statistics. These homogeneous segments will be extracted by performing a complexity analysis of the shot to automatically determine the optimal number of segments so that valuable information is not lost. To achieve this task, we will use an important concept coming from information theory: the Kolmogorov complexity which quantifies the intrinsic complexity

of an object. In statistical inference, this is a fantastic tool to find the explanation which follows Occam's razor principle: the most simple model that explains data is the one to be preferred. These ideas can be formulated efficiently using the Minimum Message Length (MML). We propose to use the MML criterion to infer the segmentation which maximizes homogeneity and favors the solution that verifies Occam's razor hypothesis. For the videos which do not contain any shots (video surveillance, wearable camera, ...), methodologies based on visual homogeneity are the only possibility to obtain a temporal segmentation, because of the non-existence of any editing. For the videos which have been edited and contains different shots, this finer decomposition brings useful information about how to index and represent the content of each shot.

Finally, we will consider the semantic video structuring problem. Such video structures combine several shots to pursue the goal of adding a semantic understanding of the video, but the problem is difficult due to the semantic gap problem. It has to be noted that each video genre has its own syntax, semantic and rules. For the problem to become tractable, there is a trade-off to find between the needed knowledge about the context and the narrowness of the domain of application. For a collection containing drama movies, a higher-level of segmentation is a scene as defined in dramaturgy. When considering a collection of tennis videos, the game, set and match are interesting units for a semantic temporal segmentation. Also with a collection of news broadcasts, the semantic unit is the news story. In our work, we will focus on the extraction of **news stories** in broadcast news videos coming from the TRECVid community. We have at our disposal ground truth and annotations in order to evaluate the performance of our algorithms. This problem can only be solved by using a multi-disciplinary approach: computer vision, audio processing, natural language processing, machine learning and statistical sequence modeling will be used to reach our objective. We will see that this problem has been tackled by considering news story detection as a multimodal classification problem. We will be particularly interested in studying how contextual information surrounding a news story transition can be incorporated to support inference. By capturing more relationships between features/labels as well as labels/labels, we will see that the performance of multimodal classification can be improved.

1.4 The problem of evaluation

At the beginning of time for the field of CBVR, the results of different research groups have shown to be difficult to compare, as everyone used different corpora, the manual annotations as well as the metrics used to evaluate performances were not shared and varied in quality. A well known saying mentions that "repeatable experiments using published benchmarks are required for a research field to progress" and CBVR research has to go in that direction to develop.

Moreover, a large amount of training and test data are required to validate any algorithm. Such a large amount of data requires to be annotated manually so that results can be evaluated which requires a tremendous amount of time to do. Sharing data, annotation effort and metrics is a way to allocate our resource on the interesting work and is simply necessary.

This is precisely the intention of the TRECVid forum [33, 44], which facilitates research in information retrieval via open, metrics-based evaluation. It provides a large test collection and an uniform scoring procedure and is devoted to research in automatic segmentation, indexing and content-based retrieval of digital video. TRECVid is essential in enabling the evaluation and comparison the work of different research groups.

As training and testing data TRECVid chose to use a video collection containing CNN and ABC broadcast news programs. This corpus has been annotated according to a predefined set of labels called the ground truth. In this work, the dataset distributed in 2003 and in 2004 are used to benefit from the dataset, the annotation effort and the comparison possibilities of TRECVid.

ret455In this thesis, we did consider results coming from the campaigns of 2003 and 2004. Using more recent campaigns (2005 and 2006) was not feasible due to the absence of the story segmentation task.

1.5 Applications

As mentioned in [45], all the amazing applications that may come from the research in content-based multimedia technologies create a lot of excitment. However, not many convincing stories about successful applications are found on the market, the reason behind this fact is that this field of research is facing the same kind of barriers than the AI research

community. The difficult question is how to provide a link between the extracted low-level features and the high-level semantic description.

The answer will probably not come from one specific algorithm, but more likely through a rigorous analysis of user needs and thorough formulation and evaluation of CBVR systems. The potential users should be divided into two categories: the unexperienced non-technical users and the trained professional users. The applications have to constrain the complexity of their settings and of the amount of their possibilities according to the level of knowledge of the target users.

The following is a short and non-exhaustive list of the possible applications of such technologies: organizing information coming from news broadcasting, facilitating retrieval inside sport video, music video clips or home video collections, the possibility to dig into advertising collection so that advertisers can make sure that an idea is new.

At the current level of performance of CBVR systems, a constraint for their economical success is to focus on video collection which are so large and so cheap that it is not beneficial to employ humans to organize and annotate the data. For example, this excludes the movie industry who could have the financial power to create manually organized and annotated DVDs (or HD-DVDs) if the tools to search and browse was offered in consumer electronic devices.

Recently, the rapid and continuous decrease of the hard disk storage cost is enabling the interest of hobbyists for Digital Video Recording (DVR). It includes stand-alone set-top boxes for the living room as well as digital video cameras and software for personal computers which enables video capture and playback to and from disk which can today reach capacities in term of terabytes. Tools to organize and search inside this amount of data are bound to become successful one day.

1.6 Outline of the thesis

In this thesis we will present several algorithms that aim at extracting the inner structure of video documents so that the navigation and browsing of a video collection can successfully be improved. At the input of the system, we have only a raw audiovisual stream and at the output we wish to have a correctly partitioned representation of the video at different temporal scales. An important added-value to the video collection is then provided, because a potentially needed document is much easier to characterize and

find. To illustrate this statement, it is far easier to access and look for information in a collection of one month of broadcast news programs if one has access individually to each different story compared to the simple access to a large amount of frames. More intelligent navigation system can be devised on the basis of a properly extracted video structure.

The thesis is organized as follows. The second chapter of this thesis will show how content-based multimedia retrieval applications extract features from the raw audiovisual data. It contains information in many forms: visual, audio, but also textual which can be extracted using automatic speech recognition or OCR done on closed captions. We present how descriptors are extracted and how they are used.

The third chapter will present a contribution to the problem of temporal segmentation. From an algorithmic point of view, a video is a long sequence of frames without any structure whatsoever. We propose to partition shots into homogeneous segments. Inside a given shot, multiple actions may be taking place. A complexity analysis of the content of every shot should be made not to lose valuable information. For example, it is important to make this complexity analysis to infer the optimal number of keyframes needed to obtain the best possible summarization of its content.

The fourth chapter will propose two applications deriving from the segmentation in homogeneous segments. We will present a multiscale keyframe selection method which lets the user choose the coarseness of a video representation depending on his or her information needs. In addition, we will see that the partitioning of a shot in homogeneous segments helps to simplify the problem of shot detection even if very smooth gradual transitions occur: this is known to be a hard problem [2]. Another important advantage is that we will detect shot boundaries in a generic manner disregarding the kind of transition that occurred.

The fifth chapter will go further into the structuring of our raw data by grouping shots into stories ; this is the problem of semantic video structuring. We will see that the state of the art approaches are focused on considering news story segmentation as a multimodal classification problem. The novelty of our proposition will be in showing how inference can be supported by modeling a context around potential news story boundaries as well as how all these multimodal informations can be fused. A semantic concept such as a news story transition appears in a given context and we propose to use that knowledge to improve existing techniques.

1.6.1 Main contributions of this thesis

The main contributions of this thesis can be summarized as follows:

- Partition a video stream by an information-based segmentation to obtain a set of dynamically homogeneous segments with a global and parameter free optimization ;
- A multi-scale key frame selection method to summarize the video content adaptively at the data-level and at the user-level ;
- A generic algorithm for the detection of shot boundaries disregarding the type of special-effects involved. A quantitative and comparative evaluation is shown ;
- A contextual model to attempt the semantic segmentation of video into news stories by using a generic set of low-level features. A quantitative and comparative evaluation is shown.

Chapter 2

Multimodality of Video Data and Feature Extraction

Temporal structuring heavily relies on the characteristics extracted from the video documents for two different reasons. First, a limited dimensionality of the data is required for further analysis. Secondly, features should offer invariance properties according to non-relevant changes in the data and enable the detection/discrimination of relevant variations. Feature extraction aims at solving these two different problems by using knowledge coming from the study of the human perceptual/cognitive systems. From a general point of view, the ultimate goal of feature extraction is to understand and extract the inner patterns contained in the data in the same way as our senses and brain analyze them. This is not an easy task for several reasons. A large amount of information contained in the raw video data is not useful for characterization and hides what is important. Moreover the raw data captured by a camera or a microphone is the superposition of signals coming from different sources. There are many other problems. The state of computer science is still at its infancy stage when considering this ultimate goal of matching human cognitive abilities even if computer vision and audio analysis now have a history of several decades. We will show that the most commonly used features are actually quite simple and that even if more advanced sets of features have been proposed, they still pose several serious problems for practical use.

Nonetheless when dealing with classification algorithms (to reach the semantic level), the importance of the choice of features is fundamental. A simple classifier can achieve miracles with highly discriminative features whereas non-discriminative features will leave

highly complex classifiers helpless. Different categories of features can be distinguished. Low-level features (color histograms, texture response of gabor wavelets, optical flow, etc.) are extracted directly by estimation methods from the raw data whereas high-level features (face detection, speech recognition, etc.) are the results of a classification process based on low-level features and are less robust due to their higher complexity. This means that they are more easily prone to false or missed detections which makes them harder to rely on. Furthermore features can be categorized as generic when used to describe the video content for various analysis tasks or ad-hoc when designed specifically for one purpose (specific logo/jingle detection). In our work, we will prefer to consider generic features so that the developed systems stay also as generic as possible.

In this thesis, we use features that have been reported to work well for the segmentation/classification tasks we have to manage. It is important to invent new features just as it is important to consolidate the knowledge of reliable features under different frameworks of analysis. The focus of this chapter will be on the description of the computer vision, audio processing and text analysis aspects for extracting features out of the raw audiovisual data.

2.1 Visual information

2.1.1 Color

Color is clearly the most widely used low-level feature in the context of content-based indexing and retrieval of visual documents. Obtaining a description of the color content can be done by the selection of two sets of parameters defining the color model and the color statistics extracted.

Color models

The color of an image is represented through a color model which is specified in terms of a 3-D coordinate system. A wide choice of color models exists, we will only describe three of them: Red, Green, Blue (RGB), Hue,Saturation, Value (HSV) and Luminance, Bandwidth, Chrominance (YUV). The RGB color space specifies the chromaticities of the primary colors and uses the additive property of color on this three axis to generate all different colors. This is the color space used by computers to display an image on a

cathode ray tube (CRT) display. This color space is not perceptually uniform, meaning that a small perturbation to a component value is not approximately equally perceptible across the range of the value [46].

The HSV color space is approximately a perceptual uniform model. It is based on the artist concepts of Tint, Shade and Tone: it mimics how an artist mixes color on his or her palette. The color space is cylindrical where the value V is the axis of the cylinder, the saturation S is the distance perpendicular to the axis and the hue H is the angle around the axis. The three components decompose hue, saturation and value (corresponding to the brightness) which can be useful to develop illumination invariant similarity measure.

The YUV color space has been developed specifically by the video and television industry to achieve high quality compression through a perceptually uniform color model. This is the color space we find in the Moving Picture Expert Group (MPEG) video standard. This representation of color is normalized, it has the advantage of suppressing the intensity information and hence is invariant to changes in illumination intensity. The Y corresponds to the gamma-corrected brightness whereas the U and V are chromatic components. The YUV offers then the important advantage from the point of view of computational complexity to be directly encoded in the MPEG streams we are going to deal with. Moreover similarity metrics can be designed thanks to the perceptually uniformity of this color space.

Color statistics

To obtain a representation which is invariant to scale and orientation of the image, color histograms are commonly used. In the pioneering work of [47], color histograms are found to be suitable in image retrieval because "it is invariant to translation, rotation about an axis perpendicular to the image, change slowly with rotation about other axes, occlusion and change of distance". This statistical representation is not unique: two images representing a completely different content may possess the same color histogram. Spatial information about the pixels is totally lost. Still, CBIR research have empirically proven via benchmarking efforts that, in the context of query-by-example, the comparison of color histograms ranks usually higher very similar images in term of content [25].

To overcome the problem of the loss of spatial information, different representations have been proposed such as color-block histograms where the statistics are extracted in a decomposition in rectangular blocks of the images. The comparison between color-block

histograms then accounts for a rough spatial information. More advanced techniques to decompose the images spatially by using quadtree decomposition attempt to enforce the color homogeneity in each block which will vary in size and scale [48]. Color coherent vectors [49] have been proposed to also overcome the problem of loss of spatial information. Connected components analysis is performed to compute the degree of coherence of a particular color: the coherence is high for a compact block of pixels sharing the same color and low if the neighboring pixels possess different colors. The information about "coherence" is then incorporated by the similarity measures. The color similarities are simply ignored when the level of coherence is significantly different. The complexity of computing such a feature is much higher than a simple color histogram and has shown to be more sensitive to the overall image brightness which in turn is a problem for robustness. Color correlograms go even further in that direction by mixing color and texture description [50]. These captures local correlation between colors by building a table where the k-th entry for the pair (i,j) contains the probability to find a pixel of color i at a distance k of a pixel of color j. Only local probabilities are computed making the color correlogram fairly robust to global changes and efficient to compute via dynamic programming. However compared to the simple color histogram, the computational burden is much higher and the robustness is lower.

Another problem of color histogram is the high-dimensionality of the representation. Different methods have been proposed to attempt to reduce the dimensionality. A compact representation is obtained by estimating the color moments (mean, variance, third order moment). Even so a study [51] showed that it finally performs slightly worse than a high-dimensional color histogram in a content-based image retrieval application. Another proposal is the dominant colors [52] which is a compact descriptor based on the idea that a small number of color is usually sufficient to describe the color information of a given region. This hypothesis holds very well if one considers that a region is actually defined as a spatially contiguous area which is homogeneous in term of color by most image segmentation algorithm. The description of dominant color provides information about the distribution relative to a given region. The dominant color descriptor is very descriptive but requires a computationally expensive preliminary image segmentation into regions. Such a computational cost is currently acceptable when dealing with small image collections, but not for a large video collection such as the one we will use later in our experiments. In [53], color clustering is performed at several resolution levels from the general color histogram to the dominant colors. However the definition of dominant colors here does not take into account the spatial information about the pixels. Depending on

the application, this representation offers the possibility to use more or less information about color in a consistent manner.

We have seen here that various color descriptors have been proposed in the literature. All of them offer different advantages and drawbacks. The choice of the color feature depends mainly on the particular application. For analysis of large collection of video documents, we will retain color and color-block histograms in the YUV as a good trade-off between robustness, computational complexity and descriptive power.

2.1.2 Motion estimation

The automatic analysis of image sequences cumulates the difficulties related to static image analysis and the temporal dimensionality of the data. The temporal dimension provides a large new amount of information about motion, depth and objects of the scene. For video technology, motion estimation is useful for applications such as object tracking, global motion compensation, video coding and in addition with a descriptive purpose for indexing and retrieval applications.

In a typical scene, a motion estimation algorithm has only access to the optical flow: the 2-D projection in the image plane of a 3-D motion which is a combination of the motion of the camera and of the objects in the scene.

Many different profiles of 'motion' can be observed in a typical video scene: the global motion which is related to the motion of the camera, objects with smooth trajectories (ex: a car), objects with random trajectories (ex: soccer players, bushes), small or large objects, deformations and occlusions. Different families of approach can be distinguished in this field of research. Correlation-based techniques compare parts of the image at time t and $t+1$ using a similarity measure on the brightness patterns. Block-matching is the simplest method of all and is very popular for video encoding when speed is more important than the quality of the estimation. This class of algorithm assumes that:

- objects are rigid bodies - we neglect possible deformation of objects ;
- objects move in a translational movement ;
- illumination is spatially and temporally uniform ;
- occlusions of one object by another are neglected.

The block-matching strategy (see Figure 2.1) consists in partitioning the images into rectangular blocks. The distances used to compare two blocks can be the normalized cross-correlation function, Mean square error (MSE), Mean absolute difference (MAD) or the number of thresholded differences. The search of the displacement (r_x, r_y) is done on a range of discrete values. A single displacement vector is then estimated assuming than all the pixels in the block share the same displacement vector. This is untrue, but good enough for video coding if the prediction errors are coded. This strategy is inaccurate, but fast, robust and is able to measure displacements of a large magnitude. In addition it is commonly used in video compression standards such as MPEG 1 and MPEG 2.

Motion vector filtering is generally required to eliminate any outlier vectors which are generally unavoidable. Outliers occur as a consequence of the fact that matching blocks of pixels may fail for several reasons: flat untextured areas, too fast moving objects, abrupt appearance of objects. There are many different proposed approaches, but a typical pipeline for motion vectors filtering contains three different stages. The first stage is to filter motion vectors from areas with no texture or edges. Then a robust filter such as the median filter is used to remove outliers. Finally, an average smoothing can be performed to enforce regularization.

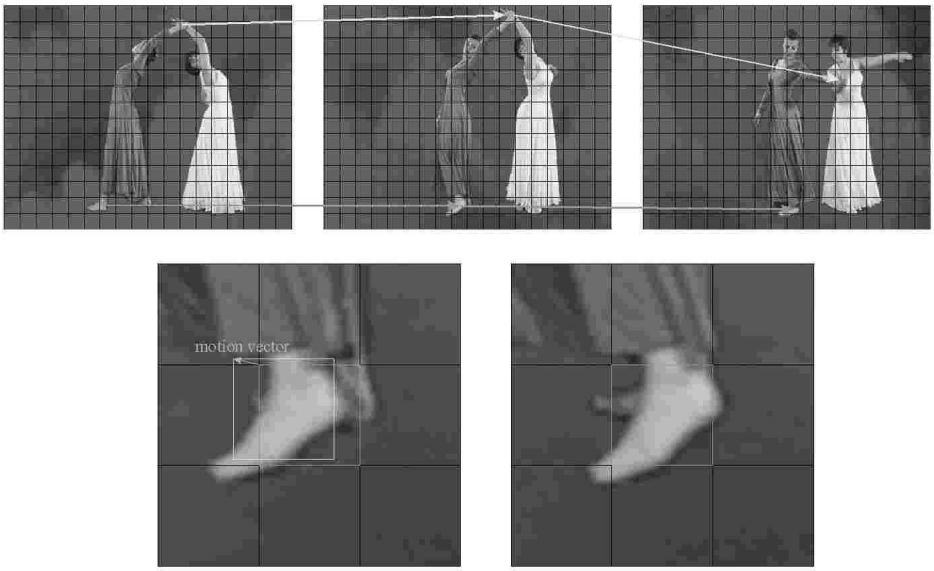


Figure 2.1: Illustration of motion estimation by block matching.

The feature-based methods analyze the optical flow at a small number of well-defined image features such as corners, edges or blobs and uses tracking algorithm such as

Kalman or particle filtering to estimate the trajectories. Corners are typically extracted from images thanks to the SUSAN [54] or Harris corner detector [55]. The problem is then to track features from one frame to the next. Ambiguous matching possibilities are the major problem to overcome. Parametric motion model can be used with constant velocity or constant acceleration. The Kalman filter [56] is a set of mathematical equations useful for tracking applications because it provides means to recursively estimate the state of a temporal process in a way that minimizes the mean squared error. It models the past behavior of a feature to predict the best possible match in the future. Such models are commonly used to disambiguate and track features from one frame to the next. By design, a limitation is the impossibility with such techniques to obtain a dense optical flow estimation.

Finally, there are gradient-based methods. Spatio-temporal partial derivatives are computed to estimate the motion at each point of the image by solving the optical flow equation which makes an hypothesis about the brightness constancy between successive frames. Multi-resolution strategies [57], [58], regularization [59], [60], [61] and robust estimation methods [62] are needed to improve the results. Currently, the high computational of such motion estimation technique is a problem when considering large video archives. Even with modern computers, one second of computation per frame is way too high when dealing with hundred of hours of video information.

In practice, the motion vectors directly extracted from the MPEG stream are preferred. The motion vectors we will use are then computed from a simple block-based estimation. Only the motion vectors for the P-macroblocks are considered, since I-blocks will have zero length motion vectors but this does not represent zero motion. For the video retrieval community, it is handy to use the motion vectors which have been pre-calculated and are directly encoded in the compressed video stream. Some filtering techniques are usually necessary to remove outliers and noise from this low-cost motion vectors. In the future, as the video compression community improves their algorithms for motion estimation, the indexing and retrieval community will benefit from it as well.

Once the motion is estimated, statistics are computed as for color information. There is a large number of possible representation for motion vectors ; we will mention here three different examples: motion histogram, dominant motion and motion wavelet coefficients. A histogram is constructed from the motion vectors where amplitudes and angles of vectors are stored in a given number of bins [63]. Dominant motion [64] also have been proposed in analogy with the dominant color feature. Smooth, multi-scale and compact representation

of the motion vectors in a basis such as the wavelet decomposition also have been proposed in [65]. The motion histogram is a rough, but stable and discriminative feature when motion intensity matters.

2.1.3 Texture

Texture is a characteristic of images which helps human to differentiate objects or parts of objects even when color or motion are not informative. It makes little sense to describe globally the image content by computing textural features. However textural feature can be used by an image segmentation algorithms together with different image features (edge, color and motion information) to discriminate between different regions [66]. Also when the region-map of the image is generated, textural features can be used to locally describe the different regions. When querying with an example of a particular texture, the images that should be retrieved contains the texture, but the spatial organization of these texture primitives is, in worst case, random. Therefore, most of the work on texture image retrieval is stochastic from nature.

We will mention here different techniques to extract textural features which are popular for their reliability. Texture is by definition a spatial property and is often captured by building a two dimensional matrix called the Gray Level Co-occurrence Matrix (GLCM) which measures the local dependencies between gray level values at various displacements and orientations. From the matrix, six statistical features (energy, contrast, entropy, homogeneity, correlation, inverse difference moment) [67] are computed to provide a descriptive and compact description.

A lot of research has been devoted to find a way to describe texture which matches human perception. It has been shown in [68] that the primary visual cortex can be modeled by Gabor functions tuned to detect different orientations and scales. This biological relevance has pushed the CBIR community to describe texture via Gabor filters to extract map of responses for various scales and orientations. Moments such as mean and standard deviations are then computed from the set of map of responses and stored in feature vectors [69]. More recently, wavelets have been used to analyze texture information into multiscale and oriented subbands via efficient transforms. The wavelet coefficients have been modeled using a Generalized Gaussian Density with success in [70] where very few model coefficients (18) are able to improve retrieval rates from 65% to 77% over traditional approaches in a CBIR experiment.

2.1.4 Shape

Shape provide a powerful clue to object identity and functionality, and can even be used for object recognition. Humans can recognize characteristic objects solely from their shapes: proof that shape often carries semantic information. This distinguishes shape from other elementary visual features, such as color, motion, or texture, which while equally important, usually do not reveal object identity. A large body of research has been devoted to shape-based recognition, retrieval, and indexing [71].

The goal is to transform a discrete shape as a robust affine and illumination invariant representation. From an image already transformed into black and white, one simple approach [72] is to extract dominant points by finding extrema of the curvature of the convex hull of the shape. The convex hull is the smallest convex refion encompassing the object. From the set of dominant points and the convex hull, it is then possible to extract affine invariant features such as the angle between pair of dominant points relative to the centroid of the convex hull. The conversion from color/grayscale to black and white generates noise which has often be the weak point of shape analysis for use in real situations. Of course many regularization schemes have been proposed such as deformable models, but are immediatly costly in term of computational complexity. Additionally, the work in this field often focuses on complete unoccluded images which are not at all guaranted to be found in real life videos.

2.1.5 Local descriptors

Points of interest: Points of interest are local descriptors characterizing only partially the image at corners or other strong geometric features of an image. They show robustness according to viewpoint changes and partial occultations of objects. There are several ways to extract points of interest usually based on first or second order derivatives of the image, but according to [73] the Harris color points of interests offer the best repeatability: that is the possibility to extract the points of interests corresponding to the same image geometric characteristics when viewpoint and illumination conditions are changing. The Harris color points of interests uses the RGB components of the image and only implies the first order derivatives of the image. They are defined by the positive local extrema of:

$$\mathrm{Det}(M) - k\mathrm{trace}^2(M) \tag{2.1}$$

where $k = 0.04$ and

$$M = G(\sigma) \otimes \begin{bmatrix} R_x^2 + G_x^2 + B_x^2 & R_x R_y + G_x G_y + B_x B_y \\ R_x R_y + G_x G_y + B_x B_y & R_x^2 + G_x^2 + B_x^2 \end{bmatrix} \tag{2.2}$$

.

$G(\sigma)$ is a gaussian smoothing term and the indices $(_{x,y})$ denotes the derivative operator according to a given spatial axis.

Once the point of interests are extracted, some invariants can be extracted from the neighborhood of each point to obtain a feature vector. In the litterature, the description often implies Hilbert's invariants [74]. Points of interests can be used for object tracking using Kalman filters or particle filtering. They can also be used in object matching in a query by example scenario. The set of points can be important and will by nature vary from one image to the next ; the matching algorithm proceeds generally as a voting procedure. For each points of the query, the closest points are found for each documents and a vote is computed for each image depending on the similarity of the closest points and their number. The query is therefore complex and is computationally inefficient without using multidimensional indexing structures.

SIFT: In the object recognition literature, the SIFT (Scale Invariant Feature Transform) [75] have shown to be a highly reliable local region descriptor in the comparison of Mikolajczyk [76]. The SIFT descriptors are a high dimensional representation of an image region that is invariant to translation, scale, rotation and robust to affine distortion, illumination variance and noise. A significant drawback of the SIFT features is the significant dimensionality of the representation and the computational cost involved. A typical image of 500x500 pixels will generate approximately 2000 features.

The SIFT descriptors are computed using four major steps. The first step is to find the scale-space extrema by searching for different scale the extrema of the Difference of Gaussian operator. Then measures of stability are used to locate each keypoints. An orientation is assigned to each keypoint based on local image gradients. Then descriptors are generated for each different region corresponding to the location, scale and orientation of the keypoints.

The matching algorithms are then quite similar than the one used for points of interest and have similar benefits and drawbacks.

2.1.6 Conclusion

In conclusion of this section, we did describe several ways to extract descriptors from videos, but all of these features will not be used in our work. There is always a trade-off to be found between the most complete description using as many features as possible and the curse of dimensionality and requirements in computational complexity that we need to keep under control in order to achieve our goal to analyze very large collection of video datas. Although these descriptors were studied in our work, some of them have not been used after analysis of this trade-off: that is that we found these descriptors to provide too little discriminative information relatively to the detection/classification tasks that we will treat in the next chapters compared to the significant increase in dimensionality and computational complexity that they imply. We therefore agreed to limit the number of different descriptors we used. We will describe accurately each used feature with more details later in the corresponding sections.

2.2 Audio information

Audio information is a mono-dimensional signal for which the automatic analysis has a long history. Audio signals are commonly visualized with the time-amplitude standard view where the time is on the X-axis and the amplitude of audio signals on the Y-axis. The amplitude is directly related to the physical position of the membrane of a microphone/speaker when it records or produces sounds. Fourier analysis plays an important role in audio processing to transform the data from the time domain to the frequency domain. The Fourier transform isolates the different frequency components of the signal and therefore facilitates further analysis.

Audio features are extracted at two different levels: the frame level and the clip level. The frame level groups samples by time windows of 20 ms within which we assume the audio signal to be stationary and where Fourier transform coefficients can be extracted. To obtain meaningful information, the clip level is a long term time frame corresponding to windows of 1 s. At the clip level, the features characterize how the frame-level features are changing during the clip.

We will present the feature extraction process by grouping features into three different groups. The first group of features are computed in the time-domain. The second group of features is computed in the frequency-domain. Finally the features of the third

group are based on the estimation of pitch.

2.2.1 Time-domain features

The Root Mean Square (RMS) volume is the most widely used and the easiest feature to compute. It measures the energy of the clip (volume, loudness). The square root in the formula makes it more perceptually accurate. For a given frame n, the RMS volume is:

$$RMS(n) = \sqrt{(\frac{1}{N}\sum_{i=1}^{N-1} s_n^2(i))} \tag{2.3}$$

where $s_n(i)$ is the i^{th} sample value and N is the number of samples in the frame n. The RMS volume is the average of the $RMS(n)$ over the all 1 second audio clip. It is particularly useful for silence detection.

The Volume Standard Deviation (VSTD) measures the repartition of the volume during the clip. It is simply done by the computation of the standard deviation of the RMS volume of each frames. This feature is useful to discriminate speech, music and noise, because a quick alternation of silences and loud frames will give a higher value for speech signals than for music where the volume is quite steady and slowly varying.

The Zero Crossing Rate (ZCR) counts the number of times the audio signal has crossed 0 over a frame. An interesting rule for silence detection is to combine a low RMS volume with a high ZCR. The use of ZCR avoids to classify the frames which are in reality low energy unvoiced speech frames as silent.

The High Zero Crossing Rate Ratio (HZCRR) is the ratio of the frames with a high zero crossing ($> 1.5 * \text{mean}$) compared with the other frames in the window. Introduced by [11], speech signals will have a higher HZCRR value than music.

$$HZCRR = \frac{1}{2N}\sum_{n=0}^{N-1}\left(\text{sgn}(ZCR(n) - 1.5\frac{1}{N}\sum_{n_2=0}^{N-1} ZCR(n)) + 1\right) \tag{2.4}$$

2.2.2 Frequency-domain features

Frequency-domain features offer the possibility to study the distribution of the frequencies during a clip, also to study the evolution over time of this distribution and finally to select particularly interesting sub-bands which can be designed to approximate human

perception of audio.

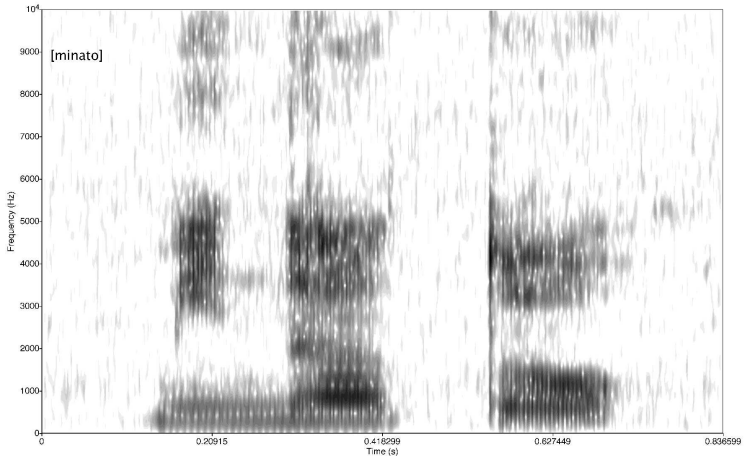


Figure 2.2: Spectrogram of a speech signal.

The spectrogram (see Figure 2.2) is a representation to visualize audio information in the frequency domain. It is a two-dimensional plot where time correspond to the X-axis and frequencies correspond to the Y-axis. The gray-value of each point correspond to the amplitude of a particular frequency at a particular time. A spectrogram is created using Short-Time Fourier transform on 20 ms frames over the whole clip. To avoid signal discontinuities due to the Gibbs phenomenon, the sound is first pre-processed over the window by multiplying the sound data by a Hamming function. And to avoid a loss of information due to the windowing step, overlapping frames are used.

The spectrogram contains all the information we might need, but its dimensionality is too high. The following features aims at providing a compact description of it.

The Spectrum Centroid (SC) and Bandwidth (BW) corresponds to the mean and variance of the spectrum of a given frame [77].

$$SC = \frac{\sum_{f=1}^{N}(f)^2|S(f)|^2}{\sum_{f=1}^{N}|S(f)|^2} \tag{2.5}$$

$$BW = \frac{\sum_{f=1}^{N}(f - SC)^2|S(f)|^2}{\sum_{f=1}^{N}|S(f)|^2} \tag{2.6}$$

where f, $S(f)$ are the frequency and the spectrum function.

The SC is related to the human sensation of audio brightness [78]. The BW shows how the frequencies of the audio signal are distributed around the spectrum centroid.

The Spectral flux (SF) measures the average variation frame per frames of the spectrum. It captures then how fast and how much the frequencies varies during the clip. It is a property about the dynamic of the sound. The equation is written in the following manner:

$$SF = \frac{1}{N}\sum_{t=1}^{N} \|S_t - S_{t-1}\| \tag{2.7}$$

where S_t is the Short-Term Fourier Transform (STFT) spectrum of the signal and t is the current frame. In general, the spectral flux is higher in speech than for music.

The cepstrum is a transformation of the spectrum to make it closer to human perception. The spectrum is then log-scaled in frequencies: $FT(log(FT(thesignal)) + j2\pi m)$. Indeed human ear is more sensitive to a variation between 300Hz and 350Hz than between 2000Hz and 2050Hz. The Cepstrum flux (CF) is a similar feature as the SF, but uses the cepstrum instead of the spectrum. It measures the average variation frame per frames of the cepstrum.

$$CF = \frac{1}{N}\sum_{t=1}^{N} \|C_t - C_{t-1}\| \tag{2.8}$$

where C_t is the STFT of the cepstrum of the signal and t is the current frame. In general, the cepstral flux is much higher in speech than for music and is therefore a discriminative feature.

The Energy Ratio of a particular sub-band (ERSB) [77] is a feature which uses a spectrum decomposition into 4 perceptual sub-bands from $0 - 630 - 1720 - 4400 - \infty$ Hz and calculates the energy ratios relative to all the different sub-bands with respect to the total energy of the frame. The four sub-bands are perceptual, because they contain the same number of cochlear filters according to the human auditory model [79]. It helps to distinguish between speech, music or noisy audio environment, because the energy distribution will vary in different sub-bands. For example, the energy of speech signals is concentrated in the low frequencies where noisy environment will have an uniform distribution of the energy in different sub-bands. The Frequency Component of the Volume Contour around 4Hz (FCVC4) focuses in a particular region of the spectrum to detect speech. The spectrum is multiplied by a triangular window around 4 Hz and the feature is

the energy of this product. The Spectrum Rolloff point (SR) was introduced by Scheirer *et al.* [80] is another measure of the repartition of the spectrum. It corresponds the point below which 85% of the magnitude lies. It approximates the measure of the skewness of the spectral shape. For the sake of completeness, we should cite the Mel-Frequency Cepstral Coefficients (MFCC) which are popular features in the speech recognition community. They use perceptual filters to distinguish between different phonems. However we will not be interested in our research by such a quality of detail in the audio analysis we will perform: discriminating speech, music and noise signals is our goal and all these different features cover different aspects of the spectrum to reach this goal.

2.2.3 Pitch features

The pitch is defined as the fundamental frequency of an audio signal. For a monophonic instrument, the pitch is nothing else that the frequency of the note actually played. But in polyphonic music, mixed sounds or noise, pitch is a more ambiguous notion. Pitch is the frequency that is the most present in the sound (i.e. the dominant note in harmony). Sometimes no frequency is more important than any other and in that particular case the pitch equals 0.

Methods to extract the pitch of a signal [81] may use an autocorrelation function, the cepstrum, an Average Magnitude Differential Function (AMDF), Comb transformations, or pattern-matching. For a given frame, the best algorithms will estimate a value which may vary significantly due to strong harmonics. The pitch is then estimated frame by frame and the value is smoothed at the clip level by using a robust estimator such as a median filter or M-estimator to remove outliers.

Pitch can be used in many applications in music retrieval, like score following and automatic music transcription for example. But some features based on the pitch and its variation can be useful in other domains, like segmentation or classification for example. The Pitch standard deviation (PSTD) is easily calculated using the standard deviation of the pitches of each frame in a clip. Liu *et al.* [77] have tested this feature but with poor results in classification, and high computational costs because of the pitch extraction step. We have seen that noise has an undetectable pitch, that is set to 0. This characteristic can be used to count the number of frames where pitch is present Voice-or-Music Ratio (VMR) or absent Noise-or-Unvoice Ratio (NUR) within the clip, and return a percentage of noise or music, that can be used to discriminate different kinds of sounds (speech from music

for example, where speech can be classified as noise during the unvoiced consonants). But here again, these features have a high computational cost because of the complexity of the pitch extraction algorithm used. This is why they are not often used.

The set of presented audio features is important and sometimes redundant, but the dimensionality of each feature is small: one real value per feature and per second for most of them. This perceptual features cover everything we need to discriminate silent, speech, music and noise audio environment.

2.3 Text information

2.3.1 Captions

Textual information is often embedded in the frames of video documents. Video OCR attempts to detect and recognize text inside video to take advantage of this important information for video retrieval. Text characters embedded in video are usually of low resolution, of various color with various complex backgrounds. In [13], video OCR is considered as a two stage process: first, text segmentation to extract textual information from the background is performed and then text recognition is applied. Text segmentation takes advantage of the specific texture of text: vertical and horizontal edges help to select candidates for potential lines of text. A classifier attempts then to provide the final decision between text or non-text. The recognition is done by using standard OCR techniques. The process is complex and the results are usually errorful. In the Informedia project [20], approximate matching techniques between strings has been proposed to deal with these recognition errors and add some robustness when trying to measure the similarity with a query and the text extracted from video.

2.3.2 Automated speech transcripts

Automatic speech recognition is a technology being developed so that the speech contained in the audio tracks can be recognized and transformed into a time-stamped transcript as accurately as possible. The process of speech recognition is highly complex and usually requires three successive stages: segmentation, feature extraction and decoding. The system needs robustness according to the words intelligibility, to the speaker characteristics (gender, voice, accent), to the vocabulary size, to noise in the audio data (crowd laughing,

thunder storm, ...) or channel conditions (bit-rate). In the TRECVid corpora, the LIMSI [12] provided the time-stamped automatic speech transcripts.

Advanced ASR systems (e.g. the cited LIMSI system) usually come with a speaker segmentation and categorization (male / female / telephone / music / noise ...). These information can then be used as available features for further analysis.

2.3.3 Textual descriptors

Once the textual data have been extracted by OCR or speech recognition algorithms, several analysis methodologies are commonplace to create useful descriptors. Many problems occur when attempting to analyse textual data. When dealing with automatic speech transcript, the problem of text segmentation arise, because no significant text symbol enables the identification of sentence boundaries. It is often a non-trivial task. Many words have more than one meaning; and the problem of word sense disambiguation is to select the meaning which makes the most sense in context. Syntactic ambiguity also is present, because the grammar for natural languages is often ambiguous, i.e. there are often multiple possible way to parse a given sentence. All these complexities are often taken into consideration by the research in natural language processing by focusing on statistical or distributional solutions. This is the reason why the most frequent way to describe text is the word vector representation. It consists in a vector where each component correspond to a given word of the vocabulary and the value of the component is the word frequency in the text under consideration. The frequency of word occurence is a useful indicator of word significance.

Several preprocessing methods are important to design textual descriptors. A problem that will arise is the fact that many words having the same meaning to a human will appear with several variations (might=may, solved=solving=solve, etc.). Lemmatization (or stemming) [82] is a preprocessing technique to reduce the vocabulary by stripping from english words common morphological and inflexional suffixes. As examples, lemmatization will change “studies”, “studied”, “study” into the lexeme “study”. Thus the vocabulary can be reduced in average by 30% in our experiments. Lemmatization is very useful, because the dimensionality of the vectors is then reduced, the word frequencies are better estimated and some invariance is introduced according to the common different suffixes of words. Another way to improve the word vector representation is to pre-process the text to remove “stop-words”. This is a list of words which are too common to be informative

such as “A”, “THE”, “IS”, etc. This reduces the dimensionality of the vocabulary and this words are so uniformly distributed in the english language that there is no loss of information. An important set of features are induced N-grams. The idea is to inductively determine the topic boundary markers common in news broadcast, such as “For XXX news, this is ...”. Around the boundary between topics in training data, all 1-, 2-grams found near these boundaries are listed as potential markers. For each, percentage of all occurrences which are near candidate boundaries is computed; then the list is sorted and the top ones chosen as boundary markers.

Named entity recognition [83] (also known as entity identification (EI) and entity extraction) has for aim to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, etc. The detection of such elements in textual data by simple string matching using regular expressions or other basic techniques. However when the textual data is noisy, some robust matching can be necessary. Named entity recognition provides additional useful features for semantic retrieval applications.

Document clustering or classification requires a similarity measure between set of words. Several measure has been proposed, but a natural one is the mutual information as defined in information theory. The Information Bottleneck algorithm [84, 85] uses the mutual information to cluster similar topics together. In fact, this unsupervised document clustering algorithm has shown to be as powerful as a supervised one to classify messages into the corresponding newsgroup categories. This is useful to determine from textual data a set of clusters relevant to topicality. The joint distribution $p(N, V)$ of documents N and vocabulary V has to be estimated. Then the information bottleneck algorithm is searching for a way to compress the set of documents N into a compact set of clusters T which preserves as much information as possible about the vocabulary V. The mutual information is used as an optimal metric from the information theory point of view. The goal is to compress the set of documents while preserving as much information as possible about the distribution of words. Hence the functional to minimize is given by the following trade-off:

$$F = I(T; V) - \frac{1}{\beta} I(T; N) \tag{2.9}$$

The trade-off β can be set to infinity if we are only interested in maximizing the relevant information V only. As β will get closer to 1, less balanced clustering solutions will be found where the small clusters T are found so that there is a minimal loss of

information about the documents N.

2.4 Video editing information

A large proportion of audiovisual databases are edited. This means that the shots are arranged next to each other in a particular manner. The way shots are edited convey information about the content of the video. We will describe here two simple features which will be useful for video structuring: shot duration and shot activity.

2.4.1 Shot duration

The average duration of the shots in a given time window gives a quantitative measure of the dynamic of the sequence. This simple feature is often used to discriminate slow versus rapid successions of shots ; the feature is useful to classify shots into semantical categories such as: music/cartoon which usually consist in short-lived shots (typically of the order of 1 second) and political/weather news for shots of longer duration in average as an example [8].

2.4.2 Shot activity

Another salient features to distinguish between semantic classes of interest for our work is the dynamic shot activity [86]. For example, anchor person shots in news broadcasts are filmed using a static camera and there is little object motion within the frame in these shots (i.e. low in visual activity) whereas sports or outdoor reports are filmed with a moving camera (i.e. high in visual activity). With this in mind, a measure of the amount of visual activity in a shot should prove useful for the discrimination of such semantic classes. An activity measure can be calculated as the ratio between the number of blocks with a small average magnitude in motion vectors to the total number of blocks in a frame, so a higher value indicates less visual activity. This activity measure is calculated frame by frame over a whole shot and is averaged.

2.5 Summary

In this chapter, we presented how visual, audio, textual and editing features are extracted from the video data. These features come from various signal processing, statistical or physiological inspirations. However, all of them share invariance properties according to what is not important in the data and discriminative properties according to what we want to detect or classify. Another constraint for us is that the features we can afford to use when dealing with large audiovisual databases also have to be computationally efficient. Our goal now is to propose different frameworks to make the best possible use of this classical and reliable pool of features in our problematic of video structuring.

Chapter 3

Segmentation at the Information-level

As we have seen in the introduction, the temporal segmentation of video documents is the first step toward a full-featured CBVR system. The goal is to divide the video stream into basic elements for indexing purposes. We have seen that the most commonly used basic element is the "shot" for the research community.

In this chapter, we propose to structure the video at a finer level of details than the "shot", because in a typical video document a transition in the action can be due to the simple and obvious fact that the camera often shows continuously one thing at a given time and a completely different thing at another time. Inside a given shot, the visual content may change due to several reasons such as a translation, rotation over an axis, a zoom of the camera or the appearance of large objects in the scene in a *continuous* manner. Especially for indexing purposes, it is important to detect these events as well, because we want to help the user to find anything in the video collection with a minimal loss of information. We propose then to structure the video into what we will call "homogeneous temporal segments".

This problematic has not been addressed in a formal way before by the CBVR community which usually choose the shot as the basic temporal unit by focusing on detecting the transition between shots. The originality of our work is to propose to segment the video according the information it contains and not according to only the detection of transitions. By analogy with the field of image segmentation: we will propose to use a method analog to region-based approaches whereas the community has focused its in-

terest to something similar to edge-based approaches methods. By using a region-based approach, we will focus on finding the best possible segmentation to get a relevant set of homogeneous segments. We will see that it requires to perform the segmentation with an information-based approach meaning that we will be doing a complexity analysis of the video content as a whole. Such approaches have been ignored by the community because of their high computational complexity. We will show that simple heuristics have a major impact to reduce the computational complexity and that an information-based approach can be competitive in term of computational time with other methods.

An informative segmentation aims at detecting all possible variations of the visual content. This leads to a different question: what is the optimal number of segments needed for the most complete partitioning of the content of the video ? The answer is obviously a variable number of segments per shot. How to determine this number ? Two different aspects have to be addressed for an attempt to solve this question:

- data aspect: the variability of the content within each shot - The amount of information displayed by a shot will vary depending on its visual complexity. The number of 'visual scenes' shown by a shot is not always equal to one. It is equal to the required number of key frames necessary for the best description. A complexity analysis of the visual data is required to find this optimal segmentation ;

- user aspect: the variability of information needs - it is characterized by two extremes where the user needs to see the video content at the highest degree of detail and at the other extreme needs only a very coarse representation of the document for a quick overview. The optimal number of segments is then chosen depending on what the user needs to learn about the document.

In this chapter, we will focus on the data aspect to segment the video data into atomic segments which will then be the basic blocks of further applications to address the user aspect of the problem (see section 4.1). We will present here an illustration to highlight the novelty of our proposal. With the approaches proposed in the literature dedicated to "shot detection" and with a typical video browsing interface, the data aspect is not really taken into account: only transitions between different shots are detected and anything can happen during a continuous shot and only one keyframe will be shown to the user. This represents a loss of information for the user. Our atomic temporal segments will let a typical user see a complete view about what happened during a given shot. We

note that this finer level of segmentation as well incorporates the information about "shot boundaries" and it will be useful to detect them afterwards.

In the first section, we will present the content-based video analysis aspect of the work. The video content is abstracted by a dissimilarity profile which is a time-series representing the frame-by-frame inter-distances. We will focus our efforts on the visual modality for several reasons : the visual modality is able to capture relevant changes in the content and is also interesting for browsing. A human user can process a lot of visual information in parallel and it is possible to represent the temporal variations of the visual content of the video by a set of well-chosen keyframes. We will propose a linear model based on the cumulative sum of the dissimilarity profile to enable the detection of changes in the content and in the dynamic of the video data. Then, in the second section we will fit the dissimilarity profile according to a model and introduce a regularization term coming from information theory. The regularization term makes it possible to obtain a segmentation verifying Occam's razor principle: the simplest model which fits the data is the one to be selected. Our information-based segmentation is based on the MML criterion proposed by [87] and will be applied to partition the video into segments where the evolution is homogeneous by taking into account all the available information. The segments are inferred in order to maximize locally the homogeneity of the evolution but also to minimize globally the complexity of the partitioning using a Dynamic Programming algorithm. The optimization process is global and therefore more satisfactory than greedy or agglomerative strategies. No threshold is required in such an approach. The number of segments as well as the location of the segments are inferred only from the complexity of the data.

3.1 Dissimilarity profile

3.1.1 Dissimilarity metrics

In order to perform a temporal segmentation of the visual information, we need a distance measure between two successive frames. The analysis will be done on the resulting temporal profile of the frame-by-frame distances. When the content is changing significantly, we expect the frame-by-frame distances to increase. At the very least, the similarity measure between two frames should satisfy the following properties:

- it should be stable with respect to changes that are common during a segment rep-

resenting a continuous action in time and space such as small affine transformations due to the camera motion, lighting changes, deformations, appearance of objects, etc.

- it should give an accurate quantitative information about the amount of change that has taken place.

The visual analysis is then reduced to the generation of a dissimilarity profile which the time-series of frame-by-frame dissimilarities. The literature is rich in ways to compute dissimilarities between frames. The approaches can be classified as: pair-wise pixel, histogram or feature-based dissimilarities.

Pair-wise pixel dissimilarity: The most simple way to define the dissimilarity of two successive frames is the image difference (also called the absolute sum of pixel differences):

$$D(i,j) = \frac{1}{N} \sum_{(x,y)\in I} |I_i(x,y) - I_j(x,y)| \tag{3.1}$$

where i and $i+1$ denote the indices of the considered frames I. $I_i(x,y)$ is the intensity value of the pixel at the coordinates (x,y) in frame i. For color images, the similarity is usually taken as the sum of the dissimilarities color channel per color channel.

However, this similarity measure will remain constant for the two following cases:

- if a large number of pixels have varied a little ;
- or if only a small amount of pixels have varied a lot.

As a consequence, this dissimilarity measure is sensitive to motion of the camera or of objects in the scene. Block-based approaches decompose each frame into N blocks and compare them to the corresponding blocks in the next frame. Blocks add rough spatial information to locate where the changes took place. More robustness can then be incorporated by considering in the similarity measure not only the amount of change pixel per pixel, but in addition the number of blocks concerned by significant changes.

Histogram dissimilarity: To go further in the direction of a robust measure of dissimilarity, we need to find a descriptor with good invariance properties. The color histogram of

an image is invariant to image rotation and slowly changing under the variation of viewing angle and scale [88] as we mentioned in the chapter 2. With unchanging background the motion of an object will impact very little an image histogram. The simplest approach proposed by Zhang *et al.* [89] is to compare globally the histograms of successive frames using the absolute sum of histogram difference (also called L_1 metric):

$$D(i,j) = \sum_{k=1}^{N} |H_i(k) - H_j(k)| \tag{3.2}$$

where $H_i(k)$ is the bin k of the histogram of the frame i.

In the CBIR research field, several similarity measures have been proposed considering histograms. The histogram intersection [25], the L_1, the L_2, the χ^2 test [90], the Jeffrey divergence and others. Experimentally, the Jeffrey divergence gives better quantitative results than the L_1, L_2 or chi-square metrics with a low computational cost. It shows the best results when dealing with busy scenes where pixel intensities change substantially from one frame to the next.

The Jeffrey divergence comes from the field of information theory [91]. It is related to a measure of the distance between two distributions (approximated here by an histogram) called relative entropy or Kullback Leiber (KL) distance.

$$D(i,j) = \sum_{k=1}^{N} H_i(k) \log\left(\frac{H_i(k)}{H_j(k)}\right) \tag{3.3}$$

One issue is that the KL distance is not symmetric. The Jeffrey divergence is nothing else than a symmetrized version of it. It measures how compactly one histogram can be coded using the other as a codebook. If H_i and H_j are two histograms containing N-bin, the Jeffrey divergence between H_i and H_j is defined by:

$$D(i,j) = \sum_{k=1}^{N} (H_i(k) \log\left(\frac{H_i(k)}{m(k)}\right) + H_j(k) \log\left(\frac{H_j(k)}{m(k)}\right)) \tag{3.4}$$

where $m(k) = \frac{H_i(k)+H_j(k)}{2}$.

Histogram-based dissimilarities take into account the proportions of each color values and are more robust to camera or object motion than pixel-based dissimilarities, but ignore the spatial information. Two images may have the same histogram and be significantly

different if one considers the spatial organization of the pixels. Therefore block-based histogram comparison are interesting to add the robustness of the statistical measure of histograms to the spatial analysis of block-based methods for more accuracy.

Feature-based dissimilarity: Different approaches focus their efforts on the computation of dissimilarities using features such as edges, motion or image region. In Zabih *et al.* [92], image edges are used, because during a cut or a dissolve transitional effect, the edges will disappear and reappear at a new location and usually far away from the previous edges. The detection of cuts, fades and dissolve is then done simply by counting the amount of exiting and entering edges. For these approaches, the compensation of motion is very important: otherwise many false detection will occur simply due to the motion of objects or of the camera. The edge detection methods will limit the effectiveness of the method, because it is not robust to rapid changes in the brightness and are hard to extract from light or dark frames. In [93], the L_1 norm of the first image derivative is compared to the norm of the second image derivative for detecting dissolves.

Another feature which is often used is motion. Bouthemy *et al.* [94] considers that motion information is more intrinsic of the video structure than pixel intensities. Motion vectors are used to compute dissimilarities between frames and detection techniques are applied to locate shot transitions. Even so, motion compensation is often applied to reduce the effect of motion of objects or of the camera when computing pixel-wise differences [93]. One major issue with using motion is that accurate motion vectors are extremely expensive in term of computational cost.

Going even further into the complexity of the video analysis, a measure of dissimilarity between frames based on an unsupervised image segmentation and object tracking has been proposed by Chen *et al.* [95]. This kind of approach is not economic when considering computational complexity, but the segmented images can be used for further video content analysis. The comparison is made on the image mask resulting of the top-down Maximum a posteriori (MAP) image segmentation.

In this section, we reviewed a large set of metrics which are attempting to measure frame-wise differences in a robust way. For our research, we are interested in obtaining the dissimilarity profile which offers the best trade-off between computational complexity, robustness and informativeness. Our choice is the color histogram which has proven to be a very stable representation in the content-based image retrieval research field. The distribution of color is invariant and stable for frames representing a similar content. We

will compute the histogram in the YUV color space, because it gives the best performance/speed ratio when dealing with MPEG video streams. We have chosen the Jeffrey divergence (3.4) as a dissimilarity measure, because gives better quantitative results in our experiments than the L_1, L_2 or chi-square metrics. By computing the Jeffrey divergence for the pairwise computation of histogram differences for the complete video stream, we obtain a frame-by-frame dissimilarity profile $Diss_{info}$ that will be used in the next sections:

$$Diss_{info}(i) = D_{col}(i, i+1) \tag{3.5}$$

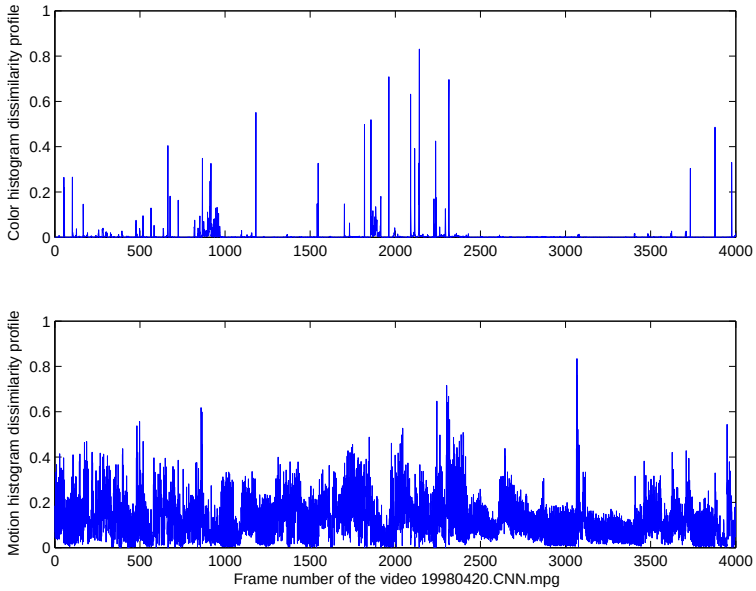


Figure 3.1: Color and Motion dissimilarity profile in function of the frame number for a typical video.

The Figure 3.1 shows a typical example of the kind of dissimilarity profile we obtain. In the color histogram dissimilarity profile, peaks correspond to abrupt transitions between shots where the image content changes a lot from one frame to the next. Note that, in our framework, video information is abstracted by its features. In this respect, it is possible to replace color information, for example by motion or sound, and get a partitioning that will hold a different interpretation than that obtained with the methods that will be presented next. However, our goal is to detect major changes in the video content and color statistics shows empirically to be the best alternative. We observe that a motion histogram dissimilarity profile is a lot more noisy and unstable relatively to what is considered changes that are common during a segment representing a continuous action in time and space.

Of course, it also depends on the level of sophistication used during the motion estimation process. In our example, the motion vectors have been estimated via the block-matching algorithm included in the MPEG compression standard. This is the motion estimation algorithm we would use in a real application because of its low computational cost.

3.1.2 Properties and modeling of the color dissimilarity profile

The partitioning of our video is done by considering that a video segment has been generated by a given model. The choice of the model is constrained by the following criteria:

- it should be able to fit the data during a homogeneous segment ;
- it should show an excellent detection performance in order to capture when dynamic of the video has significantly changed: the model should not fit discontinuities nor any major changes in the temporal evolution ;
- it should be generic enough so that it stays valid for any type of video document.

We will use the cumulative sum of the dissimilarity profile for the information-based segmentation because it measures the *trend* defined as the accumulated effect of the fluctuations of a time series. If the evolution of the colors is homogeneous and the frame-by-frame dissimilarity is roughly constant, the trend is expected to have a linear behavior.

The model that we will use for a segment is thus:

$$y_m^{\theta}(t) = a_1 t + a_0 + e_t \tag{3.6}$$

The additive error terms, e_t, are assumed to be $i.i.d$ and the error density $N(0, \sigma)$ for unknown σ. We use least square estimates of the linear coefficients.

This model is interesting because the grouping by similarity will take into account the static (parameter a_0), but also the dynamic properties (slope a_1 and variance σ) of the color content of the video.

The trend of the whole video will then be modeled as a changing linear regression model with piecewise constant parameters. We need to estimate the number of segments G and the sequence of changing points $s = (s_1, ..., s_G)$.

3.2 Optimal partitioning by complexity analysis

We present here how the video is partitioned into a series of homogeneous segments. We first define and model what we mean by "homogeneous". Then, we will present a criterion based on compact coding theory that we will minimize in order to infer the number and the location of change-points within the video document. Finally, we also give simplifying heuristics chosen in order to make the algorithm more efficient.

3.2.1 Information-based partitioning

The segmentation problem is about finding the partitioning that best explains the data assuming a model $y_m^\theta(t)$ with different parameters $\theta = (a_0, a_1, \sigma)$ in each segment.

In the literature, such estimation problems can be tackled by using a statistical criteria such as Maximum Likelihood (ML) or MAP. The problem is then reduced to the minimization of the sum of the squared residuals after fitting our model. This is minimized for the degenerated solution where each frame is considered as a partition. Therefore a penalty term is needed to constrain the complexity of the partitioning. Several criteria have been defined to solve such problems. A popular criterion is the Minimum Description Length (MDL) which has been used by several researchers for the segmentation of image data [96], [97]. We propose to use the MML criterion [87] to infer the number of segments and the location of the cut-points from our dissimilarity profile describing the content-based evolution of the video. The MML criterion has experimentally shown to be more powerful to accurately locate the boundaries of the segments than other criteria such as the MDL, Bayesian Information Criteria (BIC) or Akaike Information Criteria (AIC) on temporal univariate data [98]. It was originally tested on the Lake Huron dataset which collects monthly water-levels of the lake Huron. The MML is based on the compact coding theory . The idea is that the best explanation of the data is the one that provides the briefest encoding of a two-part message. The first part contains the information about the statistical model while the second part contains the remaining information needed about the data assuming the model. This is a quantification of the trade-off between the model complexity and the goodness of fit. The idea is that the partitioning to be preferred is the one that best fits the data using a model as simple as possible. The code length of the messages are computed using Shannon's theory where the length of the string coding an event E in an optimally efficient code is given by $-log(p(E))$.

The message length used to calculate the expected length of a message which transmits the model and the data of the jth segment containing the time series $y = (y_1, ..., y_n)$ can be approximated according to [99] by:

$$ML(\theta)_j = C(\theta)_j + D(\theta)_j \tag{3.7}$$

where $C(\theta)_j$ is a penalty term and $D(\theta)_j$ is the data fitting term.

$D(\theta)_j$ corresponds to the ML (Maximum-likelihood) estimator for i.i.d Gaussian errors. The minus log-likelihood is minimized when the model best fits the data. $D(\theta)_j$ corresponds to the code length of specifying the data assuming the model.

$$D(\theta)_j = n\log(\sqrt{2\pi}\sigma_j) + \frac{1}{\sigma_j^2}\sum_{t=1}^{n}(y_t - a_{1j}t - a_{0j})^2 \tag{3.8}$$

As it stands, this quantity will be minimized for the degenerated solution where each sample is an independent segment. There is therefore a need for a second term for our solution to follow the parsimonious principle.

$C(\theta)_j$ is a penalty term that takes into account *a priori* information P_r and a code length on the cost of specifying the model on the given set of data of length n. This code length is related to the uncertainty of the estimation of each of the model parameters. The standard error in the estimated variance is $\pm\frac{1}{\sqrt{n-2}}$ and the associated code length is $\frac{1}{2}\log(n-2)$ assuming an uniform distribution. The standard error in the estimated linear coefficients is $\pm\frac{\sigma}{\sqrt{n-2}}$ with an associated code length of $\frac{1}{2}log(n-2) - log(\sigma)$ for each coefficient.

$$C(\theta)_j = -\log(P_r) + \frac{3}{2}\log(n-2) - 2\log(\sigma) \tag{3.9}$$

P_r is a prior information that we will design in order to meet our requirements. The *a priori* information will depend on the length of the segment to penalize the creation of too small partitions.

$$P_r = \sum_{w=0}^{\lambda(k)} \frac{\alpha^w}{w!}e^{-\alpha}. \tag{3.10}$$

The parameter α is chosen such that the prior reaches a non-informative value after a given

number of frames ; we chose 5 for example. The associated code length is the negative log of the probability.

The partitioning $s = (s_1, ..., s_G)$ containing G segments that maximizes the homogeneity of the data according to the model y_m^θ also is the one that minimizes the total message length:

$$ML_{total} = \log^*(G) + \log(\binom{K-1}{G-1}) + \sum_{j=1}^{G} ML(\theta)_j \tag{3.11}$$

where G is the number of partitions and K is the total number of frames of the video, $log^*(G)$ and $log(\binom{K-1}{G-1})$ are the code length needed to specify the number of segments and which particular partitioning has been chosen assuming that all of them had the same probability.

3.2.2 Global minimization of the criteria

The problem is now to conduct the optimization in order to get the best partitioning of the ordered set of K numbers into G contiguous groups. This is a combinatorial problem and there are $\binom{K-1}{G-1}$ possibilities to explore.

This problem has been solved in polynomial time by W. D. Fisher [100] using a Dynamic Programming Algorithm (DPA). The search algorithm is based on the optimality principle that states that in an optimal sequence of decisions, each subsequence also must be optimal. The time complexity of the DPA is reduced because the optimal solution is a combination of optimal solutions of sub instances. For a set of K numbers and a maximum number of groups G_{max}, the time complexity is $O(G_{max}.K^2)$.

The MML/DPA strategy presents two very interesting advantages over other segmentation techniques like agglomerative or greedy clustering strategies:

- global optimization: every possible partitioning is taken into account during the minimization process and there is no risk to end in a local minima ;
- parameter free: no threshold is needed to stop the clustering process and no need to specify the final number of partitions. This is theoretically more satisfactory.

The number G of partitions and the locations of the boundaries are then computed, and we know that this partitioning and this number of partitions will maximize

Figure 3.2: Keyframes extracted from the "tennis" sequence containing two shots and where the first shot is segmented by our algorithm into two segments.

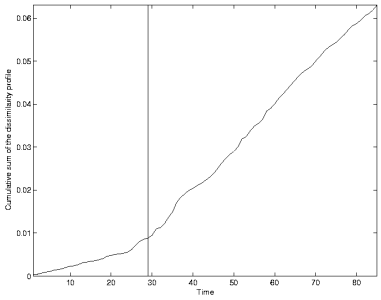


Figure 3.3: Trend of the dissimilarity profile for the first shot of the 'tennis' sequence and partitioning obtained by our algorithm.

the homogeneity of the data in each partition according to our model. All the same the computational complexity is still too high to analyze globally entire real life videos. We will use simplifying assumptions in order to restrict the search when necessary and to make it as efficient as possible. Examples of segmentation are shown in figure 3.2, 3.3, 3.4 and 3.5. Keyframes associated with the homogeneous segments, the dissimilarity profile and the associated segmentation are shown for these simple examples. The 'Tennis' scene consists in two shots. The first shot shows the first player and the second shot the second player. However our algorithm detected three different homogeneous segment, because the first shot is composed by two different camera actions: a slow zoom out filming the ball and then a fixed camera on the moving player. The trend of the dissimilarity profile is changing according to these two camera actions and the segmentation detects the transition. In the 'Ariel' sequence, one can see that the changes in the dissimilarity profile are detected and correspond to smooth dissolve transition in the video document. The third and the fourth segment of the 'ariel' sequence corresponds to the disappearance of the dog out of the scene. It shows that the appearance/disappearance of large objects have a large enough impact on the dissimilarity profile and will be detected by the segmentation algorithm.

Figure 3.4: Keyframes extracted from the “ariel” sequence where the transitions are very smooth dissolve effects.

(a) (b)

Figure 3.5: (a) Dissimilarity profile of the “ariel” sequence. (b) Trend of the dissimilarity profile and partitioning of the ’ariel’ sequence.

3.3 Restricting the search for the solution

In practice, we use two simple ideas in order to significantly speed-up our segmentation by reducing the number K involved in the computational complexity of the Dynamic Programming algorithm. The main idea is that we should perform an information-based segmentation only when necessary, between two abrupt transitions and only if something is going on. It makes it usable for partitioning of large video collections.

3.3.1 Hardcut detection

An abrupt transition between two different shots is relatively easy to detect in the dissimilarity profile. It is likely to correspond to a strong peak and we use statistical detection theory (see 4.2.2) to detect their presence.

As we know for sure that these strong transitions will be present in our final solution when using the DPA, the minimization can then only be performed within these cuts. It makes our optimization less dependent on the total length of the document, but simply on the typical length between two cuts. For a video containing K frames and containing X sub-sequences separated by abrupt transitions of length $(K_1, ..., K_X)$, the time complexity

is reduced from $O(G_{max}K^2)$ to:

$$\sum_{x=1}^{X} O(G_{max}K_x^2) \tag{3.12}$$

with each $K_x \ll K$.

3.3.2 Greedy grouping

The second preprocessing uses the fact that it seems unnecessary to perform our minimization when it is obvious that nothing happens in the video stream.

We will group together sub-sequences of the video where nothing significant is happening using a very sensitive greedy algorithm and a threshold. Groups of "still" frames will be regarded as one frame. A significant speed-up is then obtained by starting the minimization of our criteria on a new set of K_{over} sub-sequences such that $K_{over} \ll K_x$. The over segmentation is found in a very sensitive way such that it does not reduce the optimality of the segmentation, but avoid useless computations. Examples of typical numerical values for K, K_x and K_{over} for a 30 minutes broadcast news are $K = 20000$ frames in total, $K_x = 400$ frames per shot and $K_{over} = 100$ group of still frames per shot.

3.4 Summary

In this chapter, we proposed to decompose the video stream into color homogeneous segments using a full complexity analysis of the video content. We first abstract the video content by a color dissimilarity profile which is then divided into dynamically homogeneous segments. We have described an offline temporal segmentation algorithm based on the minimization of an information-based criterion. Our framework offers the advantages over existing approaches to be global and parameter free. The MML criterion efficiently constrains the maximum-likelihood estimation and offers the possibility to incorporate *a priori* knowledge. The minimization process is global and fast by using the characteristics of video data like the presence of cuts and redundancies.

The obtained segments can then be used as building blocks for further analysis for two main reasons. The first reason is that the decomposition is highly informative, because it is based on the complexity of the data. If a shot is complex and shows various different actions, it will be decomposed accordingly. If a shot is simple, it will be considered as a single homogeneous segment. Secondly, the decomposition is useful, because it significantly

reduces the complexity of the raw video stream and thus facilitates further processing. The information-level decomposition will enable the development of new applications such as user adaptive summarization and in addition the generic detection of shot boundaries as we will see in the next chapter.

It should be noted that the segmentation methodology is not dependent on the modality. If a dissimilarity profile is computed on the basis of audio or textual data, a set of homogeneous segments can be infered automatically in the same manner. Nevertheless, the features and similarity measures to choose to obtain a meaningful dissimilarity profile for these modalities is a complex task which has not been treated in this work.

Chapter 4

Applications based on the Information-level segmentation

Using the algorithm presented in the previous chapter, some knowledge about the structure of the raw video has emerged and many problems are now simplified. In this chapter, we demonstrate that this information-based temporal segmentation offers significant advantages for two different applications.

First, an application of video browsing is presented using a multi-scale key-frame selection technique. The coarseness of the video overview is chosen by the user who has the possibility to adjust a scale parameter. The various information needs find here a solution by adaptively improving the trade-off between browsing speed versus accuracy. The atomic segments are used as the ground layer and a hierarchy is build by grouping first the segments which are the most similar in content and in dynamic.

The second application presented concerns the detection of shot boundaries. We show a simple and generic technique to detect abrupt or gradual transitions separating shots irrespective of the kinds of special-effects involved. We will present a state of the art of methods dedicated to partition a video into shots. We will define the different types of transitions. Two types can be distinguished: abrupt or gradual. The problem of gradual transition detection is known to be hard and we attempt here to provide a principled solution. Moreover we will discuss techniques dealing into the image domain as well as the compressed domain. The problem of threshold selection is a common issue for all these approaches and is inherent to such detection problems. We will see how our information-based segmentation is helping us to define a generic algorithm for all kind of transitions

and we will detail and compare our performances with the TRECVid community.

4.1 Adaptive summarization by multi-scale key-frame selection

When reviewing collections of videos, users are often interested only in an overview of these documents. The aim of video browsing is to assist users to locate specific video passages quickly and provide them with visual summaries of the videos by an interactive process. Key frames are commonly used to summarize a time unit of video and also to provide an access point. Properly selected key frames can help the browsing.

Most of the related work has concentrated on breaking video into shots and then finding key frames corresponding to those shots. Many of the systems described in the literature use a constant number of key frames for each detected shot. Tonomura *et al.* [101] proposed the simple idea to use the first frame of each shot as a key frame. Ferman *et al.* [102] use clustering on the frames within each shot. The centroid (the frame closest to the center of the largest cluster) is selected as the key frame for that shot. Other authors [103] proposed to create mosaic images to cover in one image what is seen by a camera in motion. Different authors have been interested to select key frames in term of visual quality. Günsel *et al.* [104] selects the first clean frame of each shot as a key frame. Pushing the idea even further, algorithms have been designed to attempt to discover which key frame is more "important" or "interesting" in a given shot. Of course, these notions have to be defined in a restrictive way so that a computer can apprehend them. For example in the Manga system [105], key frames are selected using an importance score based on shot length and rarity within an agglomerative clustering framework.

As we presented it in the previous chapter, a shot can be complex and several key frames can be necessary to summarize a given shot. The simplest method for key frame generation is uniform sampling. Although simple and computationally efficient, sampling based methods may produce no key frame for a short yet semantically important segment, whilst producing too many key frames with identical content to represent a long static segment, thus failing to effectively represent the actual video content. As mentioned in the previous chapter, such a method does not take into account the data aspect of the problem.

When it is considered that the number of key frames is decided before analysis, key

frame selection is treated as an optimization problem to locate the key frames respecting a set of summarization constraints. In the literature, the summarization has been formulated according to several distinct criteria: sufficient content change, maximal coverage , reconstructive power. For Zhang *et al.* [106], the optimization is sequential and based on the sufficient content change. A new key frame is selected if its visual content has changed significantly from the current one. It is a method of choice when speed and simplicity is important: it possess the ability to provide a variable number of key frames depending on the dynamic progression of the visual changes. The definition of a threshold is required to quantitatively indicate what is a sufficient change. This threshold can be tuned so that the pre-defined total number of key frames is found. Different researchers focused on getting the maximum frame coverage. In [107], the maximal coverage is obtained by a greedy approach that extracts the frame with maximal coverage at each step, removes the frames that are now covered from the set of frames to cover until no frame is left to cover. In [108], the summarization constraints are formulated according to a criteria called the Shot Reconstruction Degree which measures the capability of the key frame set for reconstructing the original video sequence. It is quite close to our modeling of the dissimilarity profile into a piece-wise linear model, but here it is modeled as a piece-wise constant model. The breakpoints are found so that the pre-determined number of key frames reduce the shot reconstruction error as much as possible iteratively. All these approaches formulate in a different manner the same need to take the visual content into account, but do not propose any solution to estimate the total number of key frames.

One way to go even further is to let the number of key frames be determined by the level of visual change itself within each shot. The problem becomes more complicated: the aim is to find the number of key frames AND to select them. In [109] or [64], every shot is allocated a fraction of the given N frames depending on the cumulated frame-by-frame difference of the shot relatively to the cumulated frame-by-frame difference of the whole video. In this case, more dynamic shots are allocated more key frames, but the total number of keyframes is still pre-defined. For estimating it, a global criteria needs to be optimized constraining the complexity of the key frame representation and verifying the summarization constraints. The reader will notice that a proposal for such a complexity analysis was proposed in the previous chapter. The segmentation obtained at the information-level is an important step to get a rich set of key frames taking wholly into account the data aspect of the key frame selection problem. However, depending on the information need of our user, selecting one key frame per homogeneous segment may not be an optimal solution. If the user is only interested by a rough visualization of the content,

so many key frames representing the video with high accuracy is not wished. Then, we would like to let the user choose the level of accuracy of the key frame representation of the video. The reviewed systems do not meet our goal of extracting an adaptive number of representative key frames which may vary depending of the data aspect of a given shot and of the information need of the user. In the following, we propose to solve all these issues by performing a multi-scale temporal segmentation of the video followed by an informative key frame selection that has the property to be persistent through different scales.

4.1.1 Multi-scale segmentation

In a similar manner as the Scale-Sets representation of images of Guigues *et al.* [110], we build a hierarchy of coarser and coarser segmentations of the video by grouping contiguous segments.

Considering a partition a, a generalization of the criteria of the equation (3.7) is to add a real and positive parameter λ that we will call the scale parameter such that:

$$ML_a = \lambda C_a + D_a \tag{4.1}$$

As λ is increasing, we constrain more and more the penalty term of the criteria, the number of found partitions G is then decreasing. We thus obtain a progressive simplification of the segmentation until only one segment remains. Hence, λ can be seen as a scale parameter as in regularization algorithms. By controlling the scale parameter λ, the user will be able to adapt the segmentation to its information needs.

We will build a binary tree to represent the progressive merging of contiguous segments when the scale λ grows to infinity. Considering two contiguous partitions a and b, the scale of appearance of a node x in the tree that group these partitions together is defined as λ^+ which is the unique solution of the affine function:

$$ML_x(\lambda^+) = ML_a(\lambda^+) + ML_b(\lambda^+) \tag{4.2}$$

$$\lambda^+ = \frac{D_a + D_b + D_x}{C_a + C_b + C_x} \tag{4.3}$$

Considering all the partitions obtained for the complete video with $\lambda = 1$, we compute the scales of appearance of all possible grouping of contiguous segments and choose to group together the pair of segments having the minimum scale of appearance. We iterate

with the newly formed set of partitions until only one segment remains. The hierarchy is progressively built by augmenting the regularization constraint and we get at each scale the partitioning that minimizes globally our criterion. We show an example of such a tree in Figure 4.1.

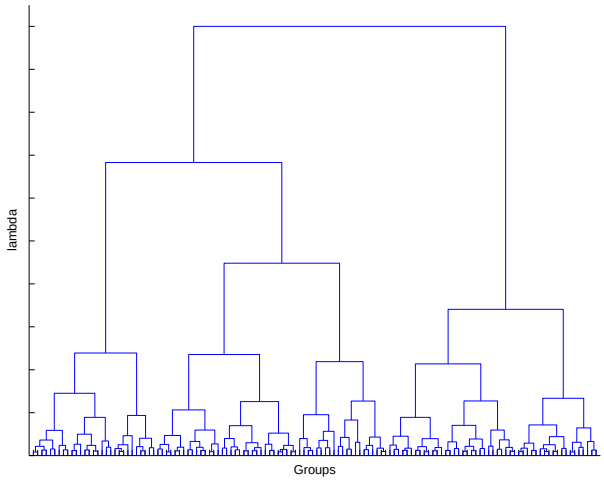


Figure 4.1: Binary tree showing the hierarchical grouping of contiguous segments when λ grows for the "AIM1MB03" video.

4.1.2 Key frame selection

In our framework, the most informative level is found by choosing one key frame per homogeneous partition for $\lambda = 1$. Considering every homogeneous partition, the key frame is selected as the frame having a color histogram which is the centroid of the segment.

When the scale parameter λ grows, we decimate our set of key frames by choosing the most informative ones. For every merging of two homogeneous partitions, we choose the key frame that is the closest to the centroid of the newly merged segment. This guarantees a persistence of the key frames throughout the different scales.

There are many further possible ways to use the hierarchy we have built. The user can interactively change the scale parameter and see the progressive simplification of the key frame representation of the video. Furthermore the user may specify the number of key frames K (s)he wants to see ; the associated scale parameter λ can then be deduced.

The different overviews (see Figure 4.2 and 4.3) presents the keyframes selected for two different videos. Three different scales have been chosen to reduce the total set of

keyframes to 400, 200 and 100. Only the first 30 keyframes are presented in the figures. The representation obtained at different scales are interesting because the segments that have the most similar dynamic behavior (eg. measured by the slope and the variance of the linear model) will be grouped together first. It means that if a keyframe disappears, this is because its homegeneity in term of the trend of the dissimilarity profile is higher. The representations do exhibit interesting semantic properties. For example, in news videos, the static studio settings with the anchor person segments will be unlikely to be merged together with dynamic outdoor segments as can be seen in Figure 4.3. The user may skip several levels of details, but still preserve the global structure of the news report with the successive apparitions of the anchor person. The general features of this video overviewing methodology are the following:

- representation is adaptive to the video content to the video content (low activity videos will require less key frames than high activity videos) ;
- representation is adaptive to the user (by selecting the coarseness level which is directly related to λ).

Developing an objective evaluation procedure for a video overviewing application is complicated as has been acknowledged by other researchers in this field. Much of the problem comes from the subjectivity of users and the variability of information needs. Therefore the absence of standardized metrics is due to the difficulty to define a proper solution.

Different research groups have attempted to make use of characteristics of algorithms that can be quantified and measured. Towards that end, two types of metrics have been devised: 1) Significance Factor is used to denote the significance of the content represented by each keyframe; 2) Compression Factor is used to quantify the reduction of information displayed comparing the set of key frames and the frame count of the original video source.

In our work, the significance and compression factor will be dependent of the scale of representation. The ratio between significance and compression is approximately constant for all scales by design: the minimized global criteria ensures that all scales offers a higher significance at higher compression or higher compression at lower significance. However such factors are a quite poor representation of the quality of the selection of keyframes. A qualitative evaluation is necessary. Only human users can judge by trying several competitive algorithms on a extended set of examples to determine which one is the

best. According to us, this is a too subjective problem in order to attempt a quantitative evaluation.

4.2 Generic detection of shot boundaries

In order to detect shot transitions, the simplest method is to consider a simple threshold over a dissimilarity profile as presented in the previous chapter: if the dissimilarity is higher than a given threshold, a cut is detected. The problem would be very easy if there were no:

- false signals:
  - fast objects moving or (dis)-appearing in the frames ;
  - fast camera motion (hand camera) ;
  - fast illumination changes (explosion, fire, shadows, reflections, flashes) ;
  - special effects also can be a source of problems with screen split, video in video, text overlay, etc.
- noise: MPEG errors, compression and camera noise, etc.

In this section, we will first introduce the most common shot transitions. Then we will present the advanced methods for shot boundary detection and that more advanced pattern matching methods are necessary for the detection and recognition of gradual transitions.

4.2.1 Definition of a shot transition

We will define and describe here the most common transitions between shots: the cut for abrupt transitions and a limited number of gradual transitions (fade, dissolve and others). Only the imagination of the film-maker may limit the number of special effects that can be designed to artificially combine two shots. However the most commonly used gradual transitions are fades and dissolves.

Cut: A cut is an instantaneous transition from one shot to the next. There are no transitional frames between two shots. It happens naturally when stopping and restarting

(a)

(b)

(c)

Figure 4.2: Key frame selection at different scales for the "AIM1MB03" video. The news report is summarized by (a) 397 key frames (b) 199 key frames (c) 97 key frames.

(a)

(b)

(c)

Figure 4.3: Key frame selection at different scales for a "19990201_CNN" TRECVID video. The 60 minutes news report is summarized by (a) 394 key frames (b) 197 key frames (c) 98 key frames.

the camera, the resulting transition is abrupt and the two shots are simply concatenated. We will see that it is the easier to detect because it creates a visual discontinuity in the video stream.

An important issue to take into consideration when trying to detect cuts is the presence of flash. False detection of cuts can occur due to the discontinuities introduced by their presence. In real world video, there can be many flashlights during a given period of time and this influence one or many frames. Robust methods implement a flash detection method to circumvent this problem described later in the chapter.

Fade: A fade is a gradual transition between a shot and a constant image (fade-out) or between a constant image and a scene (fade-in). Often the constant image is the black screen where all pixels are black. During a fade, images have their intensities multiplied by some value α. During a fade-in, α increases from 0 to 1, while during a fade-out α decreases from 1 to 0. As illustrated in the Figure 4.4, a fade-out is characterized by the progressive modification of all pixels to become a constant image (black in this example). The choice of the constant image and of the speed of modification will vary from one transition to the next.

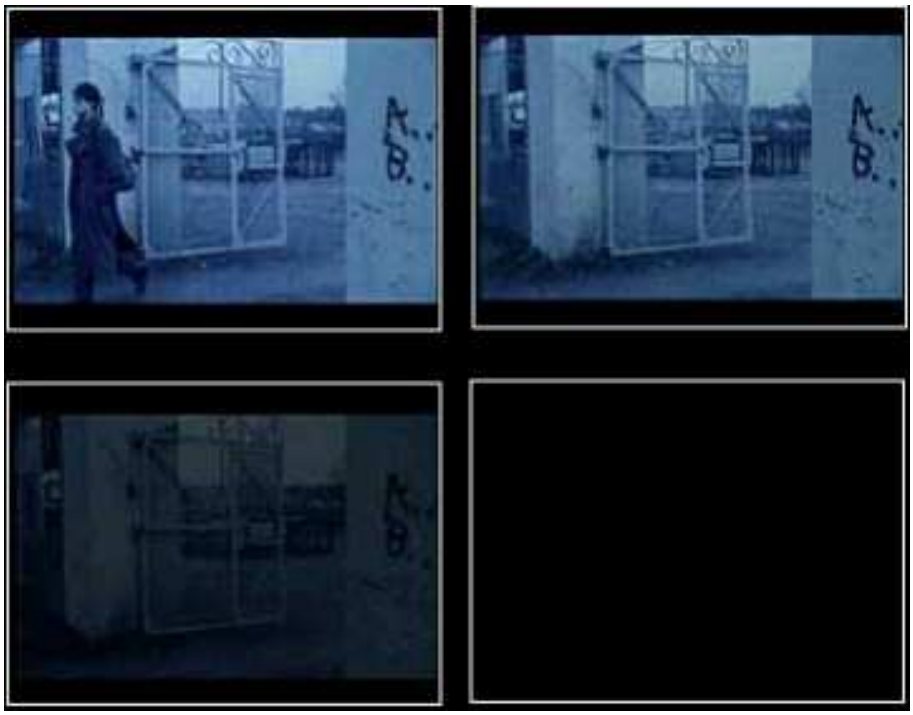


Figure 4.4: Illustration of a fade-out transition.

Dissolve: A dissolve is a gradual transition from one scene to another in which the first scene fades out and the second scene fades in. As shown in the Figure 4.5, a dissolve is characterized by the progressive modification of all pixels to become an image of the next shot.

Figure 4.5: Illustration of a dissolve transition.

Wipe: A wipe is a transition where a line moves across the screen with the new shot appearing behind the line.

Miscellaneous: An unlimited number of special effects can be designed, but are typically less used that the types of transitions we already mentioned. Typically, cuts, fades, wipes and dissolves represents 95% of the transitions found in broadcast news material [111]. We will cite two more examples of fancy special-effect transitions. A turning page is an effect that makes the transition appear as if somebody was turning the page of a book. Morphing is another popular effect where the image of an object or of a person is progressively modified to something else. The shape of objects as well as the pixel intensities are smoothly modified to achieve the effect.

Hampapur *et al.* [112] proposed an interesting classification for transitional effects (see table 4.1). The transitional effects can be classified into 4 categories: transitions with no manipulation such as the simple cut, with a manipulation of the coordinates of the pixels, with a manipulation of the intensity levels of the pixels and a mixed mode where intensities and spatial coordinates of pixels are modified.

Edit Type	Meaning	Examples
Null	Concatenate	Cut
Spatial	Manipulate pixel space	Translate, Turning page
Chromatic	Manipulate intensity space	Fade, Dissolve
Combined	Manipulate pixels and intensities	Wipe, Morphing

Table 4.1: Classification of transitional effects in term of features.

4.2.2 Overview of existing shot boundaries detection methods

Cut detection

Simple thresholding: We have already seen that the detection of a cut can be made by a simple thresholding on the dissimilarity measure. The choice of the threshold T is difficult, because even with a robust dissimilarity measure the variance of camera operations and object motion may exhibit temporal variances of the same order than for a cut. Adaptive methods are then better suited to solve our problem.

Adaptive thresholding: Adaptive thresholding makes use of statistics on the considered neighborhood to place a threshold which will adapt locally to the variance of the dissimilarities. A simple way to perform adaptive thresholding is the following: a moving window is used on the dissimilarity profile and a shot is detected if a maximum is detected and is significantly higher that the relative difference to the mean value exceeds a fixed threshold. Such methods are very popular for their low computational cost and good results considering only cut transitions (see [113], [114]). In the recent TRECVid 2005 event, the reference shot boundary detection has been computed with adaptive thresholding for cut detection [115]. Good results are easily obtained for hardcut detection, but such methods are inefficient to detect gradual transitions.

Statistical detection theory: A more formal and statistical way to resolve the problem of detection is to formulate it according to the statistical detection theory [116], [117]. The

cut detection can be formulated as a binary-hypothesis test problem for the input variable z corresponding to the values of the dissimilarity profile in our case. We introduce two hypotheses:

- hypothesis S: there is an abrupt transition present between frames k and $k+1$;
- hypothesis $\bar{S}$: there is no transition present between frames k and $k+1$.

The test can fail if we make a false detection (i.e. S is chosen when $\bar{S}$ is true) or a missed detection (i.e. $\bar{S}$ is chosen when S is true).

A well-known result in hypothesis testing is that the following decision rule is equivalent to the minimum risk of error:

$$\frac{p(z|S)}{p(z|\bar{S})} \begin{array}{c} \bar{S} \\ < \\ > \\ S \end{array} \frac{1-P_k(S)}{P_k(S)}. \tag{4.4}$$

The likelihood functions $p(z|S)$ and $p(z|\bar{S})$ and $P_k(S)$, the probability for the validity of S, can be modeled for cut detection.

The shape of the distribution of the values of the distances when a transition is present is plotted using training data consisting of broadcast tv shows exhibiting a large diversity of content. Experimentally, the likelihood function $p(z|M)$ is found to be correctly approximated in the family of Gaussian functions:

$$p(z|M) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{4.5}$$

Using the same method with the distribution of the values when no transition is present, the likelihood function $p(z|\bar{M})$ was found to belong to the family of Exponential functions:

$$p(z|\bar{M}) = e^{-hz} \tag{4.6}$$

$P_{k,l}(M)$ is then defined as a conditional probability $P_{k,l}(M|D_{seg}(l))$ that depends of the dissimilarity of the frames before and after the transition.

$$P_{k,l}(M) = P_{k,l}(M|D_{seg}(l)) \tag{4.7}$$

The conditional probability $P_{k,l}(M|D_{seg}(l))$ (see Figure 4.6) is computed using the similarity measure before and after the boundaries of the segments. The conditional probability should not be too sensitive when $D_{seg}(l)$ has extreme values. Between these values, the transition should be smooth to avoid the rejection of good candidates and this is the reason why $P_{k,l}(M|D_{seg}(l))$ can be chosen as:

$$P_{k,l}(M|D_{seg}(l)) = \frac{1}{2}\left(1 + erf\left(\frac{D_{seg}(l) - d_c}{\sigma_c}\right)\right) \tag{4.8}$$

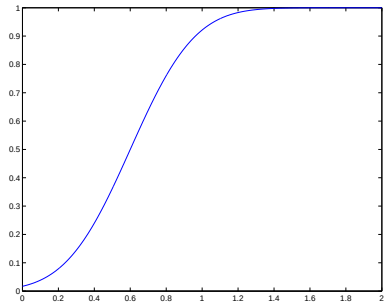


Figure 4.6: Plot of the function used to design the conditional probability $P_{k,l}(M|D_{seg}(l))$.

Gradual transition detection

Dissolve detection: Dissolve detection is a difficult problem, and the performance of state of the art methods is still far below the results obtained for cut detection. The major problem is that dissolve detection possess more degrees of freedom, they are characterized by three parameters: speed, linearity and evenness.

First, the speed of the dissolve will vary from one transition to the next. We will denote t_1 and t_2 the instant when the gradual transition starts and ends. The length of the transition $t_2 - t_1$ will vary due to the fact that slow and rapid dissolves are observed depending on the general rhythm of the scene. Then, one may assume the linearity of the transition which can be formulated in the following manner:

$$I(x,y,t) = I_b(x,y,t) + \alpha(t) I_a(x,y,t) - (1-\alpha(t)) I_b(x,y,t) \tag{4.9}$$

$$\alpha(t) = \frac{t - t_1}{t_2 - t_1} \tag{4.10}$$

where $I(x,y,t)$ represents a pixel value at time t and location (x,y) during the gradual transition. I_b and I_a are the frames of different shots located before and after the gradual transition. $\alpha(t)$ is in this model a linear function of the time. An issue is that in practice the function α might not always be linear. It can take any shape, but a frequent example is a parabolic shape in order to give the feeling of acceleration. Evenness is a third parameter characterizing dissolve. It measures the variance of difference images. During a dissolve transition, the changes are more evenly distributed in the frames than during camera or object motion.

For the CLIPS laboratory in [93], the property of linearity is used. Dissolves are detected if the $L1$ norm of the first image derivative is large enough compared to the $L1$ norm of the second image derivative in order to verify that the pixel intensities follow a roughly linear law depending on the frame number. For [118], a dissolve will be found if the linearity hypothesis holds as well. For this, the normalized linear error is computed. This measure equals 0 if the transformation is linear. A dissolve is detected if the normalized linear error is lower than a threshold T and if the change during a large number of frames is significant enough (25 frames as example). Petersohn [119] goes even further in the modeling of the properties of a dissolve. He proposed a methodology which is based on 6 successive stages: selection of candidate by matching a U-shape in a edge dissimilarity profile, check image differences several frames apart to determine t_1 and t_2, check histogram and motion compensated image differences, check for linearity of the dissolve, check for the evenness of the changes and check for the global motion. The edge energy dissimilarity profile is computed on successive frames and the U-shape expresses the fact that, in the middle of the transition, the edges lose in sharpness more than before and after the dissolve.

Wipe detection: Wipe detection is a quite complex problem. In [120], Petersohn exploits the fact that the amount of changes is very concentrated spatially where the line is present. This corresponds to a minimum in evenness. A Hough transform is used to detect linear segments in the difference images. The transforms are added and binarized. Patterns in the movements of the boundaries during the wipe are detected by a second

Hough transform. The method uses all the patterns which are characteristics of a wipe effect to enhance the detection. However, it also makes the algorithm sensitive to noise.

Artifacts

Flash removal: We will conclude this subsection by considering the problem of flash removal. Flashes are a problem when trying to detect shot transitions, because their presence creates peaks in the dissimilarity profile leading to false detections. Therefore, it is required to detect and remove flashes as valid candidates for shot transitions. A flash is defined as a high-intensity light of very brief duration, this definition makes them quite easy to discriminate them from the data. We will use the following facts:

- the duration of a flash is very short (1 to 3 frames) ;
- the frames before and after are likely to be very similar ;
- the frames become white during a flash (near saturation of the high-end of the range of possible pixel values).

We denote $D(I_i, I_j)$ the dissimilarity between the frame i and j. For a given candidate t for a cut transition, we consider the values of $D(I_{t-2}, I_{t+2})$, $D(I_{t-1}, I_{t+2})$, $D(I_{t-1}, I_{t+1})$, $D(I_{t-2}, I_{t+1})$. A simple and robust method is the following: if any of these values is low enough, it is possible to remove the candidate for cut detection.

4.2.3 Assumptions for generic detection

The vast majority of research groups are focusing their efforts on cuts, dissolves, fades and wipes, because these are the most commonly found transitions in the majority of documents. It is reasonable to focus on them, but in addition it should be considered that the number of possible transitions is endless. We believe that it is possible to find shot transitions without any particular modeling on the specific features of every possible transitions, but simply by making a set of minimal assumptions. Many articles propose to design a specific detector for each type of special effect transitions [121], [122]. Here, we rather depart from this solution in order to avoid ad-hoc techniques.

Within our framework, the detection of shot boundaries is reduced to the merging of the segments that are not due to a transition between shots. We essentially face two kinds

of artifacts: signal level noise (MPEG or optic of the camera artifacts) and noise that has a semantic meaning (for example when someone passes just in front of the camera). In order to perform a reasonable detection of the transitions, we use the following minimal assumptions:

- the lifetime in number of frames of a transition is included in a range between 1 and 20 frames ;
- the visual content of the video before and after a transition separating two different shots is usually significantly different ;
- we use the homogeneous segments found by our information-based segmentation algorithm to reduce the number of candidate points for shot transition.

4.2.4 Statistical detection framework

We consider the frame content similarity before and after the boundaries of the homogeneous segments. If k is the index of the frame that separates the segments i and $i+1$, we use different windows of comparisons $k-l$ and $k+l$ for $l = 1, 2, 4, 8$. This informs us as to whether the transition is abrupt or gradual and gives an estimate of the length of the transition.

To detect such transitions, it will be important to calculate the dissimilarities taking spatial information into account. A robust similarity measure that uses differences in color and spatial distribution of pixels information is the color-block histogram of the frames. We define the distances $D_{seg}(l)$ between the frames $k-l$ and $k+l$ as a vector containing the Jeffrey divergences of the color block histograms in the YUV color space and 4 rectangular blocks.

We use statistical detection theory (see 4.2.2) to make a detection which minimizes the minimum risk of error. The estimation of an approximate value for the length of the transition is found by finding the minimum l such that the detection of a transition is positive.

Using training data (see next paragraph), the parameters h, μ, σ are estimated and d_c and σ_c are chosen in order to maximize performances.

4.2.5 Experiments

Evaluation protocol

The evaluation protocol consists in comparing the results of shot boundary detection according to a ground truth (see [123] for a tutorial). We define:

- N_t: the total number of transitions in the ground truth ;
- N_d: the number of transitions missed by the algorithm, but existing in the ground truth (missed detections) ;
- N_i: the number of transitions detected by the algorithm, but not existing in the ground truth (false detections) ;
- N_c: the number of transitions found by the algorithm and existing in the ground truth (correct detections).

The reader may notice that $N_t = N_c + N_d$. Consequently, the following indicators are used to measure performance:

$$\text{Recall} = \frac{N_c}{N_t} \tag{4.11}$$

$$\text{Precision} = \frac{N_c}{N_c + N_i} \tag{4.12}$$

$$\text{F-Measure} = 2\frac{\text{Recall.Precision}}{\text{Recall} + \text{Precision}} \tag{4.13}$$

Precision and recall are standard evaluation measurements in the context of information retrieval. The precision measures the number of the correct transitions found by the system as a fraction of all transitions found by the system. A low precision value indicates a high amount of "noise" introduced by the automatic segmentation. The recall measures the number of correct transitions found by the system as a fraction of all transitions existing in the ground truth. A high recall is necessary to have a usable system. As an illustration, popular search engines have generally a very high recall but a poor precision. Many results are shown to the user and he has to filter out what is not relevant to

his query. The F-Measure combines precision and recall as their harmonic mean. Improving recall and improving precision is the final goal of our research, but efforts to improve one often degrade the other. Different trade-offs can be proposed favoring one measure over the other. By focusing on improving the F-measure, we know that we achieved the development of a better system in terms of precision and recall simultaneously.

Large scale evaluation

In order to test our algorithm, we performed a large scale evaluation with 70 videos of the TREC Video Retrieval Evaluation (TRECVID) 2003 corpus using the evaluation framework of [123].

We used 35 hours of news programs coming from CNN and ABC. Each video contains around 400 transitions of every kind. There are many types of special-effects involved because the videos come from television. The results are given in percentage in Table 4.2. We chose to accept a tolerance of 12 frames for the accuracy of the location of the transitions.

Performances	Our algorithm	Cuts detection alone
Recall	86.2	67.6
Precision	77.2	78.8
F-measure	81.45	72.7

Table 4.2: Global performances of the shot boundaries detection algorithm when compared with cuts detection alone for all transitions.

The comparison in the table 4.2 shows the advantage of using the information-based segmentation over the simple "cut" detection which misses a significant number of transitions. It shows that the number of gradual or smooth transitions can not be neglected without a serious loss in performances.

Performances	BIC	MDL	MML
Recall	83.1	83.6	86.2
Precision	74.1	74	77.2
F-measure	78.34	78.51	81.45

Table 4.3: Comparison of shot detection performances using BIC, MDL or MML criterion.

The table 4.3 shows that MML slightly outperforms other popular regularization terms such as the Bayesian Information Criteria (BIC) and the Minimum Description

Length (MDL). MML empirically seems to be good choice for such segmentation problems. From the practical point of view, MML has the advantage to incorporate "a priori" information that we embedded in the regularization term.

Performances	6 frames	12 frames	24 frames
Recall	83	86.2	88.2
Precision	77.7	77.2	73.5
F-measure	80.26	81.45	80.18

Table 4.4: Comparison of shot detection performances using various tolerance windows.

The table 4.4 shows how the performances change when the tolerance window is modified. The results show that the algorithm obtains the best performance in term of the F-measure measure when the tolerance window is set at 12 frames which represent half a second. This duration is in addition the typical length of gradual transitions. It is difficult to accurately locate where the gradual transition starts and stops and the ground truth itself is quite often not so accurate. The window of tolerance is necessary to obtain the best performances while staying short enough to be accepted by the user.

Performances	Cuts only	Graduals only
Recall	89	75
Precision	79	73
F-measure	83.7	74

Table 4.5: Performances of our algorithm considering cuts or graduals transitions only.

The table 4.5 shows that the gradual transitions are properly captured by the information-based segmentation, because the recall is high.

Performance comparison with TRECVID 2004 participants

TRECVid 2004 evaluation protocol: In this section, we will describe how the shot boundary evaluation has been performed during TRECVID 2004 and compare our results by using strictly the same settings. The evaluation of performances for shot boundaries detection algorithms uses 12 randomly selected ABC/CNN/C-CSPAN video data. The ground-truth has been manually annotated and is supposed to contain no errors. The total number of frames is 618409 and the number of transitions is 4806. 57.7% of the transition are cuts, 31.7 % are dissolves, 4.8% are fadein/out and 5.7 % are from other kinds.

Participants: We will detail here 3 selected participants with which we will compare the performance of the algorithm we propose. We chose these particular groups, because they are among the best participants in terms of results and for the variety of their approaches. Fraunhofer is the best performing group among all of the TRECVid 2004 participants for this task.

Fraunhofer [119], [120], [115] uses adaptive thresholding for the detection of cuts on pixel and edge differences and complex pattern matching for dissolves and wipes detection. The approach shows good performance and low computational complexity.

Clips-IMAG [93] uses a complex system with combines modules for image comparisons (motion peak detector, image difference, motion compensated image difference, peak image detector and dissolve detector) which are sent to detection modules with a fusion of the different inputs. Filters are present to remove flashes and enforce coherence. Dissolve transitions are detected by comparing norms of first and second temporal derivatives of the images.

IBM [124], [111] shot boundary detection algorithm consists in graph-based, multiple pair-wise frames comparisons. Each frame is considered as a node in the graph, a shot transition is detected if a cut in the graph is present. The comparison is based on color and edges histograms. Adaptive thresholds are used for the decisions which is embedded in a finite state machine to detect different states such as shot, cut, dissolve, fade-in and out and video errors.

We should mention that our evaluation results should be taken with caution, because our results are not based on the same ground truth than the one of the participants. The ground truth of the official TRECVid campaign was generated semi-automatically based on CLIPS segmentation tool and then checked manually. The manually checked ground truth was not provided to us. However, the accumulated errors should not be very large for the high numbers of shot we consider.

Performances: The task of TRECVID2004 gets better performance values as shown in the table 4.6 than the large scale evaluation due to the restricted number of videos and the error-free annotation. By comparison with other groups, we are leading for the detection of gradual transition thanks to our information-based framework. However our algorithm is not the best for the detection of cuts. The performance figures are not telling everything, we should now have a look at the computational complexity of the different

Performances	Fraunhofer HHI	Clips-IMAG	IBM	Our algorithm
Precision Cuts	**94**	86	89	89.6
Recall Cuts	**94**	86	89	88.8
F-measure Cuts	**94**	86	89	89.2
Precision Graduals	77.5	80	**88**	79
Recall Graduals	72.5	74	62	**80**
F-measure Graduals	74.92	76.88	72.75	**79.5**

Table 4.6: Performance comparison between selected TRECVID 2004 participants.

approaches.

Computational complexity: We measured execution time on a PC with a Pentium IV 2.0Ghz processor. The decoding and feature extraction time denotes the time it takes to read the MPEG files and extract the YUV color histograms. The results presented in table 4.7 show that our proposed algorithm is quite fast if one considers that it does perform an information-based segmentation with a global optimization of polynomial complexity.

Avg total time (in sec)	Avg decoding time (in sec)	Avg segmentation time (in sec)
6670	4600	2070

Table 4.7: Computational complexity shot boundaries detection algorithm.

In Figure 4.8, we present results obtained by other groups who published these numbers. We normalized the numbers so that it can be compared to our configuration consisting in a 2Ghz processor. Fraunhofer and IBM experimented with a 3Ghz processor, CLIPS-IMAG with a 3.2Ghz processor, therefore we normalized the values to show how it would behave on a 2Ghz machine. The complexity of the algorithm from Fraunhofer is much lower than our information-based algorithm, but it can be seen that our algorithm is slower only by a factor of approximately 2. CLIPS-IMAG and IBM uses intensively motion compensation and this is a very time-consuming process, this mainly explains the large difference between the different groups.

	Fraunhofer HHI	CLIPS-IMAG	IBM
Avg total time (in sec)	1500	24000	between 9000 and 27000

Table 4.8: Comparison of the computational complexities of various algorithms.

4.3 Summary

In this chapter, we presented an original multi-scale key frame selection system that enables a user to interactively adjust the coarseness of the video representation. The adaptiveness is complete from the data as well as from the user point of view. Due to the large and increasing amount of digital video data possessed by households (DVDs, divX, home videos), we believe reasonable to think that such video browsing softwares should become mainstream in the years to come.

Moreover we have shown that the detection of shot boundaries is highly simplified by disregarding the type of special effects involved during abrupt or gradual transitions and performs well using our methodology. The performances are good because the shot boundary detection algorithm operates on a reduced search space thanks to the information-based segmentation. Also, the approach combines a region-based segmentation algorithm thanks to the MML/DPA as well as an edge-based segmentation procedure by hypothesis testing. We have evaluated our results on large datasets and compared them with state of the art approaches. Our methodology performs quite well with a reasonable computational complexity.

Chapter 5

Segmentation at the Semantic-level

5.1 Introduction

An interesting way to present fewer and more relevant results after a query is to operate at the semantic-level. The semantic-level is constructed from the segmentation obtained at the editing-level by grouping shots with similar semantic properties. The general goal is to organize the video collection with a minimal amount of partitions while keeping all the relevant semantic information as depicted in the Figure 5.1. In this figure, the three levels are depicted: information-level in homogeneous segments, the editing-level corresponding to shot detection and finally the semantic-level. The inclusive relationship between the editing and semantic-level will be discussed in the section 5.3.2 of this chapter.

It should be noted that the capacity of humans to assimilate information at the semantic level is amazing. A human cannot recall every frame watched during a movie (as a computer does), but can recall a small number of scenes that creates a story-line. Humans have naturally the ability to extract out of large amount of multimodal data only the relevant information. This is not the case for machines. Attempts to incorporate semantic and attach high-level labels to low-level features have been proposed in the pioneering work of Vasconcelos *et al.* [125]: from the very beginning the idea was to use models for inference (Bayesian models) and to incorporate prior knowledge.

The semantic properties, needed for the definition of a criterion to group shots,

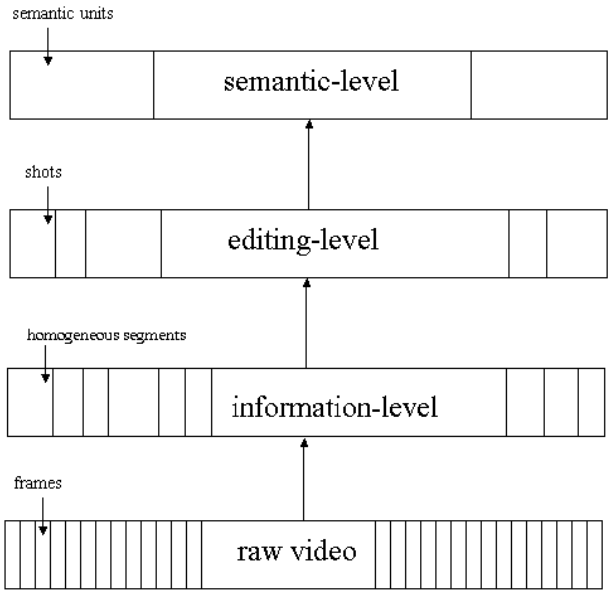


Figure 5.1: A general view of the problem of semantic video structuring.

will vary depending on the nature of the video collection and the information needs of a potential user. Therefore we face an obligation to clearly specify the scenario of application so that we may define this particular criterion.

In this introduction, we will first present how different types of video collection can be structured. Then we will address general considerations about the problem of semantic segmentation and introduce the concept of computable scene. Then we will motivate why we focused on the problem of news story segmentation and why this has proven useful to improve news retrieval applications.

5.1.1 Structuring various types of video collection

Different types of video collection have been studied. The specificities of each video collection require different solutions to structure them at the semantic-level as presented in the table 5.1.

Sport video collections are structured according to the rules of the game. For soccer videos, users are particularly interested by the detection and recognition of the important events to know which actions has been taking place during the match. An important cue

Type of video collection	Typical structure	References
Soccer	Semantic temporal structure is undefined. Users are only interested by the presence of the important events such as a goal or a fault	[126, 127]
Tennis	Hierarchical temporal structure defined by the scoring system: game, set and match	[7, 128]
Films	Sequence of scenes which are smoothly linked to each other according to a storyline, but hopefully follow patterns due to film making technique	[5]
Home video	Similar memories are usually stored in a contiguous manner. The transitions between memories and typical patterns to be found will mainly depend on user behavior with camera.	[6]
News broadcast	Sequence of stories explicitly separated and considering different topics (finance, weather, politics)	[8, 129, 130]

Table 5.1: Show how different types of video collection can be temporally structured.

comes for the audio track. For example, in [127] important events are detected by classification of audio track into excited/unexcited commentary. In [131], a common framework is proposed for the detection of events for baseball, soccer and golf. The main idea is that in these sports the crowd will show its appreciation of the action by cheering and applause. Applause and Cheering detectors are then build from audio features and significant events are detected based on this information. In [21], the video is labeled into "play" or "break" using dominant color and motion information. Domain specific knowledge is used: the field is detected with a specific feature: the grass-area-ratio. These examples show the importance of restricting the video collection for taking advantage of their unique characteristics.

Tennis video collection can be structured by the nature of the scoring system itself: game, set and match. Sudhir *et al.* [128] incorporated a field model and object tracking. In that case, all the features are visual and the information contained in the audio track is not used. Kijak et al. [7] proposed an algorithm based on domain-specific audio and visual features and a Hidden-Markov model for the classification and segmentation. Again, the knowledge of the structure of game helps to bridge the semantic gap.

Movies are implicitly structured in scenes as defined in dramaturgy. Sundaram et

al. [5] uses the knowledge about the film-making process and the psychology of audition. The proposal is valid under quite strict conditions that will often not be validated, because every film maker uses different rules and manners to create his/her production. The video collections containing drama movies are very problematic, because little domain-specific knowledge is relevant to facilitate the design of automatic algorithms.

Home video collections will vary a lot from one user to the next. Basically the home user wishes to quickly retrieve movies depicting his/her holidays from two years ago in its collection and has no time to spend on manual annotation. Particular problems appear due to the fact that home videos possess some typical characteristics such as: the absence of storyline or explicit semantic structure, the fact that scenes are often ordered in temporally adjacent shots and the important motion of the camera which is carried by the hand of the user. Structure discovery has been attempted by using a hierarchical clustering algorithm to enable the automatic creation of video summaries in [6].

News broadcast are explicitly structured as a sequence of stories covering different topics from world events to weather forecast. The structure is in itself simple, but algorithms have to capture high-level semantic information to work properly. Various approaches have been proposed using multimodal classification techniques ([130, 132]) and statistical sequence modeling with Hidden-Markov models ([8]). News stories is a type of video collection which is somewhere in between drama movies and tennis videos in term of structure: semantic ambiguity is still there, but some rules and patterns actually exist.

5.1.2 Preliminary considerations about a computable scene

Considering a video collection made of entertainment films H. Sundaram et al. [5] defined the notion of a computable scene. According to the authors, a scene is computable if it respects criteria coming from the knowledge of the film-making process and the psychology of audition. It is interesting to learn about this kind of thinking: from a given video collection, [5] defines as computable the scenes which respects criteria of homogeneity that will make it possible to develop a segmentation algorithm around it.

This concept of computable scene is important to clearly understand the limit of semantic-level analysis of multimedia documents. It should be understood that different video collections will possess various levels of computability depending on the respect of rules which renders possible for a computer to capture the cues that leads to a proper semantic grouping of shots. First, it is clear that we are not trying to develop algorithms

that actually understand the video data. Also any video collection is likely not to obey to the rules all the time, because they are edited by humans. However, we may expect to locate patterns that will help to significantly reduce the human effort needed for an error free annotation of the video content.

5.1.3 Overview of different approaches for high-level video segmentation

Explicit rules

The first category of approaches considers that the video collection will generally follow a set of predefined rules and can be modeled according to a theory such as film-making theory or psychology of the audience.

The first example of such a proposal comes from Aigrain *et al.* [133] who defined a set of nine rules coming from film-making theory and discussions with video directors and analysts. They claim that the fixed set of rules defined is sufficiently medium-invariant to enable automatic semantic segmentation. Three rules are related to transition effects, three rules to similarities between shots and their repetitions, one rule for the editing rhythm, one for the audio track and the last one for the camera motion. Possible conflicts are resolved by attributing a priority to the various indicators to deal with. The problem is that various video collections will obey different patterns and that the thresholds and priorities used will constantly need to be readjusted. Specializing in drama movies, Sundaram et al. [5] argues that it is reasonable to assume chromatic and lighting homogeneity by understanding how a director is usually placing the cameras during filming. Also experimental observations in psychology of audition encourage the authors to focus on short-term grouping strategies for sound. The segmentation is based on a memory model (FIFO) to enforce coherence due to the fact that viewers have a limited memory and attention span. Both audio and video changes are chosen as a criteria for scene change detection. Chromatic and lighting homogeneity are assumed for the visual features of a given scene. Correlations amongst the audio feature during the attention span are used to detect an audio change. Semantic constraints are also added by using the detection of dialog shots which are often recurring in drama movies. Audio and visual scenes are then found and the merging is done by using a set of heuristic rules such as:

- if there is an intersection between the ambiguity windows of the audio and visual scene, select the transition which separates the visual scenes ;

- if there is no intersection between the ambiguity windows of the audio and visual scene, select the transition which separates the audio scenes.

The visual scenes are then used for their accuracy, but the prevalence is given to the audio scenes. The reason is that film director uses relatively often the audio track to create the feeling of semantic continuity. All the same it is possible to conclude that these approaches are still far to obtain usable results due to the simplicity of the hypothesis and the complexity of the problem.

Such approaches are for the moment the only interesting ones when considering drama movies or documentaries where the scene transitions are often ambiguous even for a human observer. For more specialized video collections, we will see that more accurate models and more performance-oriented approaches are possible.

Combining information from multiple modalities

This category of approaches use information coming from multiple modalities to improve the semantic segmentation. The goal is to be more robust or meaningful than with only one modality. The first attempts in this direction are segmenting separately audio, video and text streams. For the Informedia project, Hauptmann *et al.* [134] uses speech recognition to produce transcripts and to detect silences. Audio segmentation is provided by the silences in the audio track whereas transcripts are used for the text segmentation by detecting critical keywords. Image analysis is also done by combining color histograms and optical flow to generate the visual segmentation. If a user queries the system, keywords are searched in the transcripts. Then the system searches for the visual and audio scene breaks that are matching and containing all the keywords. In fact, the purpose of the system is not really to provide a semantic-level segmentation of video content, but to integrate audio/video/text segmentation methods in order to return appropriate video clips to a user query. In Nam *et al.* [135] or Saraceno *et al.* [136], modalities are combined to reinforce the inference. When conflicts arise between the structure found by different modalities, the priority is given to audio or to visual information depending on the context of the study. No clear and convincing formalisms were proposed in these first experiments to attempt to combine the information extracted from multiple modalities to get closer to a semantic segmentation.

Using "a priori" information

When a video collection is consistent from the structural point of view from one document to the next, the use of the "a priori" information is an advantage to consider. For Günsel *et al.* [104], a semantic partitioning is achieved by using the knowledge that a particular representative logo will appear and help to discriminate between news and non-news programs. A region tracking algorithm is used for this purpose. This is an extreme example of the use of "a priori" information. The drawback is that the algorithm is so specific that it will be possible to use it for very restricted video collections. It also poses a problem of robustness, the semantic segmentation is directly related to the object recognition and tracking steps which are known not-to-be error-free.

Statistical modeling

This category of approaches does not make any strong assumption on the rules and patterns which govern the detection of a scene transition, but are aiming at discovering these structures by using a statistical model and training data. Also statistical models provide a principled way to combine modalities and let the system designer use "a priori" information combining the advantages of all the approaches we talked about previously. With all these advantages,it is not surprising that modern approaches are all based on statistical models. Statistical models are often supervised and this implies the use of training data. The training data will serve as examples to infer the optimal parameters of the statistical model. No strong assumptions is made on the content of the data, but there is the assumption that the test data on which the inference will be done is similar in terms of statistical patterns and structures to the training data.

A large amount of different studies proposing statistical models to solve all types of problem in CBVR exists. We will illustrate this section with two different examples of what can be done with a statistical model to find the semantic structure of video collections in a supervised and an unsupervised manner. Zhao *et al.* [137] defines a scene as a sequence of consecutive shots that are semantically correlated: sharing same semantics in terms of time, place, objects or events. To detect scene boundaries, the problem is then to determine if two shots are semantically correlated. A probabilistic model (Left-Right HMM) is used on color histograms to cluster consecutive shot depending on the scene membership hidden state variable. This is similar to agglomerative clustering with the exception that here the model is probabilistic and considers the order of the dataset. Of course, the use of color

information only makes the approach valid for specific video collections. For example, if the video collection only contains a very limited set of possible color backgrounds, then color histograms hold more semantic information and help to deduce semantic structure of the video document. Introduced by Fine *et al.* [138], Hierarchical Hidden Markov Model (HHMM)s have been used for automatic structure discovery of soccer video by Xie *et al.* [21]. This unsupervised approach models the structure of the video content and creates a hierarchy. It is hoped that the higher level of the hierarchy will be semantically meaningful whereas the lower level should provide a segmentation more related to the low-level features. Model selection as well as parameter estimation has to be performed altogether. The algorithm performed well for a simple problem where there were two semantic events "play" and "break" to capture in two soccer videos representing 40 minutes of data. Color and motion features are used for this purpose. The approach has the fantastic advantage of being totally unsupervised (requiring no training data), nevertheless the computational complexity is very high due to the parameter estimation of the HHMM ($O(T.Q^{2D})$ where T is the sequence length, Q is the number of states and D is the number of levels in the hierarchy) and the discovery of the structure of the model itself by Markov Chain Monte Carlo (MCMC) algorithm.

5.1.4 Motivation for our scenario of application: news story segmentation

We will explain here why we focused our efforts to an application scenario which is the problem of news story segmentation as defined by the TRECVid forum. There are several reasons which lead us to focus on this particular problem that we will list here.

Increasing news retrieval efficiency

First of all, if you consider a task where a user tries to retrieve relevant stories out of a news broadcast video collection, the baseline technique would be the linear search: the user manually takes time to look for what he is looking for in a sequential manner. A little more sophisticated technique would be to perform a keyword search as it is currently done with search engines found in the Internet (Google, Yahoo!). It has been empirically shown [9] that locating the news story boundaries increases the speed for these tasks by several orders of magnitude. According to experimental results, a manual linear search is typically taking 80% longer than a keyword search which is itself 10000% longer than a

search based on already segmented news stories.

Brown *et al.* [139] also demonstrated the need for a proper segmentation. They compared the results of queries when using a perfectly segmented news stories and the simple use of several text-windows of fixed width (12, 24, 36, 48 lines of text) overlapping by half the window size. The average precision dropped from 0.821 to 0.538 in their news retrieval experiment. An improvement of 34.5% can be expected when trying to retrieve document with a perfect segmentation versus a fixed window of text.

When a user submits a query against a collection of multimedia documents, the user is expecting to retrieve documents which are most similar to the combination of audio-visual features and textual keywords referring to a particular topic described by the query. The goal should be to determine whether or not a document is relevant to the query. By grouping video sequences into a temporally and topically connected manner, news story segmentation helps to significantly reduce the complexity of the search space. More generally, it will help search and navigation systems to dig efficiently into large video collections.

A problem properly defined

Another interesting point is that the criteria for news story segmentation is clear: shots related to the same news story should be grouped together. Generally speaking, a story is defined as a topically contiguous section of news in a broadcast. In the TRECVid guidelines, the definition has an additional requirement which is that a story should be defined as a segment with a coherent focus which contains at least two independent, declarative clauses. However, even with very clear and detailed definitions, the task is still subjective as it depends on the meaning of the video material and can be interpreted in different ways. Even so, we have at our disposal a lot of "a priori" information that we will be able to use to reduce the complexity of the problem when choosing to consider only news video. We know "a priori" that:

- we have to look for a sequence of stories dealing with different topics ;
- we know that there is always an anchor person who is presenting the news broadcast ;
- news broadcast are edited in order to make them clearly understandable by a human viewer. So we may expect the presence of jingles, silences, logos when switching from

one story to the next.

Thanks to this "a priori" information, we will show that models can be developed to move towards a solution.

Evaluation of results

Finally, another important reason to focus on this problem is related to the evaluation of the obtained results. To evaluate our algorithms, a video collection that would be used by other research groups and a large amount of annotations were needed. Finding a large and consistent annotated video collection represents a tremendous effort for a research group. By sharing resources, TRECVid proposed all this and a task in 2004 was devoted to develop news story segmentation algorithm.

5.2 News story segmentation

In this section, we will present a review of the state of the art approaches proposed in the literature to attempt to solve the problem of news story segmentation.

5.2.1 A clustering problem

A first category of approaches focused on trying to identify coherent block or clusters. The basic assertion is that coherent topics should share similar vocabulary or features.

Text only

In the natural language processing community, Hearst [140] developed the idea of "text tiles" which are coherent regions separated by topic shifts. The topic shifts are detected by computing the similarity using the vector space model of information retrieval between adjacent block of text with a sliding window and applying a threshold. The similarity measure is a cosine distance considering two text blocks. The words are given a TF.IDF weight respectively to the considered block so that a word which is frequent in a block and rare in the remaining of the document is given a higher weight. The similarity profile is then smoothed with a median filter which possesses some robustness to outliers. Valleys of the smoothed similarity profile are detected and used as topic boundaries. The original

experiment was text-only. The semantic meaning of the similarity measure is related only to the 'locality' properties of keywords. 'Locality' refers to the fact that a set of references to a particular concept occur in proximity to one another. Locality is a good indicator of topicality. It distinguishes if a keyword appears in a document because there is really a discussion on a topic linked to the keyword or if it is only present as a passing reference. The approach shows how segmentation in different topics can be performed based on textual information. We now will see how the same idea is applied on the visual information contained in videos.

Visual only

Scene Transition Graph: By using visual information only, Yeung *et al.* [129] proposed to represent video as a graph: a Scene Transition Graph (STG) . Each node of the graph represents a shot and the edges represent the transitions between the shots based on visual similarity and a temporal locality. The segmentation is obtained by cutting the graph in subgraphs using the complete-link method of hierarchical clustering. Therefore, it performs news story segmentation by clustering similar images to reduce redundancies. This makes it easy to identify when there is a return to the same scene after a digression. The anchor person shots are recognized, because these are the most frequent cluster in a news broadcast. Hanjalic *et al.* [141] proposed in a similar way to find *logical story units* by building a graph where the linking of shots is related to an inter-shot dissimilarity measure. In the same family of approach, Veneau *et al.* [142] used a dissimilarity measure between shots and a temporal constraint to cluster shots in a hierarchy by agglomerative clustering of the video. At the beginning, every shot is considered as a cluster and is successively merged according to a dissimilarity measure based on a distance between clusters. At every location, the distance considers the distance value between clusters before and after the candidate point. The merging is done if the distance exceeds a threshold which is manually tuned for best performance.

This graphical representation of scenes as clusters of similar shots offers nice browsing properties, but one difficulty is to really reach the semantic level and to accurately detect when stories are changing because textual features are often the only important semantic cues. Shot similarity for this family of approaches is based on matching block between keyframes [141], comparing color histograms between keyframes [143], mean color histogram [104], comparing color histograms between any frame in two shots [144]. All these approaches rely mainly on similarities between keyframes and this visual comparison

is far to be significant in most cases to reach the semantic level.

5.2.2 A multimodal classification problem

A second category of approaches considers probabilistic models to solve the semantic segmentation problem. The main idea is to formulate a framework to properly perform multimodal fusion and to improve the decision process. From a probabilistic point of view, news story segmentation is about the estimation of the probability $p(b|x)$ measuring the likelihood that a story boundary b is present at a given position in the video stream knowing a vector x which contains information about various multimodal sources: visual, audio and text information coming from the video and audio streams of MPEG files and Automatic Speech Recognition (ASR) transcripts. News story segmentation can be viewed as a binary classification problem $\{0\ ,\ 1\ \}$ to decide whether a given location is a story boundary or not.

Maximum entropy classifier

Beeferman *et al.* [145] introduced a statistical approach based on the maximum entropy model to fuse binary cues. The partitioning of news stories was based on text-only features of short or long ranges. A set of lexical hints also are learned to associate with the presence of story boundaries using annotated data. Hsu *et al.* [132] used the same maximum entropy model as Beeferman *et al.* [145] but extended it to cover audio and video information. A feature wrapper has been designed to convert audio and video cues into binary cues. The aim is to develop a robust statistical framework so that the fusion of diverse modalities can be achieved. In addition the framework is able to adapt to different news video sources.

First, a set of binary features $f_i(x, b)$ is defined concerning heterogeneous information about face, motion, significant pause, commercial detector, text segmentation score. The fusion of these features is performed by using a maximum entropy classifier. The interest of using a maximum entropy classifier is that features can be mutually dependent. It works then better than a Bayesian classifier for such kind of problems. The probability of presence of a news story boundary is defined in a straightforward manner as an exponential model:

$$p_\lambda(b|x) = \frac{1}{Z_\lambda(x)} e^{\sum_i \lambda_i f_i(x,b)} \tag{5.1}$$

where $Z_\lambda(x)$ is the partition function, λ is a vector containing the weights to give to all different features $f_i(x, b)$. The approach is based on the statistical framework of feature selection for random fields and exponential models [146, 147]. The model is then defined and the problem is now to estimate the optimal weight vector λ by supervised learning. This is done by minimizing the KL divergence defined as:

$$\text{argmin}_\lambda D(\widetilde{p}||p_\lambda) = \sum_x \sum_b \widetilde{p}(b, x) \log \frac{\widetilde{p}(b|x)}{p_\lambda(b|x)} \tag{5.2}$$

where $\widetilde{p}(b, x)$ is the empirical joint probability estimated over the training instances. The algorithm used to solve this functional is the iterative scaling algorithm. Feature selection is possible by scanning all features and adding them one by one by selecting the features offering the largest gain in log-likelihood when added to the current set of features incorporated to the model. The log-likelihood is given by:

$$L_{\widetilde{p}}(p) = \sum_x \sum_b \widetilde{p}(b, x) \log p_\lambda(b|x) \tag{5.3}$$

This approach has been among the first to propose a sound way to fuse such heterogeneous and multimodal features and to obtain improved performances over monomodal strategies.

Boosting and Support vector machine

Hsu *et al.* [130] demonstrates with a number of experiments that news story segmentation can be achieved with various techniques for multimodal classification: maximum entropy, boosting and support vector machines.

Boosting techniques have proven to be an effective method to improve the performance of weak classifiers. At the origin, PAC learning [148] is the inspiration of Boosting. Probably Approximately Correct (PAC) learning formally proved that by appropriately combining learners which exhibit a performance slightly better than random guessing, the final classifier can perform arbitrarily well. The first boosting algorithm with a polynomial complexity was derived by Shapire *et al.* in [149]. At each iteration, Boosting trains what is called a weak learner which is usually a decision stump, a perceptron or a decision tree which gives the lowest classification error for the given problem. The weak learner is called "weak", because it is not expected from it a good classification: the best decision stump

for a complex problem may have an accuracy of 51%. Despite this, performances are obtained by calling such weak learners to solve a sequence of learning problems. The succession of learning problems is obtained by re weighting the instances so that an emphasis is put on the ones which have been incorrectly classified previously. The final classifier is a weighted combination of weak learners. In [150], it was shown that boosting can be considered as a stage-wise gradient descent procedure that minimizes an exponential cost function. In practical applications, boosting has been impressively used for the problem of face detection [18].

Support vector machine [151] is another powerful method for discriminative learning. The decision margin is found so that the structural risk is minimized. In Hsu *et al.* experiment, support vector machines show superior performances. However, the features and classification are local and no context is taken into account. The KDDI group [86] uses SVM classifiers with low-level generic features. Performances are enhanced by learning several classifiers dedicated to specialized sections of a news report such as "top stories" and "headline sports" which exhibit different characteristics when compared with the general content.

5.3 Contextual News Story Segmentation

5.3.1 A contextual problem

Why taking context into account ?

Why are humans so superior to computers when trying to interpret semantic ? One reason is attributed to the aptitude to combine recognition of semantic and knowledge about the association of semantic concepts with other observations. For humans, the knowledge of context helps to resolve the problem of identifying a semantic concept even when the observations are ambiguous.

The figure 5.2 illustrates how contextual information helps a human being to recognize an object even in an ambiguous situation. The blurring of the pictures makes it hard to distinguish the hairdryer or the drill. However, by looking around, we recognize easily the objects thanks to the bathroom and garage contexts which remove the ambiguity.

In the field of computer vision, a large number of researchers have acknowledged the importance of the use of contextual information to achieve the goal of scene understanding

Figure 5.2: Importance of context for the problem of object recognition. A hairdryer can be extremely similar to a hole driller from the visual point of view.

[152–154]. Such scene understanding systems attempted to recognize significant objects in the scene and to identify the relevant object relationships. A semantic concept such as a news story transition appears in a given context and we propose to use that knowledge to improve on existing techniques.

We distinguish three different approaches for such classification problems as shown in Figure 5.3. The classical approach consists in using visual features X_v to infer a visual class Y_v and independently using audio features X_a to infer an audio class Y_a and so on. Such approaches are still commonly used by research groups that belong to a community focusing on one particular modality.

The multimodal approach is about using a combination of features $X = (X_v, X_a, X_t)$ coming from every modality to infer visual, audio and text labels. As an example, the word "temperature" which a textual feature is likely to be useful to infer the visual label "weather news". Multimodal classification is generally done using SVMs or Boosting. In [130], several multimodal classification methods are compared for the purpose of news story segmentation.

The contextual and multimodal approach goes further by not only considering the link between feature vectors and labels but also the relationships between labels associated with different modalities: in our case, a context is thus defined as the compatibility for a label to appear together with labels related to other modalities.

First it should be noted that in order to provide a context to our algorithm, we will need to define a set of semantic concepts. The extracted multimodal features will be our observations to infer these concepts. The context will then be represented as the set of

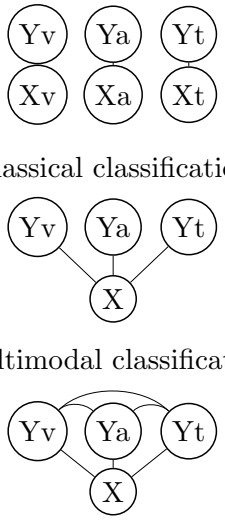


Figure 5.3: Various classification models where $X = (Xv, Xa, Xt)$.

relationships connecting the different semantic concepts. A context can be formulated as a Bayesian network. The co-occurence of semantic concepts is an important added-value of information. Some semantic concept usually appear together: for example a visual label "anchor person" is most likely to appear with the audio label "speech". Some semantic concept are incompatible: it is unlikely that the text label "Weather news" can actually appear together with the visual label "Sport scene". All this contextual information should be used to improve on existing algorithms. This should be done by explicitly capturing all the interactions between labels.

In terms of performance, there are two main motivations for using contextual relationships. First by considering a classification task to infer a semantic label, the recall of the classifier is improved, because even if low-level features happen to be ambiguous the presence of context will support inference. Secondly, the precision of a classifier is improved by reducing the set of possible labels knowing the context. By imposing prior knowledge, context enhances the detection of semantic labels. An extended use of the context is key to enable an accurate and complete video modeling.

Statistical sequence modeling

Few approaches have used the context to improve results of news story segmentation algorithms. We can refer to the work of Chaisorn *et al.* [8] who introduced the idea of a

two-level multimodal system. The semantic labeling of video shots classification is first performed by using decision trees to infer semantic concepts out of low-level features. Specifically the concepts used by Chaisorn are: Intro/Highlight, Anchor, 2Anchor, Meeting/Gathering, Speech/Interview, Live-reporting, Stillimage, Sports, Text-scene, Special, Finance, Weather, and Commercial. The contextual relationships between these labels are then learned with a Hidden Markov Model (HMM) for the segmentation of news stories. The contextual relationships captured by the algorithm are then only temporal. The multimodality of features is used only for the classification into semantic concepts. In addition the inference of the concepts and of the context is done in two non-interacting successive steps.

Network of semantic concepts

In his pioneering work, Naphade [155] proposed to model such contextual relationship using what he called a multinet. A multinet is a probabilistic graphical network which encodes contextual information.

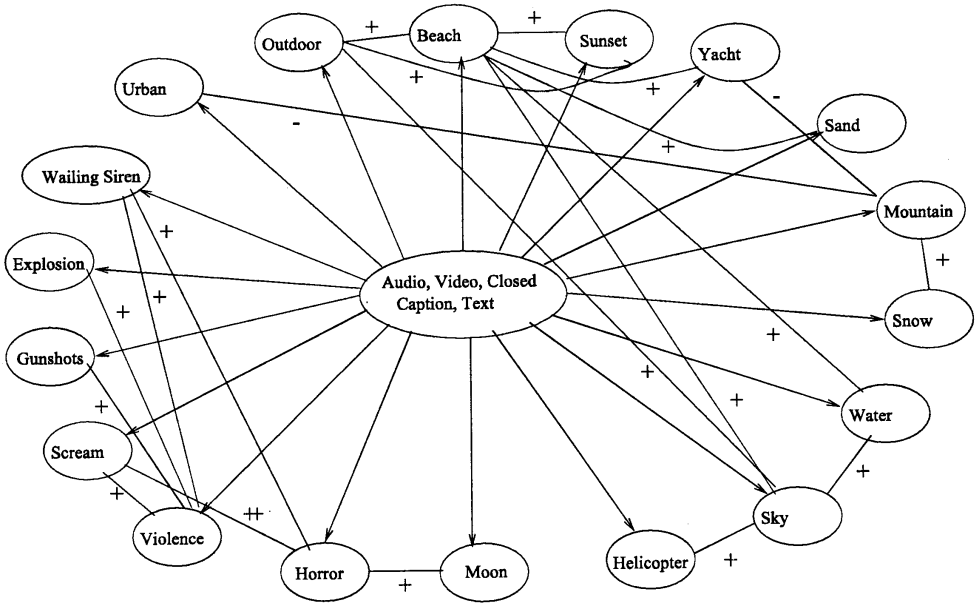


Figure 5.4: A conceptual figure of a multinet. Signs indicate positive or negative interaction between concepts.

In Figure 5.4 from Naphade [155], it is possible to see how labels interact with

each other and how contextual information may support inference. From the algorithmic point of view, multinets are learned by Bayesian estimation with the maximum likelihood formulation of the Expectation-Maximization (EM) algorithm. 20% of improvement in classification performance has been achieved by taking context into account. These results were found encouraging to focus our research in this direction.

We will present now our contribution to semantic video structuring: a contextual model for the task of news story segmentation. We will start by describing the different steps of feature and label extraction. Then we will propose a contextual model and validate the idea using an information gain measure. Finally we will enter into the details of the learning algorithm. The model will have to be trained in a supervised manner and requires training data. Figure 5.5 gives a general view of how we generated training instances to learn our model.

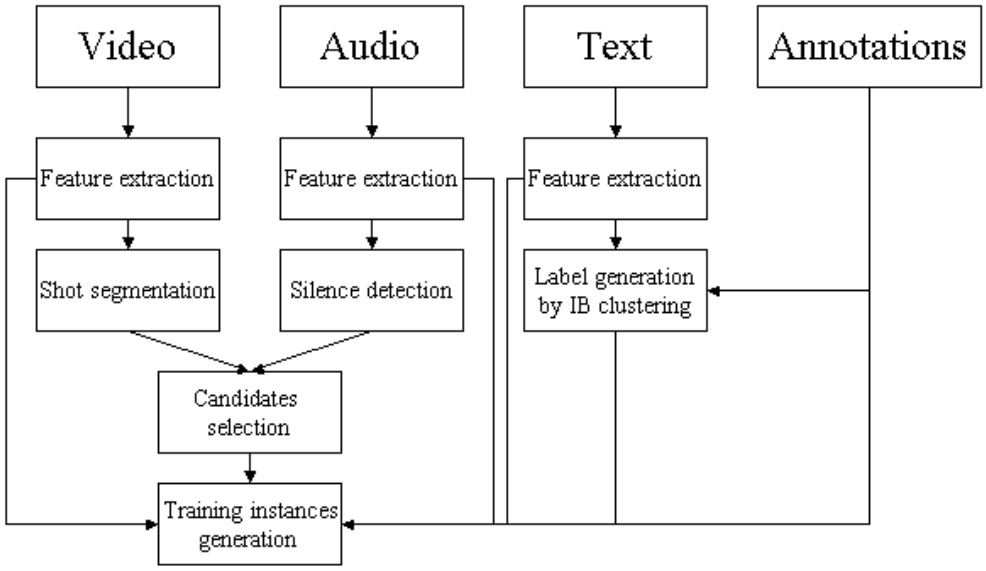


Figure 5.5: Training data generation for news story segmentation.

5.3.2 Feature and label extraction

Design of the feature pool

An expressive feature pool is necessary so that our model might find relationships between features and our set of semantic labels. The goal here is to feed the system with a representation of features describing the video content for semantic labeling as well as news story boundary detection. Another restriction is that the set of features has to be generic enough to adapt to different datasets like CNN or ABC news broadcasts.

Candidate points

We will not assign semantic labels to all shot units, but we will restrict the search for news story boundaries to a reduced set of candidate points and only consider the labels of the shots which are located around them. We will define in this section how we selected the candidate points.

In the section 4.2, we proposed a generic algorithm to detect all types of transitions between different shots. By using shot boundaries alone as candidate points, the set covers 91% of the news story boundaries.

A simple and useful cue to detect a news story boundary is the presence of a short silence during the transition from one shot to another, as the anchor persons usually mark a short pause (around 200 ms) when switching from one story to the next. By combining with silence detection, the set of candidate points is reduced by 33%, but still covers 87% of the news story boundary. This is advantageous and hence the candidate points will be the union of the set of shot boundaries and the set of audio silences with a tolerance window of one second.

Visual information: Following the discussion of section 2.1, the color information will be represented by a color histogram of the keyframe for a given video segment. This is needed to characterize particular backgrounds in frequently occurring video segments. We compute a 128-bin histogram in the YUV color space directly taken from the MPEG data. The dimensionality of the color vector is 256, because we consider two keyframes before and after a candidate point. In the figure 5.6, we show a set keyframes coming from randomly selected anchor person shots. It shows that using color is relevant for the labelling into “anchor person”, because the studio setup is constant among various

news broadcasts with the blue color of the screens in the background. The same remarks can be made for other semantic labels such as "weather" or "sport" since golf, hockey and basketball always appears with similar color setup in the background (green for golf, white for hockey and light brown for basketball).

The motion information is represented as a 64-bin histogram of the amplitude and angles of the horizontal and vertical components of the optical flow directly extracted from the MPEG stream as well. A measure of the motion intensity is added to distinguish between small, medium and high level of actions. Motion is particularly important to discriminate sports scene/ads from news subject monologue/weather news as an example. Again, the dimensionality of the motion vector is doubled at 128, because we consider motion during the shots before and after a candidate point.

Figure 5.6: Set of randomly selected anchor person shots.

Audio information: The features used to represent the audio information in the time as well as in the spectral domain is obtained by considering one-second audio clips. We will use a perceptual and expressive set of features previously described in the section

2.2: RMS volume, VSTD, ZCR, HZCRR, SC, BW, SF, CF, FCVC4, SR. Considering 5 seconds before and after the candidate point, we take the min, max, mean and standard deviation of the feature's values in order to reduce the dimensionality and obtain a vector of dimensionality 44. The redundancy of the audio information is allowed by the still low dimensionality of the feature vector. We will show that the most useful features will be selected later by the machine learning algorithm.

Text information: The text information is provided in the TRECVID evaluation data in the form of time-stamped ASR transcripts. The ASR engine has been developed by the LIMSI [12]. The text content is modeled by the word vector representation after stemming and stop-word removal as presented in the section 2.3.3. We use as a feature the occurence of bigrams which are correlated to the beginning or end of a news stories. The bigrams are selected according to two criteria: they appear often together during a transition between stories and they appear very rarely together in the whole collection. The top combinations respecting these two criterias are then considered as good markers for detecting transitions between different news.

News story boundary specific features: To improve the results, more news-story specific features has been added. We do not take any other assumptions than the following one: our video collection is a set of broadcast news. Therefore we are not interested by specific jingle or logo detection, but we remain as generic as possible in the choice of our features. A significant pause is characterized by the low RMS value as well as a high ZCR and also by the length of the silence. We scan the audio data in a window of 5 seconds around the candidate point and returns the characteristics of the most significant pause found.

Semantic concepts to use as context

In the context of the TRECVID experiment, participants did create and share annotation of the training data in a collaborative way [156]. We have at our disposal annotations for the following semantic classes in the training set:

- visual content: news subject monologue, studio-settings, outdoors, man-made scene, cartoon, weather news, sport scene, text scene, graphics, ads ;
- audio content: speech, music, noise, speech+music, speech+noise, other sound.

For the text content, we do not have any set of predefined and annotated labels. We will therefore use an unsupervised clustering algorithm to define a set of N classes from the training data. We have chosen to use the Information Bottleneck algorithm presented in the section 2.3.3. The implementation used is the sequential optimization motivated by the Information Bottleneck method of Noam Slonim [157]. In fact, this unsupervised document clustering algorithm has shown to be as powerful as a supervised one to classify messages into the corresponding newsgroup categories. This conforted us to use this algorithm to cluster our stories into a set of labels relevant to topicality. In the experiment, we reduce the set of textual labels to 20 and we set parameter β to the value 50.

	Most frequent words
Cluster 1	rain weather with storm continue today across temperature forecast new california coast
Cluster 2	med won gold olymp when team with game five two today hi second sport
Cluster 3	presid clinton today south with say hi africa nate first visit lead talk
Cluster 4	day with look go what plain like all wait make get again off well first win season
Cluster 5	presid with house clinton white about lawyer investigation jury grand said case sexual
Cluster 6	point nasdaq dow wall street gain back market stock industrial today jones eighty seven close
...	...

Table 5.2: News story clustering into a set of 20 topics.

The table 5.2 shows a short summary of the content of the different news story clusters found thanks to the information bottleneck algorithm. We will use the cluster numbers as labels for the textual modality. This is a way to obtain meaningful labels and to incorporate contextual information about the topics even if we were not given any topical labels in the annotations of TRECVid. The number of 20 textual labels has been chosen, because it corresponds roughly to the number of topics covered by news broadcasts (advertisements, weather, world news, US politics, judiciary, financial, sport, health, ...) each of them with sub-thematics. By increasing the number of clusters, we offer a better description but which is less discriminative. At the opposite, reducing the number of clusters has an effect on the descriptive power of textual information. The trade-off has been experimentally found and attempt to match human understanding of news clusters.

From now on, training instances have been generated. We will now describe how we propose to model the context of a news story.

5.3.3 Contextual modeling

Description of the contextual model

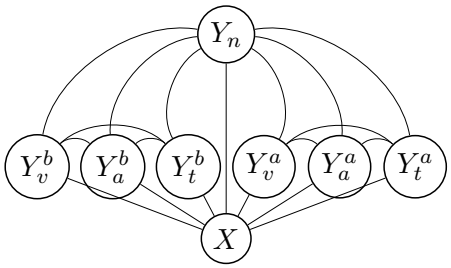


Figure 5.7: Contextual news story model. The subscripts n,v,a,t denote "news boundary", "visual", "audio", "text" labels. The superscripts b and a denote video segments located "before" and "after" the candidate points.

In [158], we proposed a contextual model as shown in the Figure 5.7. We wish to jointly classify the set of candidate points with the news boundary/non-news boundary labels as well as the video segments located before and after the considered candidate point into a set of semantic labels related to the three modalities. For the video segment before/after the candidate points, the classification should take into account the interaction between labels to improve consistency. For example, it will reduce the possibilities to see the visual modality being assigned the label 'sports' if audio and text are both pointing toward 'financial news'. Moreover all labels related to all 3 modalities before and after the candidate points will interact with the news story segmentation labels Y_n in this model. Finally and of course all labels are interacting with our observable and low-level features set X.

At this stage, we face a very complex model taking into account:

- the multimodality of features ;
- the relationships between semantic concepts concerning different modalities ;
- temporal modeling.

Validation of the idea

Here, we will discuss how much improvement in news story segmentation can be expected by adding the contextual information considering that the labels are given. Information gain is a measure used to evaluate the worthiness of an attribute to infer a label. It is given by:

$$\text{InfoGain}(\text{Label}, \text{Attribute}) = H(\text{Label}) - H(\text{Label}|\text{Attribute}) \tag{5.4}$$

where H is the entropy.

N°	Labels	Information Gain
1	face after	0.008638
2	graphics and text after	0.0086
3	graphics and text before	0.008296
4	news subject monologue after	0.006669
5	non studio settings after	0.005548
6	non studio settings before	0.003592
8	ads before	0.003543
9	music after	0.000927
10	text cluster 29	0.00083
11	text cluster 24	0.000764
12	text cluster 14	0.000721

Table 5.3: Ranking of the most informative labels related to news story segmentation.

The table 5.3 shows the most informative labels to infer a news story boundary ranked by information gain. It validates the idea that semantic labels can actually be informative to improve classification accuracy.

The table 5.4 shows the ranking of the top labels that help to infer the audio label "speech after".

All this demonstrates that the semantic labels before and after our candidate points provide an informative context. The inference problem to solve is dual. We need a context to improve inference and inference is needed to determine a context. Only a joint estimation of both the labels using the features and the context is optimal in terms of the use of the information at disposal at every learning step. All these results explain why it is important to perform a joint estimation not to waste informative resources.

N°	Labels informative for "speech"	Information Gain
1	speech before	0.21630
2	ads after	0.02942
3	face	0.00786
4	music after	0.0062
5	news subject monologue after	0.00499
6	music before	0.00344
8	news subject monologue before	0.00327
9	graphics and text after	0.003221
10	text cluster 16	0.003191

Table 5.4: Ranking of the most informative labels related to the "speech after" audio label

5.3.4 Contextual learning algorithm

We propose to build a contextual model designed for news story segmentation and to use Boosted Random Fields, as presented by A. Torralba *et al.* in [19], to estimate the parameters of the model. We will focus this discussion on the advantage of using Boosted random fields. It shares the same ability as boosting to dig deeply into the set of features to select the ones which are really helpful to improve the classification results with minimal over-fitting. The correlation or the redundancy between the different features does not really matter, because the learning process will perform feature selection by design. It is in addition a promising new approach to model relationships between labels so that higher consistency of the classification results can be achieved.

Generative modeling versus discriminative modeling

In the following, X is a random variable over the low-level features and Y_i are random variables over corresponding visual, audio, textual and news boundary labels with respect to different finite labels alphabet.

Naphade [155] proposed to model contextual relationships using what he called a multinet. A multinet is a probabilistic graphical network which encodes contextual information. Such Bayesian models are generative models, assigning a joint probability to paired observations and labels $p(X, Y)$ which involves implicit modeling of the observations via $p(Y|X)$. On the contrary, in a discriminative framework, one directly models the distribution $p(Y|X)$. In [159], it is noted that a potential advantage of using this when the true underlying generative model is quite complex even though the class posterior

is simple. In that case, a generative approach spends a lot of resources on modeling a joint probability which is not particularly relevant to the task of inferring the class labels. A more complete comparison between discriminative and generative models is provided in [160] and [161].

For generative models, the criterion to estimate model parameters is typically to maximize the joint likelihood of training examples. It is not practical to represent multiple interacting features of the observations, since the inference problem for such models is intractable when the graphical model is not a chain or a tree.

These problems are the main motivation that led us to consider conditional models as an alternative. A conditional model does not focus its effort on the observations, but it specifies the probabilities of possible labels given an observation sequence. For each node i, N_i corresponds to the neighbors of the node i. The distribution of the labels conditioned on X is the same as for Conditional Random Fields [162] and is given by:

$$p(Y|X) = \frac{1}{Z} \prod_i \left(\phi_i(Y_i, X) \prod_{j \in N_i} \psi_{i,j}(Y_i, Y_j, X) \right)$$

where the functions ϕ_i correspond to evidence associating locally observed data and labels. The $\psi_{i,j}$ are compatibility potentials depending on observed data and also on neighbor labels, allowing label interaction. Z is the partition function. It is possible to note at this stage that the model is discriminative: the focus is on the conditional probability $p(Y|X)$ and we will never consider the joint probability $p(Y, X)$ as it is done when using generative models such as HMMs. The Conditional Random Fields have obtained promising results in a number of domain where there is interaction between labels such as tagging, parsing and information extraction in natural language processing [163], [164], [165].

We still provide a temporal modeling of the video data by considering shots located before and after the candidate points. A discriminative model also requires less strongly labeled data for training, because at the contrarily of a generative model we will never attempt to estimate the joint probability $p(X, Y)$. This is advantageous, because the joint probability requires a lot of training data to be properly estimated.

Model parameters estimation via Boosted Random Fields

With such a model, the problem of learning the ϕ_i, even if the functions $\psi_{i,j}$ are known, is intractable when the graph structure is not a chain or a tree because of the computation of Z (an exponential sum). Boosted Random Fields provide the theoretical tools and optimization framework to estimate the parameters of such a complex model in a consistent way. The problem is solved by constructing an iterative approximation of the solution by fitting an additive model for the local evidence and label compatibility functions.

About boosting: In this paragraph, only binary classes will be considered. The training data contains M instances of training pairs ($x_{im}, y_{im} \in \{-1, 1\}$). The pseudo-code for the popular Adaboost.M1 is the following where N is the number of boosting iterations:

1. Initialize the observation weights $w_i = \frac{1}{M}$, i=1, ..., M
2. For $t = 1$ to N
   (a) Fit a weak learner $B_t(x)$ to the training data using weights w_i
   (b) Compute the weighted error rate on the training samples
   $$\text{err}_t = \frac{\sum_{i=1}^{M} w_i I(y_i \neq B_n(x_i))}{\sum_{i=1}^{M} w_i} \tag{5.5}$$
   (c) Compute $\alpha_t = log(\frac{1-\text{err}_t}{\text{err}_t})$
   (d) Set $w_i = w_i \exp(\alpha_n I(y_i \neq B_t(x_i))$, i=1, ..., M
3. Output $B(x) = sign(\sum_{t=1}^{N} \alpha_t B_t(x))$

Boosting is a way of fitting an additive model. The general problem of additive modeling can be formulated in the following way:

$$\min_{\{\alpha_t, \gamma_t\}_1^N} \sum_{i=1}^{M} L(y_i, \sum_{t=1}^{N} \alpha_t b(x_i; \gamma_t)) \tag{5.6}$$

The aim is to minimize a loss function L over the training data related to the classification error by finding the optimal coefficients α_t and parameters γ_t of every basis functions $b(x_i; \gamma_t)$. Many loss and basis functions can be proposed. Popular loss functions are

squared-error or likelihood-based loss function. Additive models are at the basis of many algorithms such as decision trees or hidden layer neural networks.

For boosting, the final classifier that produces inference can also be expressed as a weighted sum of "weak" learners B_t:

$$B(x) = \sum_{t=1}^{N} \alpha_t B_t(x) \tag{5.7}$$

where the α_t and $B_t(x)$ are computed during the learning process. There is a clear analogy, between $B(x)$ and the term $\sum_{t=1}^{N} \alpha_t b(x_i; \gamma_t)$ of equation (5.6). The choice of the weak learner defines the basis function b, the parameters of the weak learner defines γ_t.

However, the computational cost of solving equation (5.6) is too high when choosing many loss functions and/or basis functions. When it is feasible to find an efficient way to solve the subproblem of fitting only one basis function (see equation 5.8) efficiently, forward stage wise additive modeling makes it possible to sequentially add new basis functions without having to adjust the parameters and coefficients of the already added basis functions.

$$\min_{\{\alpha,\gamma\}} \sum_{i=1}^{M} L(y_i, \alpha b(x_i; \gamma)) \tag{5.8}$$

Forward stage wise additive modeling works in the following way. At the initialization, the current model is $f_0(x) = 0$. At each iteration, the basis function (or weak learner) and coefficient is estimated to solve the equation (5.8). The basis function is then added to the current expansion $f_{n-1}(x) = \sum_{t=1}^{n-1} \alpha_t b(x_i; \gamma_t)$ to give $f_n(x)$ and the algorithm go to the next iteration. The process becomes iterative and computationally efficient.

It has been shown by Friedman *et al.* [150] that Adaboost.M1 is equivalent to a forward stage wise additive modeling using the exponential loss function:

$$L(y_i, B(x_i)) = \exp(-y_i B(x_i)) \tag{5.9}$$

The exponential loss function can be seen as a monotone continuous approximation of the misclassification loss. It penalizes in a continuous manner the negative margin values more than it rewards the positive ones. Popular weak learners for Boosting are decision stumps or trees, because they have low computational complexity classifiers and can be combined to produce strong classifiers.

Handling multi-class problems: The generalization of Boosting to the multiclass case is straightforward. We describe the AdaBoost.MH algorithm defined by Shapire and Singer, because it dominated other approaches in the study [166]. For a number of classes L, Adaboost.MH proceeds in the following way:

1. Expand the M observations into $M \times L$ pairs

$$((x_{im}, 1), y_{im1}), (x_{im}, 2), y_{im2}), ..., (x_{im}, L), y_{imL})) \tag{5.10}$$

   where $m = 1, ..., M$ and $y_{iml} \in \{-1, 1\}$ response for node i, observation m and class l.

2. Apply Boosting to the augmented dataset, which leads to:

$$F(x, l) = \sum_t f_t(x, l) \tag{5.11}$$

3. The result of the classification is:

$$\text{argmax}_l F(x, l) \tag{5.12}$$

In the rest of the discussion, we will only consider binary classification, but it should be kept in mind that the number of training instances has "exploded" by the augmentation of the dataset required to generalize to the multiclass problem.

Boosted Random Fields: Coming back to our problem, we need to estimate from training data the functions ϕ_i and $\psi_{i,j}$ of the model shown in figure 5.7 and defined by the following equation:

$$p(Y|X) = \frac{1}{Z} \prod_i \left(\phi_i(Y_i, X) \prod_{j \in N_i} \psi_{i,j}(Y_i, Y_j, X) \right)$$

Considering all training instances m, the cost function to minimize at iteration t is the classification error J^t defined by:

$$J^t = -\prod_m \prod_i p(y_{im} = +1|x_{im}, t)^{\frac{y_{im}+1}{2}} p(y_{im} = -1|x_{im}, t)^{1-\frac{y_{im}+1}{2}}$$

where the belief $p(y_i|x_i,t)$ of a node i at iteration t is proportional to the local evidence function multiplied by the product of all messages coming into the node i from all of its neighbors.

$$p(y_i|x_i,t) \propto \phi_i^t(y_i) Mess_i^t(y_i)$$

where $Mess_i^t$ is the product of all messages coming from neighbors, which are function of the beliefs and of the compatibility functions ψ:

$$Mess_i^{t+1}(y_i) = \prod_{k \in N_i} \mu_{k \to i}^{t+1}(y_i) \tag{5.13}$$

$$\mu_{k \to i}^{t+1}(y_i) = \sum_{y_k \in \{-1,+1\}} \psi_{k,i}(y_k, y_i) \frac{p(y_k|x_k,t)}{\mu_{i \to k}^t(y_k)} \tag{5.14}$$

If we denote:

$$F_i^t = \frac{log(\phi_i^t)}{y_i} \tag{5.15}$$

$$G_i^t = \log M_i^t(+1) - \log M_i^t(-1) \tag{5.16}$$

$$p(y_i|x_i,t) = \frac{1}{1 + e^{-(F_i^t + G_i^t)}} \tag{5.17}$$

F_i^t and G_i^t are direct functions of the local evidence and of the compatibility potentials respectively. It can be shown that the cost function simplifies to:

$$\log J_i^t = \sum_m \log(1 + e^{-Y_i(F_i^t + G_i^t))}) \tag{5.18}$$

The main idea is not to directly estimate the functions ϕ_i and $\psi_{i,j}$, but to iteratively minimize the cost function via two successive stages of Boosting by using an additive model for F_i^t and G_i^t.

$$F_{i,m}^t = \sum_{n=1}^{t} f_i^n(x_{i,m}) \tag{5.19}$$

$$G_{i,m}^t = \sum_{n=1}^{t} g_i^n(b_m^t) \tag{5.20}$$

The functions F and G will be learnt by forward stage-wise additive modeling. As in logitBoost, we achieve this by optimizing the second order Taylor expansion of equation (5.18) with respect to f and g:

$$\text{argmin}_{f_{i,m}^t} \log J_i^t \approx \sum_m w_{i,m}^t (Y_{i,m}^t - f_{i,m}^t(x_{i,m}))^2 \quad (5.21)$$

$$\text{argmin}_{g_{i,m}^t} \log J_i^t \approx \sum_m w_{i,m}^t (Y_{i,m}^t - g_{i,m}^t(b_m^t))^2 \quad (5.22)$$

where the weights are given by $w_{i,m}^t = p(1|x_{i,m}, t)p(-1|x_{i,m}, t)$. The basis functions f_i^n and g_i^n are weak learners in the form of regression stumps. The functions f_i^n are dependent on the features $x_{i,m}$ of the training data whereas the functions g_i^n are dependent on the beliefs b_m^t. Iteratively, the weak learners f_i^n and g_i^n are chosen so that the cost function is minimized; this is done by weighted least square.

Discussion on the algorithm

The training algorithm is basically the following for a given number of iterations:

- search for the optimal weak learner on the features to calculate f ;
- search for the optimal weak learner on the beliefs to calculate g ;
- update according to the equations F (5.19), G (5.20);
- update $p(y_i|x_i, t)$ (5.17) and $w = p(1|x_i, t)p(-1|x_i, t)$.

The two stages of boosting are combined to jointly converge to a solution that takes into account local potentials F and compatibilities between labels G. As in Boosting, the re-weighting by w of the training instances pushes the algorithm to focus on hard examples. When estimating f and g, the feature and label selection is done by the weak learners. All features and labels are considered, but only the ones which are useful to reduce the classification errors are selected.

It is similar to a two-stage learning algorithm where the labels learned at the first stage are then used as features for the second stage commonly used to deal with the semantic gap (see [8] for a combination of decision trees and HMM) with the major difference that all labels are learned jointly and interact together during the learning process.

The time complexity of the algorithm is a linear function of the number of iterations T, the number of nodes N and the number of features V and is written $O(T.N.V)$.

5.4 Toy example

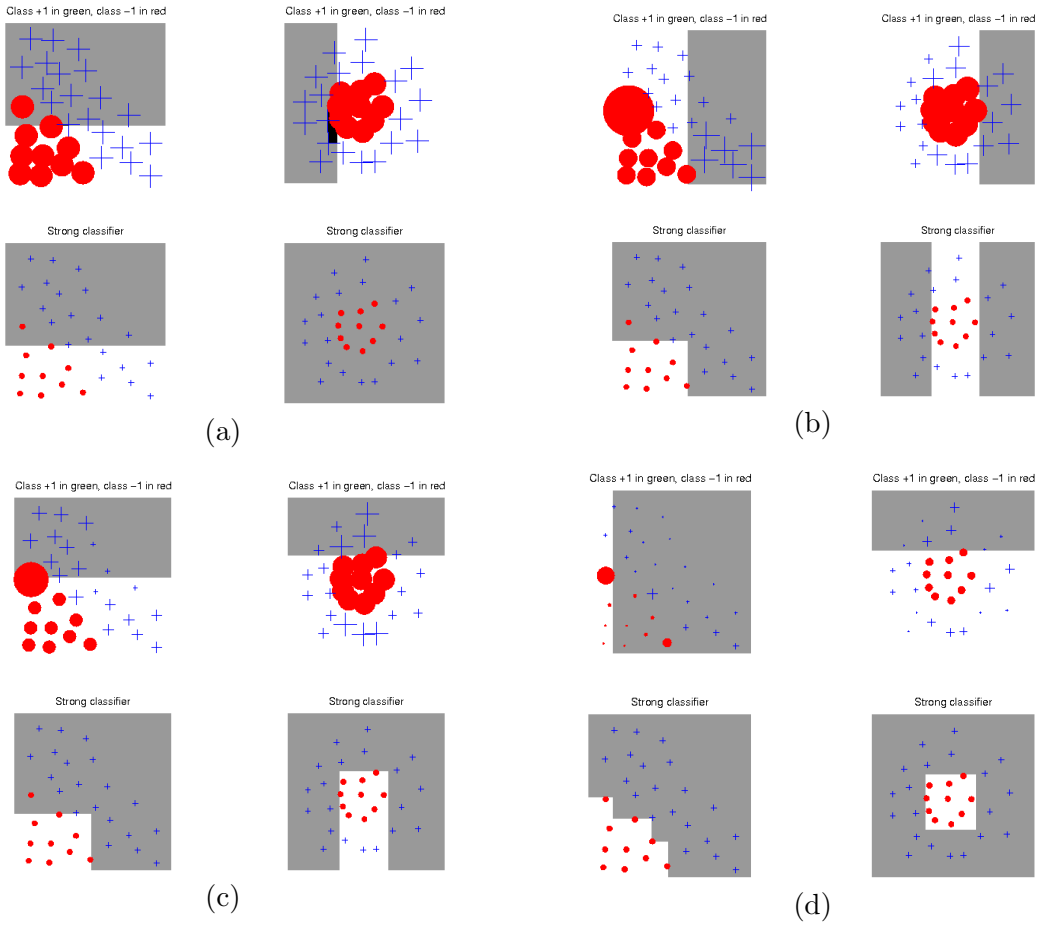


Figure 5.8: Evolution of the boosting algorithm for clean data for both problems at (a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 10

In this section, we will propose to devise a toy example in order to present in a controlled setup the behavior of the contextual learning approach and further validate the idea.

We propose two classification problems which consist of classifying points according to their two dimensional coordinates in two classes $\{-1, +1\}$. In figure 5.8, we show how

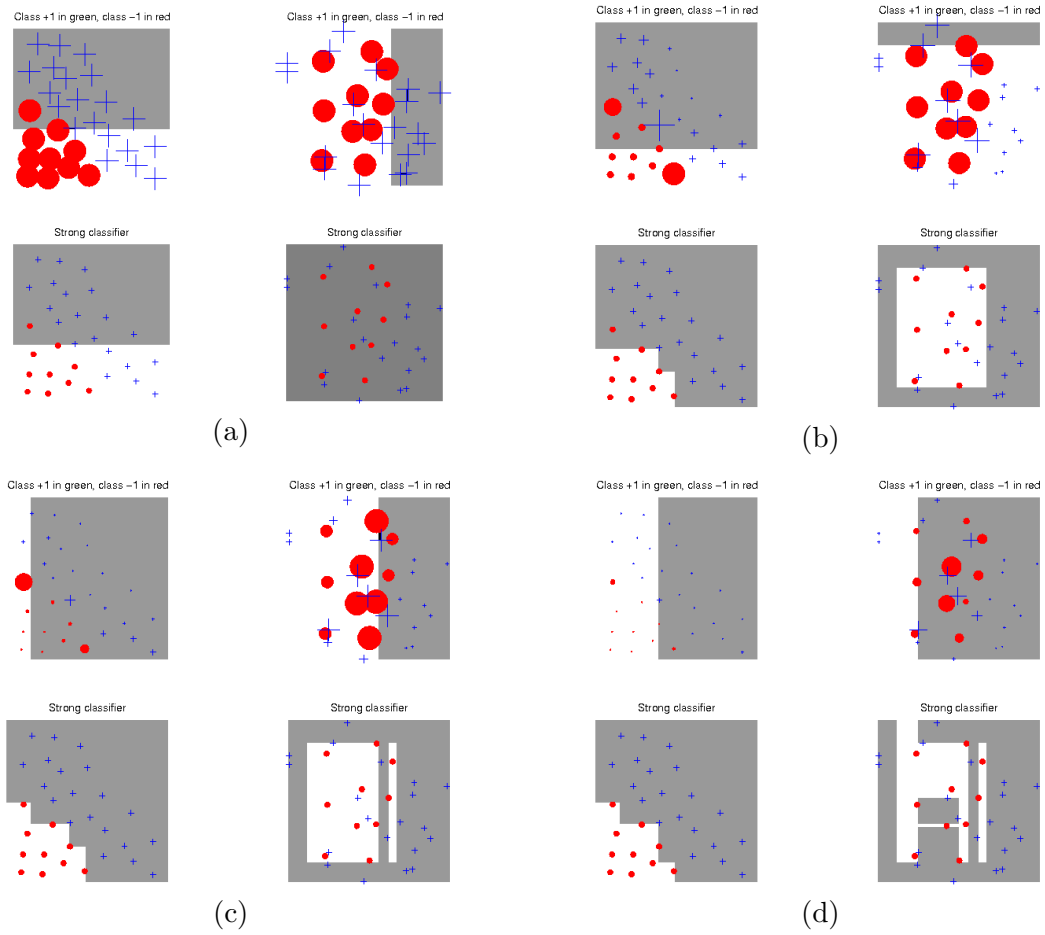
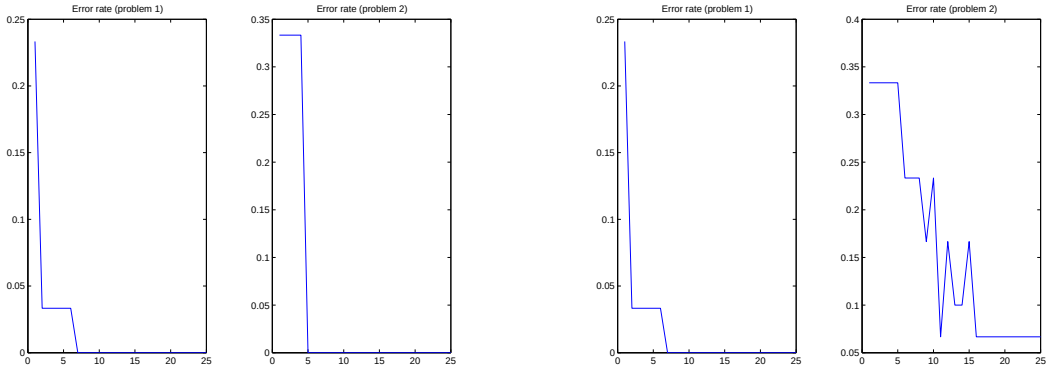


Figure 5.9: Evolution of the boosting algorithm when the second problem suffers from noisy features at (a) Iteration 1 (b) Iteration 5 (c) Iteration 10 (d) Iteration 25

the boosting algorithm performs without using contextual information. The blue crosses and the red circles corresponds respectively to the class $\{-1\}$ and $\{+1\}$. The size of the crosses or circles correspond to the weight of the training instance. The black and white area correspond to the decision plane for the weak learner and the strong classifier. We see that when the different classes are easily separable, the classifier are learnt properly in just a few iterations.

If we add noise for one of the problem as shown in figure 5.9, we observe that the boosting algorithm attempts to minimize the classification error on the set of examples provided, but it is clear that the resulting classifier has a much lower quality and tends to overfit. It is a lot harder to find proper separation boundaries simply because no clear



(a) (b)

Figure 5.10: Error rates in function of the number of iterations for the boosting algorithm for the problem 1 with clean features and the problem 2 (a) with clean features (b) with noisy features

structure are to be found in the data. This is a situation which will occur when dealing with semantic video classification problems.

The figure 5.10 shows how the classification error changes with the number of iterations for the clean and the noisy case.

Now we are ready to do the same experiments again, but this time the contextual approach will be used in order to study its benefits. The first thing to do is to create artificially some correlation between the two different classification tasks. We will then study how the results are changing when the correlation parameter changes. It will demonstrate qualitatively what is brought when contextual relationship are taken into account during the learning process. The figure 5.11 shows the evolution of the classifier related to the features. An important remark is that the classifier related to the compatibilities between label is not shown in the figures, because the function G which is related with the compatibilities is a variable of the beliefs and not of the 2-D feature coordinates. Anyway, it does actually play its role during the learning process. The strong classifiers that we see evolve in this figure are only related to the function F of the BRF algorithm. We see that the strong learner related to the function F is not able to separate quickly the two different classes.

The figure 5.12 shows how the classification error changes with the number of iterations for the clean and the noisy case when different correlations exist between the two set of labels. We see that the information coming from the compatibilities between labels helps the learning for the noisy classification problem. We observe that a fewer number

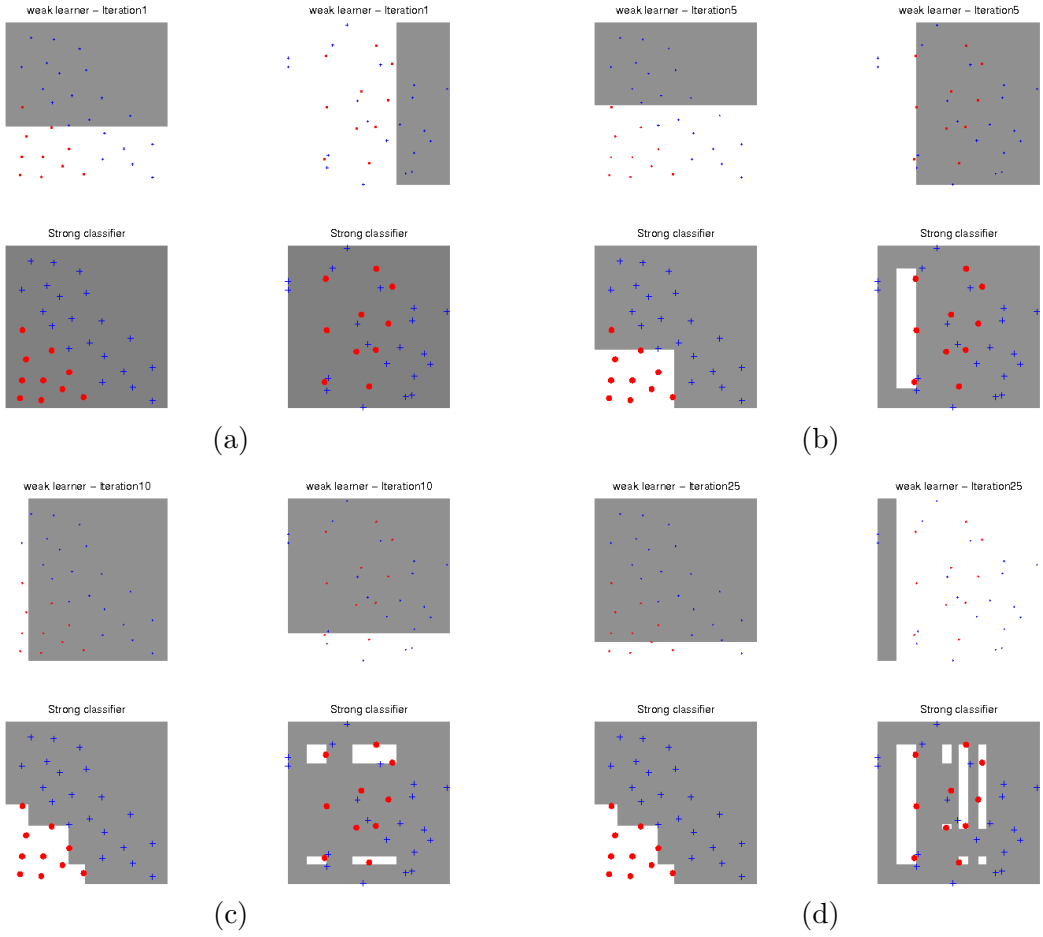


Figure 5.11: Evolution of the BRF algorithm when the second problem suffers from noisy features with a correlation of 65% at (a) Iteration 1 (b) Iteration 5 (c) Iteration 10 (d) Iteration 25

of iterations is needed when some information is shared between the two problems about the labels. In this simple example, if we compare the figures 5.10 and 5.12, we see that a correlation as low as 0.35 is helping to solve the problem as well as when the correlation is higher and results in lower classification errors in a fewer number of iterations. Of course, that will not always be the case. Depending on the problem to solve, more or less correlation between the labels might be needed. However, the benefit of using context is here qualitatively and also quantitatively proven.

There are two limit cases to distinguish:

- when the correlation is 0, the compatibilities does not bring any information. The

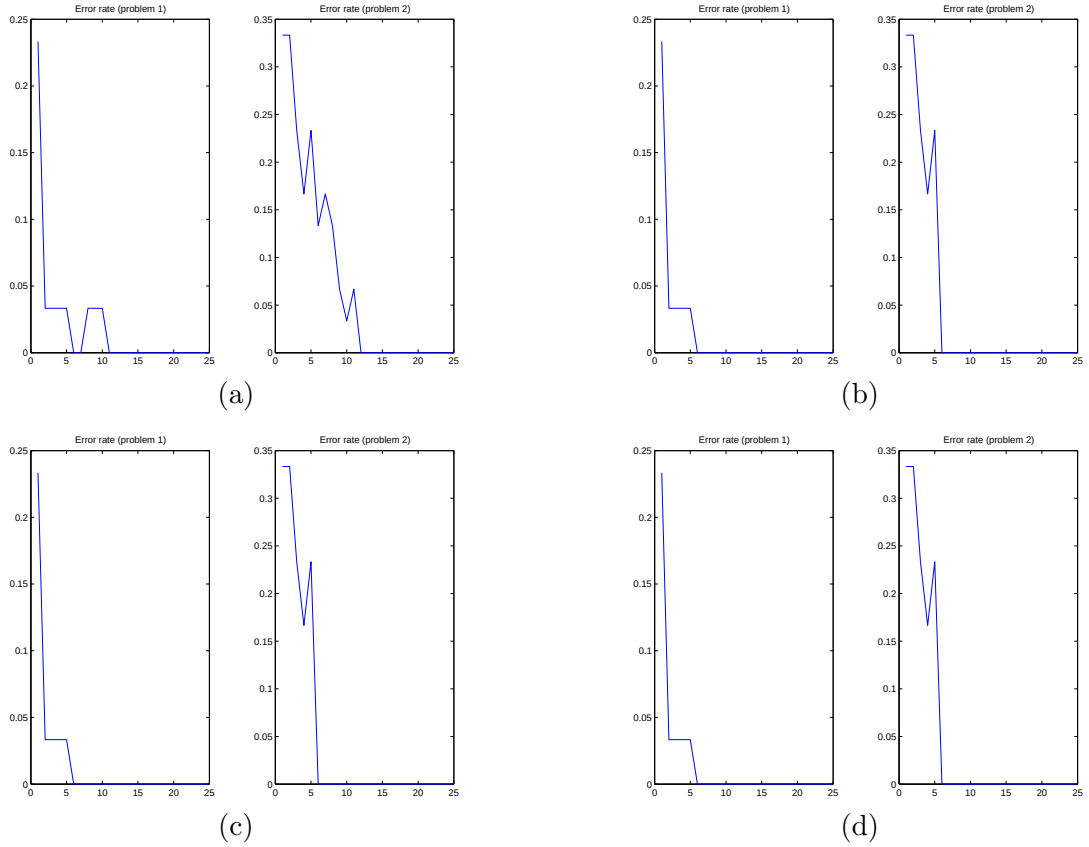


Figure 5.12: Error rates in function of the number of iterations for the BRF algorithm with a correlation between labels of (a) 0 (b) 0.35 (c) 0.65 (d) 1

two problems are solved similarly as in traditional boosting: there are solved independently.

- when the correlation is 1, the knowledge about one problem gives the solution about the second problem. If one problem is easily separable and the other is not, BRF will use the compatibilities to solve efficiently the second problem.

For all cases in between, a high correlation between the labels will help to solve the learning problem with less iterations and we can expect that it will result in highly discriminative models which are also less over-fitted when correlation is present. A very powerful novelty about this approach is that the contextual relationships are automatically discovered and do absolutely not require to be pre-specified.

5.5 Large scale experiments

Evaluation protocol

We define:

- N_t: the total number of story boundaries in the ground truth ;
- N_d: the number of story boundaries missed by the algorithm, but existing in the ground truth (missed detections) ;
- N_i: the number of story boundaries detected by the algorithm, but not existing in the ground truth (false detections) ;
- N_c: the number of story boundaries found by the algorithm and existing in the ground truth (correct detections).

Precision, recall and F-measure are defined from these updated definitions in the same manner as in the previous chapter. When considering the labeling task, we introduce another performance measure which is the classification accuracy simply defined as the ratio of correctly classified instances and of the total number of instances.

5.5.1 Semantic video segment labeling

The evaluation of performances for the semantic video segment labeling requires the presence of the annotations that were manually and collaboratively generated [156]. This reduces our training and test set to the development data of TRECVid 2003. The training and test sets contain video files from CNN broadcast news from the TRECVid corpus. 50 percent of the data was used for training and 50 percent for testing. The CNN collection contains 34 videos and every video is made of an average of 300 candidate points. In the following, “contextual” refers to the proposed contextual model which takes into account the compatibilities between labels and “non-contextual” refers to the regular boosting algorithm which classifies the data considering only the link between low-level features and labels.

Non-contextual / Contextual

Visual	Anchor	Outdoor	Financial	Weather	Ads
	57/**67**	44/**53**	65/**71**	82/**83**	56/**55**
Audio	Speech	Music	Noise		
	67/**78**	60/**64**	30/**32**		
Text	Cluster1	Cluster2	Cluster3	Cluster4	...
	80/**83**	72/**76**	75/**70**	74/**81**	...

Table 5.5: Percent of correct classification for the semantic labels with non-contextual learning.

Quantitative evaluation

Table 5.5 highlights the fact that contextual modeling helps to improve performances for the task of semantic labeling. In bold characters is given the percent of correct classification when using the contextual model, to be compared with the values of the non-contextual model in normal characters. Generally, performances are improved for semantic labeling showing that the contextual model actually supports inference in the way we expected. Nonetheless we note a few cases for some textual clusters or Ads detection where the contextual model did not help. The reason is that the interactions received by other labels were noisy due to classification errors and did not help to improve performances.

It should be noted that the classification into the audio classes does not look very good when compared with existing studies showing results reaching more than 90% of accuracy [11]. After investigation, we found that these studies are using collections which are quite small and homogeneous in the type of speech, music and noise signals. The problem here with news broadcast is way more complex with the superposition of many audio sources at the same moment (jingle, music, speech and noises) and with the high heterogeneity of the data.

5.5.2 News story segmentation

Quantitative evaluation

The measure of performance for the semantic segmentation is done by calculating the precision and recall of the system. Every reference boundary is enlarged by a tolerance window of 5s in both directions according to the TRECVid evaluation protocol.

The point of the table 5.6 is to demonstrate that the use of context shows a clear

Random choice			Non-contextual			Contextual		
P	R	F-measure	P	R	F-measure	P	R	F-measure
20	20	20	57	62	59.4	64	66	65

Table 5.6: Precision and Recall for the news story segmentation with a contextual or non-contextual model.

improvement in performance compared with the non-contextual classification. Of course, these performances can be improved further by using more advanced features that would be more discriminative, but at the cost of the capability for the algorithm to adapt to various news broadcast video collection. The performance of our algorithm compares favorably to state of the art approaches as we will show later on.

Qualitative evaluation

The table 5.7 shows the list of semantic concepts sorted so that the ones with the higher weights are shown first. This means that these semantic concepts have been found to be the most useful to reduce classification errors about the news story boundary labels after learning. The information contained in this table is important to qualitatively understand the behavior of the learned model and gives idea on which concepts to focus the efforts to design new low-level features. The label "News subject monologue after" is dominating and this is true that in the CNN and ABC news broadcast, a common pattern is too finish a story with a reporter standing outdoor and to start the new story by the news subject monologue. It is an easy interpretation, but more complex patterns are hidden in the structure of the model. In the additive model, several semantic concepts appear several time with different weights and are mixed with the cues coming from low-level features. The captured patterns are often difficult to interpret and statistical by nature.

We observe that there is a similarity between the features which are given top weights by the classification algorithm and the table corresponding to the most informative labels described in the section 5.3.3. The high ranking of "news subject monologue after", "Graphics and text" in both tables show that these labels are correctly classified and used by the algorithm. "Face after" is ranked number 1 of informativeness and "Sport event" is number 5, but are not present in the table of utility due to the lack of a good face/sport detector in our framework. It shows that there is room for improvment by the adding such detectors or by providing features which would help the boosting algorithm

to classify them correctly.

1	News subject monologue after (V)
2	Graphics and text after (V)
3	Graphics and text before (V)
4	Non studio settings after (V)
5	Sport event after (V)
6	Text cluster 4 after (T)
7	Text cluster 13 after (T)
8	Text cluster 16 before (T)
...	...

Table 5.7: Semantic concepts utility for news story segmentation. (V) for visual ; (T) for textual ; (A) for audio.

We will show some typical errors made by the model on the test set. We show in Figure 5.13 an example of a properly detected news story boundary. The two key frames on the left and the two key frames on the right are representative of the video content before and after the transition. The words obtained by the speech recognition algorithm are shown below the key frames. Several cues help the algorithm for the inference: the presence of a long silence, the appearance of the anchor shot when starting the new story, the text “A.B.C news” to conclude the old story are all highly informative. The Figure 5.14 shows a missed news story boundary. It is difficult to say exactly why the inference did fail, because it is the consequence of many interacting factors but we can notice that the topics are pretty similar (U.S. justice related) and that no clear visual, audio or textual cues are given. This illustrates the imperfection of the system: highly informative indicators about a news story transition is necessary for proper detection. The Figure 5.15 shows a false detection for a news story boundary. In this example, the classification failed because of several factors which also are hard to distinguish by design of the algorithm. However by viewing the video, a long silence added to the presence of the static graphic after the candidate point may partly explain the decision. As a general consideration, these examples show that the modeling goes quite far with our contextual model, but also exhibit the high ambiguity of the problem which is obviously not yet resolved.

Figure 5.13: An example of news story transition properly detected.

Figure 5.14: An example of a missed news story transition.

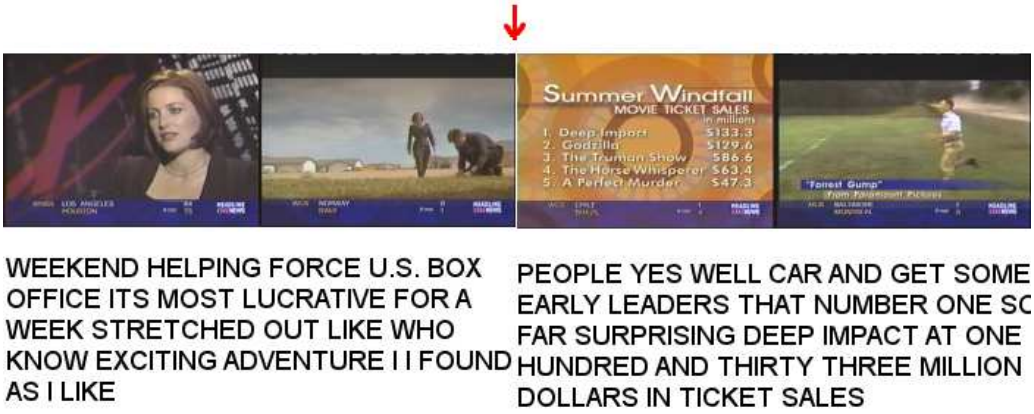


Figure 5.15: An example of a false news story transition.

5.5.3 Performance comparison with TRECVID 2004 participants

TRECVid 2004 evaluation protocol

In this section, we will describe how the evaluation has been performed during TRECVID 2004 for the news story segmentation task and compare our results by using strictly the same settings. The evaluation of performances for news story segmentation uses TRECVID 2003 development+test data for training and TRECVID 2004 data for testing. The collaborative annotation efforts is provided for the TRECVID 2003 development data only which makes the TRECVID 2003 test data hardly usable to generate training instances. We will then consider 70 videos of the complete set (we did not use 43 videos due to file compatibility issues with our MPEG decoder) for training and 128 videos from CNN/ABC for testing. The ground-truth has been manually annotated, but surely contains errors due to the large scale of the training set, the tool used for annotations and also general human ability to make errors.

Participants

A total of 8 groups did participate to the news story segmentation task in TRECVID2004. We selected three groups in order to compare our performances.

The best performances have been obtained by the KDDI group [86] with a F-measure near 70% for news story boundaries detection. The features are low-level and generic considering audio, motion and color as well as shot editing features. The classification is performed with different Support Vector Machine (SVM)s using section-specialized segmentation for “top stories”, “headlines sports”, etc. The way the system is trained requires manual care in order to extract the specialized-sections out of the rest of the training data, annotate them and produce a specific training. The performances of the KDDI group can be considered as a higher bound on what we might achieve, because we decided not to consider specialized-sections so that our approach remains generic and adaptable to other video collections.

The second group in terms of performance is IBM [167]. The approach is based on visual cue cluster construction based on the information bottleneck principle [168]. A set of clusters is generated such that the information about the news story boundary information is preserved. The features are then mapped on the space of the clusters by calculating membership probabilities. Inference is made thanks to a SVM classifier on the

reduced and informative feature space. Richness and fine tuning of the feature set explains a lot about the good performances.

The participating group obtaining the worst performances is the Imperial College's one. This approach considers that a new story starts and finishes systematically with an anchor shot. The anchor person detection is based on a k-nearest neighbor classifier. This is a very straightforward baseline approach.

Performances

Performances	Imperial College	IBM	KDDI	Our algorithm
Precision	30	61	**69**	63
Recall	31	63	**68**	64
F-measure	30.5	62	**68.5**	63.5

Table 5.8: Performance comparisons between selected TRECVID 2004 participants.

The table 5.8 shows that our algorithm offers good performances compared to other participants. The KDDI is the best in terms of performance at the cost of developing section-specific classifiers.

The proposal that we made to use contextual formulation to solve the news story boundary detection problem has proven to be useful to improve consistency of our Boosting classifier. However the weak learners are possibly not doing the best possible job to capture discrimative patterns inside our data and that is the reason why our performances are matching, but not really increasing the ones obtained by support vector machines. An idea of future research would be to combine the best of both worlds (support vector machine + contextual relationships) to maybe obtain an improvment in performances in this area.

We remark that for all groups the performances are still not high enough to build solid application on top of such a segmentation. The F-measure should ideally get closer to 90% in the future, but important new breakthroughs in this research field are still needed to reach this goal.

5.6 Summary

We developed a story segmentation method that is able to capture parts of the semantic information contained in the video by the use of generic low-level features. Due to the generality of our set of features, it is expected that this methodology is applicable for the structuring of different video collections. We presented a general approach usable in real life scenarios by avoiding carefully ad-hoc procedures: before training of the model with the development of problem-specific features (specific jingle detections, etc.) or during training by choosing manually to train section-specific classifiers to optimize our performances (specific classifiers for "top-stories" or "headline sports").

In this chapter, we have proposed a contextual model for the semantic segmentation of videos into stories by allowing labels interactions between different modalities as well as the observed data. Contextual models have not been proposed to solve such classification applications before. We believe that this is one very interesting direction for research to move forward in attempting to bridge the semantic gap. Boosted Random Fields provide a principled approach and an effective optimization framework to estimate the model parameters. The results on the TRECVid corpus validate the advantages of this model over non-contextual classification. Another remark is that the level of performance is still too low to build any application on top of automatic news story segmentation algorithms for the time being. It can still help annotators to reduce their workload, but human help is still necessary to offer solid semantic-level applications. Open issues are still directly related to the feature extraction steps: genericity and computational efficiency are prerequisites, but should also be informative and discriminative to offer a better playground to machine learning models and algorithms.

Chapter 6

Conclusion and future research perspectives

6.1 Summary

In this thesis, we have investigated the structuring of video document at several levels: the information-level, the editing-level and the semantic-level. These three levels of segmentation group together the frames according to different criteria: visual homogeneity, shot transition detection and topical homogeneity.

The information-level segmentation decomposed the video in visually homogeneous segments. To demonstrate the usefulness of this decomposition, we developed a user and data adaptive video browsing application built from it. The browsing application offers significant advantages over existing approaches in video browsing by facilitating coarse-to-fine exploration with respect to the complexity of the data.

The editing-level segmentation is directly related to the problem of shot detection and is recognized as a difficult problem when dealing with gradual transitions. The information-level segmentation has proved to be useful to develop a generic algorithm which rely on a simple set of assumptions instead of developing an ad-hoc detector for each different type of transitions. Our generic approach has been evaluated and favorably compared with participants of TRECVid particularly for the detection of gradual transitions.

The semantic-level segmentation requires the specialization of the dataset to take

advantage of "a priori" information. We chose to focus on news story segmentation. In order to capture as much "a priori" information as possible, a contextual model was proposed to take advantage of the interactions between multimodal label as well as low-level features. The performances of the algorithm have been favorably compared to state of the art approaches on the same dataset. We demonstrated that the inclusion of context brings informative support for such semantic inference problems.

Many problems arose when attempting to reach the semantic level. The trade-off between the descriptive and discriminative power of the features is still found empirically. There is no formal way to our knowledge to find an optimal trade-off for this issue. We as many other researchers have chosen to provide as many features as we could in a pool. Then we let the machine learning algorithm dig into it and make its own selection of features. The dimensionality of the feature vector still requires to be limited or the computational complexity will explode. Another difficulty and strong constraint was that we attempted to make our algorithms as parameter free as possible. This has the advantage of making possible the future use of these softwares by documentalists who are not specifically trained to tune such complex systems.

We did validate our approaches on a massive set of audiovisual documents thanks to the data and annotations coming from the TRECVid initiative. This would not have been possible a few years ago. We had the possibility to compare the performance of our algorithms with other research groups. This sort of competition will ineluctably trigger and propagate progress in the field by clearly showing everybody where lies in reality the state of the art. It is our opinion that researchers often suffer from autism by refusing fair comparison of results or from the (conscious?) disregard for the true applicative use of their works. CBVR has gotten rid of these problems thanks to TRECVid for a given set of pre-defined tasks such as shots detection and news story segmentation. Of course, research should not be reduced to the achievement of good performance levels for a given application. Other quality criteria such as the genericity of the approaches and the originality of algorithm proposals are more subjective and difficult to compare but should not be forgotten for the sake of performances.

6.2 Future research perspectives

As future research perspectives, we propose the following possible applicative extensions of our work. The video browsing application could be enhanced by merging transparently

information coming our three levels of segmentation from the highest level of details to the representation of semantic stories. A scale parameter tuned interactively by the user would enable the transition between the different representations.

The knowledge of the semantic-level segmentation offers the possibility to visualize new topical representation of a video collection as a whole. The classification into topics is facilitated by the knowledge of the segmentation and vice versa. Offering more than a simple table of content, the semantic labeling offers the potential use of intra/inter collection relationships for navigation.

In addition, the work can also be extended by considering different semantic units for various video collections such as soccer, tennis, documentaries, etc. Specific models and features will help enhance the performance. Contextual models have the potential to help improve consistency in a lot of different ways which still have to be explored.

As we suggested earlier, a possible performance increase could be obtained by combining the best charateristics of contextual models and support vector machines. The two approaches would be interesting to combine because they are fundamentally not making the same use of information: relationships between labels for the contextual approach and discriminative separation of the feature space for the support vector machine. Also the two approaches showed independently to be the best classifiers in term of performance in the experiments. This combination is far from obvious, because we saw that the contextual approach is limited to the use of simple weak classifiers in order to keep the computational complexity under control. It would be nice to use support vector machines instead and more research is likely to produce interesting results in that direction if this barrier related to the computational complexity is broken.

Finally, it shall be noted that nobody knows what the future holds. The MPEG7 standard gives the possibility for audio-visual producers to make available a large range of annotations. The cost of the annotation effort will decrease very rapidly thanks to the research in CBVR. Any information retrieval company has also the possibility to process the video content of a given consumer, extract some relevant information and make it easily available to him. For sure the information revolution is at work and the way documentalists search and navigate into video collections might change radically in the mid-term future.

Bibliography

[1] P. Aigrain, H. Zhang, and D. Petkovic. Content-based representation and retrieval of visual media: A state-of-the-art review. *Multimedia Tools and Applications*, 3(3):179–202, 1996.

[2] I. Koprinska and S. Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16(5):477–500, 2001.

[3] B. Janvier, E. Bruno, S. Marchand-Maillet, and T. Pun. Information-theoretic framework for the joint temporal partionning and representation of video data. In *Proceedings of the 3rd International Workshop on Content-Based Multimedia Indexing, CBMI'03, Rennes, France*, September 2003.

[4] Y. Rui, T. S. Huang, and S. Mehrotra. Exploring video structure beyond the shots. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, page 237, 1998.

[5] H. Sundaram and S.-F. Chang. Computable scenes and structures in films. In *IEEE Transactions on Multimedia*, volume 4, pages 482–491, 2002.

[6] D. Gatica-Perez, A. Loui, and M.T. Sun. Finding structure in consumer videos by probabilistic hierarchical clustering. *IEEE Transactions on Circuits and Systems for Video Technologys*, 13(6):539–548, 2003.

[7] E. Kijak, G. Gravier, L. Oisel, and P. Gros. Audiovisual integration for sport broadcast structuring. *Multimedia Tools and Applications (to appear)*, 2005.

[8] L. Chaisorn, T-S. Chua, C-K. Koh, Y. Zhao, H. Xu, H. Fend, and Q. Tian. A two-level multi-modal approach for story segmentation of large news video corpus. In *TrecVid Workshop*, Gaithersburg, 2003.

[9] A. Merlino, D. Morey, and M. Maybury. Broadcast news navigation using story segmentation. In *ACM Multimedia*, pages 381–391, 1997.

[10] J.C. Huang Y. Wang, Z. Liu. Multimedia content analysis - using both audio and visual clues. *IEEE signal processing*, 17(6):12–36, 2000.

[11] L. Lu, H. Jiang, and H.J. Zhang. A robust audio classification and segmentation method. In *Proceedings of the 9th ACM International Conference Multimedia*, pages 203–211, 2001.

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

[13] D. Chen, H. Bourlard, and J-P. Thiran. Text identification in complex background using svm. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2001.

[14] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the 9th ACM International Conference on Multimedia*, pages 107–118, Ottawa, Canada, 2001.

[15] M. Naphade, J. Smith, and F. Souvannavong. On the detection of semantic concepts at trecvid. In *ACM Multimedia*, pages 660–667, 2004.

[16] Y. Wu, C. Y. Lin, E. Y. Chang, and J. R. Smith. Optimal multimodal fusion for multimedia data analysis. In *ACM Multimedia*, pages 572–579, 2004.

[17] A. Amir, J. Argillander, M. Campbell, A. Haubold, G. Iyengar, S. Ebadollahi, F. Kang, M. R. Naphade, A. Natsev, J. R. Smith, J. Tesic, and T. Volkmer. Ibm research trecvid-2005 video retrieval system. In *Proceedings of the TRECVID 2005 workshop*, Gaithersburg, 2005.

[18] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57:137–154, 2004.

[19] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems 17 (NIPS)*, 2004.

[20] A. Hauptmann, R.V. Baron, M.-Y. Chen, M. Christel, P. Duygulu, C. Huang, R. Jin, W.-H. Lin, T. Ng, N. Moraveji, N. Papernick, C.G.M. Snoek, G. Tzanetakis, J. Yang,

R. Yang, and H.D. Wactlar. Informedia at trecvid 2003: Analyzing and searching broadcast news video. In *TREC Video Retrieval Evaluation Online Proceedings*, Gaithersburg, 2003.

[21] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun. Unsupervised discovery of multi-level statistical video structures using hierarchical hidden markov models. In *IEEE International Conference Multimedia and Expo (ICME)*, 2003.

[22] S-F. Chang, W. Chen, and H. Sundaram. Videoq: A fully automated video retrieval system using motion sketches. In *Proceedings 4th IEEE Workshop on Applications of Computer Vision*, 1998.

[23] C.G.M. Snoek and M. Worring. Multimedia event-based video indexing using time intervals. *IEEE Transactions on Multimedia*, 7(5):638–647, 2005.

[24] J. Z. Wang. *Integrated region-based image retrieval.* Kluwer Academic Publisher, 2001.

[25] D. McG. Squire, W. Müeller, and H. Müeller. Content-based query of image databases: inspirations from text retrieval. *Pattern Recognition Letters (Selected Papers from the 11th Scandinavian Conference on Image Analysis SCIA '99)*, 21(13-14):1193–1198, 2000.

[26] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference On Visual Information and Information Systems*, pages 509–516, 1999.

[27] M. Flickner. Query by image and video content. *IEEE Computer*, pages 23–32, 1995.

[28] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C-F. Shu. The virage image search engine: an open framework for image management. In *Proceedings of SPIE Conference on Image Storage and Retrieval Systems*, pages 76–87, 1996.

[29] A. Hampapur, A. Gupta, B. Horowitz, C-F. Shu, C. Fuller, J. Bach, M. Gorkani, and R. Jain. Virage video engine. In *Proceedings of SPIE Conference on Image Storage and Retrieval Systems*, volume 3022, 1997.

[30] J. J. Rocchio. *The Smart Retrieval System: Experiments in Automatic Document Processing.* Prentice Hall, 1971.

[31] I. Ruthven and M. Lalmas. A survey on the use of relevance feedback for information access systems. In *Knowledge Engineering Review*, volume 18, pages 95–145, 2003.

[32] L. Boldareva and D. Hiemstra. Interactive content-based retrieval using pre-computed object-object similarities. In *Proceedings of the International Conference on Image and Video Retrieval (CIVR)*, pages 308–316, 2004.

[33] A. G. Hauptmann and M. G. Christel. Successful approaches in the trec video retrieval evaluations. In *Proceedings of the 12th annual ACM International Conference on Multimedia*, pages 668–675, 2004.

[34] H. Lee and A.F. Smeaton. Designing the user interface for the físchlár digital video library. *Journal Digital Info*, 2(4), 2002.

[35] P. Aigrain and P. Joly. Discrete visual manipulation user interfaces for video. In *Proceedings RIAO"94 Conference*, volume 2, pages 12–17, 1994.

[36] H. J. Zhang, S.W. Smoliar, and J.H. Wu. Content-based video browsing tools. In *Proceedings of IS&T/SPIE Conference on Multimedia Computing and Networking*, volume 2417, 1994.

[37] M.M. Yeung, B.-L. Yeo, W. Wolf, and B. Liu. Video browsing using clustering and scene transitions on compressed sequences. In *Proceedings of IS&T/SPIE Conference on Multimedia Computing and Networking*, volume 2417, pages 399–413, 1995.

[38] R. Rao, J. Petersen, M. Hearst, J. Mackinlay, S. Card, L. Masinter, P.-K. Halvorsen, and G. Robertson. Rich interaction in the digital video library. In *Communications of the ACM*, pages 29–39, 1995.

[39] M. Rautiainen, T. Ojala, and T. Seppänen. Cluster temporal browsing of large news video databases. In *Proceedings IEEE International Conference on Multimedia and Expo*, volume 2, pages 751–754, 2004.

[40] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias. Efficient video summarization based on a fuzzy video content representation. In *Proceedings IEEE International Symposium Circuits and Systems*, volume 4, page 195, 2000.

[41] M. A. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding. In *Proceedings IEEE International Workshop on Content-Based Access of Image and Video Databases*, pages 61–70, 1998.

[42] S-F. Chang, W. Hsu, L. Kennedy, L. Xie, A. Yanagawa, e. zavesky, and D-Q. Zhang. Columbia university trecvid-2005 video search and high-level feature extraction. In *Proceedings of the TRECVID 2005 workshop*, Gaithersburg, 2005.

[43] N. Moënne-Loccoz, B. Janvier, S. Marchand-Maillet, and E. Bruno. Handling temporal heterogeneous data for content-based management of large video collections. *Multimedia Tools and Applications*, page to appear, 2005.

[44] A. F. Smeaton, W. Kraaij, and P. Over. Trecvid 2003 - an overview. In *TREC Video Retrieval Evaluation Online Proceedings*, Gaithersburg, 2003.

[45] S-F. Chang. The holy grail of content-based media analysis. *IEEE Multimedia*, 9(2):6–10, 2002.

[46] M. Tico and P. Kuosmanen. A method for color histogram creation for image retrieval. In *Proceedings in NORSIG: IEEE Nordic Signal Processing Symposium*, 2000.

[47] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–22, 1991.

[48] A. Khokar, R. Ansari, and H. Malik. Quantized cielab* space and encoded spatial structure for scalable indexing of larg color image archives. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 6, 2000.

[49] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *ACM Multimedia Conference*, pages 65–73, 1996.

[50] J. Huang, S. Ravi Kumar, M. Mitra, W-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *International Conference on Computer Vision and Pattern Recognition*, pages 762–768, 1998.

[51] W. Y. Ma and H. Zhang. Benchmarking of image features for content-based image retrieval. In *32th IEEE Asilomar Conference Signals, Systems, Computers*, volume 1, pages 253–257, 1998.

[52] Y. Rui, T. S. Huang, and S. Mehrotra. An adaptive clustering algorithm for color quantization. *Pattern Recognition Letters*, 21:341–356, 2000.

[53] X. Wan and C. J. Kuo. A multiresolution color clustering approach to image indexing and retrieval. *International Journal of Computer Vision*, 7(1):3705–3708, 1998.

[54] S. M. Smith. A new class of corner finder. In *Proceedings 3rd British Machine Vision Conference*, pages 139–148, Leeds, UK, 1992.

[55] C. G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[56] G. Welch and G. Bishop. An introduction to the kalmam filter. In *http://www.cs.unc.edu/ welch/kalman*, 2004.

[57] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings Europ. Conference Computer Vision*, pages 237–252, 92.

[58] M. Black. Robust incremental optical flow. In *PhD thesis*, 92.

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

[60] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 14, pages 367–383, 92.

[61] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proceedings Computer Vision and Pattern Recognition, CVPR-91*, pages 296–302, Maui, USA.

[62] P. J. Rousseeuw and A. M Leroy. *Robust regression and Outlier detection*. John Wiley and Sons, 1987.

[63] H. Yi, D. Rajan, and L.T. Chia. A new motion histogram to index motion content in video segments. In *Pattern Recognition Letters*, volume 26, pages 1221–1231, 2005.

[64] B. Fauvet, P. Bouthemy, P. Gros, and F. Spindler. A geometrical key-frame selection method exploiting dominant motion estimation in video. In *International Conference on Image and Video Retrieval (CIVR)*, volume 3115, pages 419–427, 2004.

[65] E. Bruno and D. Pellerin. Video structuring, indexing and retrieval based on global motion wavelet coefficients. In *ICPR02*, volume 3, pages 287–290, 2002.

[66] S.C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.

[67] R. Jain, R. Kasturi, and B. Schunck. *Machine Vision*. New York: MIT Press and McGrawHill, 1995.

[68] D. A. Pollen and S. F. Ronner. Phase relationship between adjacent simple cells in the visual cortex. Number 212, pages 1409–1411, 1981.

[69] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *Pattern Analysis and Machine Intelligence*, 18(8):836–842, 1996.

[70] M. N. Do and M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kullback-leiber distance. *IEEE Transactions on Image Processing*, 11(2), 2002.

[71] S. Ullman. *High Level Vision*. MIT Press, 1997.

[72] H.H.S. Ip, A.K.Y. Cheng, W.Y.F Wong, and F.Feng. Affine-invariant sketch-based retrieval of images. In *Computer graphics international*, 2001.

[73] P. Montesinos, V. Gouet, and R. Deriche. Differential invariants for color images. In *14th International Conference on Pattern Recognition*, 1998.

[74] D. Hilbert. *Theory of algebraic invariants*. Cambridge University Press, 1890.

[75] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[76] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[77] Z. Liu, Y. Wang, and T. Chen. Audio feature extraction and analysis for scene segmentation and classification. In *Journal VLSI Signal Processing Syst. Signal, Image, Video Technology*, volume 20, pages 61–79, 1998.

[78] E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search and retrieval of audio. In *IEEE Multimedia Mag*, volume 3, pages 27–36, 1996.

[79] N. Jayant, J. Johnston, and R. Safranek. Signal compression based on models of human perception. In *Proceedings IEEE*, volume 81, pages 1385–1422, 1993.

[80] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeatures speech/music discriminator. In *International Conference Acoustic, Speech and Signal Processing*, volume 2, pages 1331–1334, 1997.

[81] A. de Cheveigne and H. Kawahara. Comparative evaluation of f0 estimation algorithms. In *Proceedings of Eurospeech*, pages 2451–2454, 2001.

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

[83] H. Wactlar, M. Christel, Y. Gong, and A. Hauptmann. Lessons learned from building a terabyte digital video library. *IEEE Computer*, 32(2):66–73, 1999.

[84] N.Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th Ann. International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR*, Tampere, Finland, 2002.

[85] G. Chechik. and N. Tishby. *Advances in Neural Information Proceedings Systems NIPS 15*, chapter Extracting relevant structures with side information. MIT press, 2002.

[86] K. Hoashi, M. Sugano, M. Naito, K. Matsumoto, F. Sugaya, and Y. Nakajima. Trecvid story segmentation based on content-independent audio-video features. In *TrecVid Workshop*, Gaithersburg, 2004.

[87] C. S. Wallace and P. R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society*, 49(3):240–265, 1987.

[88] M. J. Swain. Interactive indexing into image databases. In *Proceedings SPIE Conference Storage and Retrieval in Image and Video Databases*, pages 173–187, 1993.

[89] H.J. Zhang, A. Kankanhalli, and S.W. Smoliar. Automatic partitioning of full-motion video. In *Multimedia Systems 1*, pages 10–28, 1993.

[90] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. In *Visual Database Systems II*, pages 113–127, 1995.

[91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

[92] R. Zabih, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying production effects. *Multimedia Systems*, 7(2):119–128, 1999.

[93] G. M. Quénot, D. Moraru, S. Ayache, Mbarek Charhad, Mickaël Guironnet, Lionel Carminati, Philippe Mulhem, J. Gensel, D. Pellerin, and L. Besacier. Clips-lis-lsr-labri experiments at trecvid 2004. In *Proceedings of the TRECVID 2004 workshop*, Gaithersburg, 2004.

[94] P. Bouthemy, M. Gelgon, and G. Ganansia. A unified approach to shot change detection and camera motion characterization. volume 9, pages 1030–1044, 1999.

[95] S.-C. Chen, M.-L. Shyu, and C. Zhang. Video scene change detection method using unsupervised segmentation and object tracking. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 57–60, 2001.

[96] D. Mumford and J. Shah. Boundary detection by minimizing functionnals. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1985.

[97] Thomas C. M. Lee. A minimum description length–based image segmentation procedure, and its comparison with a cross-validation–based segmentation procedure. *Journal of the American Statistical Association*, 95(449):259–??, 2000.

[98] L. J. Fitzgibbon, L. Allison, and D. L. Dowe. Minimum message length grouping of ordered data. In H. Arimura and S. Jain, editors, *Proceedings of the Eleventh International Conference on Algorithmic Learning Theory (ALT2000)*, LNAI, pages 56–70, Berlin, 2000. Springer-Verlag.

[99] R. A. Baxter and D. L. Dowe. Model selection in linear regression using the mml criterion. In J. A. Storer and M. Cohn, editors, *Proceedings 4'th IEEE Data Compression Conference*, page 498, 1994.

[100] W. D. Fisher. On grouping for maximum homogeneity. *Journal of the American Statistical Association*, 53(284), 1958.

[101] Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata. Videomap and videospaceicon: Tools for anatomizing video content. In *Proceedings of ACM INTERCHI*, pages 131–141, 1993.

[102] A.M. Ferman and A.M. Tekalp. Multiscale content extraction and representation for video. In *Multimedia Storage and Archiving Systems II, Proceedings SPIE 3229*, pages 23–31, 1997.

[103] Y. Taniguchi, A. Akutsu, and Y. Tonomura. Panoramaexcerpts: Extracting and packing panoramas for video browsing. In *Proceedings ACM Multimedia 97*, pages 427–436, 1997.

[104] B. Günsel, A. Müfit Ferman, and A. Murat Telkap. Temporal video segmentation using unsupervised clustering and semantic object tracking. *Journal of Electronic Imaging*, 7(3):592–604, 1998.

[105] S. Uchihashi and J. Foole. Summarizing video using a shot importance measure and a frame-packing algorithm. In *Proceedings IEEE ICASSP*, 1999.

[106] X.D. Zhang, T.Y. Liu, K.T. Lo, and J. Feng. Dynamic selection and effective compression of key frames for video abstraction. In *Pattern recognition letters*, volume 24, pages 1523–1532, 2003.

[107] H.S. Chang, S. Sull, and S.U. Lee. Efficient video indexing scheme for content-based retrieval. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 9, pages 1269–1279, 1999.

[108] T.-Y. Liu and X.-D. Zhang. Shot reconstruction degree: a novel criterion for key frame selection. *Pattern Recognition Letters*, 25:1451–1457, 2004.

[109] A. Hanjalic, R.L. Lagendijk, and J. Biemond. *Image Databases and Multi-media Search*, chapter A new method for Key Frame based Video Content Representation. Eds. World Scientific, 2001.

[110] L. Guigues, H. Le Men, and J.P. Cocquerez. Analyse et representation ensembles-echelle d'une image. In *19e colloque du traitement du signal et des images (Gretsi'03)*, 2003.

[111] A. F. Smeaton, W. Kraaij, and P. Over. Trecvid 2004 - an overview. In *TREC Video Retrieval Evaluation Online Proceedings*, Gaithersburg, 2004.

[112] A. Hampapur, R. Jain, and T. Weymouth. Digital video segmentation. In *Proceedings ACM Multimedia*, pages 357–364, 1994.

[113] D. Zhang, W. Qi, and H. Zhang. A new shot boundary detection algorithm. *Lecture Notes in Computer Science*, 2195:63–??, 2001.

[114] O.D. Robles, P. Toharia, A. Rodriguez, and L. Pastor. Automatic video cut detection using adaptive thresholds. In *Proceedings Visualization, Imaging and Image Processing*, Marbella, Spain, 2004.

[115] C. PeterSohn. Fraunhaufer hhi at trecvid 2004: Shot boundary detection system. In *TREC Video Retrieval Evaluation Online Proceedings*, Gaithersburg, 2004.

[116] T. D. Wickens. *Elementary Signal Detection Theory.* Oxford University Press, 2001.

[117] A. Hanjalic. Shot-boundary detection: Unraveled and resolved. *IEEE transactions on circuits and systems for video technology*, 12(2), 2002.

[118] M. Luo, D. DeMenthon, and D. Doermann. Shot boundary detection using pixel-to-neighbor image differences in video. In *Proceedings of the TRECVID 2004 workshop*, Gaithersburg, 2004.

[119] C. PeterSohn. Dissolve shot boundary determination. In *Proceedings IEEE European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, pages 87–94, Gaithersburg, 2004.

[120] C. PeterSohn. Wipe shot boundary determination. In *Proceedings IS&T/SPIE Electronic Imaging 2005, Storage and Retrieval Methods and Applications for Multimedia*, pages 337–346, 2005.

[121] W. J. Heng and K. N. Ngan. Shot boundary refinement for long transition in digital video sequence. *IEEE transactions on multimedia*, 4(4), 2002.

[122] R. Lienhart. Comparison of automatic shot boundary detection algorithms. In *Storage and Retrieval for Still Image and Video Databases VII Proceedings SPIE 3656-29*, 1999.

[123] R. Ruiloba, P. Joly, S. Marchand-Maillet, and G. Quenot. Towards a standard protocol for the evaluation of video-to-shots segmentation algorithms. In *International Workshop in Content-Based Multimedia Indexing (CBMI)*, Toulouse, France, 1999.

[124] A. Amir. The ibm shot boundary detection system at trecvid 2003. In *Proceedings of the TRECVID 2003 workshop*, Gaithersburg, 2003.

[125] N. Vasconcelos and A. Lippman. Bayesian modeling of video editing and structure: semantic features for video summarization and browsing. In *Proceedings of IEEE International Conference on Image Processing*, pages 550–555, 1998.

[126] Y. Gong, L. T. Sin, C. H. Chuan, H. Zhang, and M. Sakauchi. Automatic parsing of tv soccer programs. In *Multimedia Computing and Systems*, pages 167–174, Washington DC, 1995.

[127] Y. L. Chang, W. Zeng, I. Kamel, and R. Alonso. Integrated image and speech analysis for content-based video indexing. In *Multimedia Computing and Systems*, pages 306–313, 1996.

[128] G. Sudhir, J.C.M. Lee, and A.K. Jain. Automatic classification of tennis video for high-level content-based retrieval. In *IEEE International Workshop on Content-Based Access of Image and Video Databases*, page 81, 1998.

[129] M. Yeung, B.-L. Yeo, and B. Liu. Time-constrained clustering for segmentation of video into story units. In *International Conference on Pattern Recognition*, pages 375–380, 1996.

[130] W. Hsu and S.F. Chang. Generative, discriminative and ensemble learning on multi-modal perceptual fusion toward news video story segmentation. In *IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan, 2004.

[131] Z. Xiong, R. Radhakrishnan, A. Divakaranz, and T.S. Huang. Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework. In *IEEE International Conference on Multimedia and Expo*, volume 3, pages 401–404, 2003.

[132] W. Hsu, S.F. Chang, C.W Huang, L. Kennedy, C.Y. Lin, and G. Iyengar. Discovery and fusion of salient multi-modal features towards news story segmentation. In *SPIE Electronic Imaging*, 2004.

[133] P. Aigrain, P. Joly, and V. Longueville. Medium knowledge-based macro-segmentation of video into sequences. In *Proceedings IJCAI Workshop on Intelligent Multimedia Information Retrieval*, Montreal, 1995.

[134] A. G. Hauptmann and M. A. Smith. Text, speech and vision for video segmentation: the informedia project. In *AAAI Fall Symposium Computational Models for Integrating Language and Vision*, Boston, 1995.

[135] J. Nam and A. H. Tewfik. Combined audio and visual streams analysis for video sequence segmentation. In *International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 2665–2668, 1997.

[136] C. Saraceno and R. Leonardi. Identification of story units in audio-visual sequences by joint audio and video processing. In *International Conference on Image Processing*, pages 363–367, 1998.

[137] L. Zhao, W. Qi, S.-Q. Yang, and H. J. Zhang. Video shot grouping using best-first model merging. In *Storage and Retrieval for Media Databases*, pages 262–269, 2001.

[138] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

[139] M. G. Brown, J. T. Foote, G. J. F. Jones, K. Spärck-Jones, and S. J. Young. Automatic content based retrieval of broadcast news. In *ACM Multimedia*, pages 35–42, 1995.

[140] M. A. Hearst and C. Plaunt. Subtopic structuring for full-length document access. In *Proceedings ACM SIGIR-93 International Conference on Research and Development in Information Retrieval*, pages 59–68, 1993.

[141] R. L. Lagendijk A. Hanjalic and J. Biemond. Automated high-level movie segmentation for advanced video-retrieval systems. *IEEE transactions on circuits and systems for video technology*, 9(4), 1999.

[142] E. Veneau, R. Ronfard, and P. Bouthemy. From video shot clustering to sequence segmentation. In *International Conference on Pattern Recognition*, 2000. Barcelona, Spain.

[143] J. M. Corridoni and A. Del Bimbo. Structured representation and automatic indexing of movie information content. *Pattern recognition*, 31(12):2027–2045, 1998.

[144] J. R. Kender and B. L. Yeo. Temporal video segmentation using unsupervised clustering and semantic object tracking. In *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition*, pages 367–373, 1998.

[145] D. Beeferman, A. Berger, and J. Lafferty. Statisticals models for text segmentation. In *Machine Learning 34 (special issue on Natural Language Learning)*, pages 177–210, 1999.

[146] A. Berger, S. Della Pietra, and V. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[147] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[148] L. G. Valiant. A theory of the learnable. In *Communications of the ACM 27*, pages 1132–1144, 1984.

[149] R. E. Shapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

[150] J. Friedman, T. Hastie, and R. J. Tibshirani. Additive logistic regression: a statistical view of boosting. 28:337–407, 2000.

[151] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[152] M. A. Fischler. The representation and matching of picturial pictures. *IEEE Trans. Computers*, 22(4):67–92, 1973.

[153] Y. Yakimovsky and J. A. Feldman. A semantics-based decision theory region analyser. In *Proceedings Third Joint Conference on Artificial Intelligence*, pages 580–588, Stanford, California, 1973.

[154] T. M. Strat. *Natural Object Recognition*. Springer Verlag, 1992.

[155] M. R. Naphade. *A probabilistic framework for mapping audio-visual features to high-levels semantics in term of concepts and context*. PhD thesis, University of Illinois at Urbana-Champaign, 2001.

[156] C.-Y. Lin, L. Tseng, and J.R. Smith. Video collaborative annotation forum: Establishing ground-truth labels on large multimedia datasets. In *NIST TREC-2003 Video Retrieval Evaluation Conference*, 2003.

[157] Noam Slonim. Iba 1.0: Matlab code for information bottleneck clustering algorithms. http://www.cs.huji.ac.il/ noam, 2003.

[158] B. Janvier, E. Bruno, S. Marchand-Maillet, and T. Pun. A contextual model for semantic video structuring. In *Proceedings of the 13th European Signal Processing Conference EUSIPCO'05*, 2005.

[159] X. Feng, K.I. Williams, and S.N. Felderhof. Combining belief networks and neural networks for scene segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:467–483, 2002.

[160] Y. D. Rubinstein and T. Hastie. Discriminative vs informative learning. In *Proceedings Third International Conference on Knowledge Discovery and Data Mining*, pages 49–53, 1997.

[161] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, 2002.

[162] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields : probabilistic models for segmenting and labeling sequence data. In *Proceedings 18th International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[163] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 1–8, 2002.

[164] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 134–141, Edmonton, Canada, 2003.

[165] D. Pinto, A. McCallum, X. Wei, and W.B. Croft. Table extraction using conditional random fields. In *Proceedings of the 26th ACM SIGIR*, pages 235–242, Toronto, Canada, 2003.

[166] R. Shapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceeding of the Eleventh Annual Conference on Computational Learning Theory*, volume 37, pages 297–336, 1999.

[167] W. Hsu, L. Kennedy, S-F. Chang, M. Franz, J. Smith, and G. Iyengar. Adaptive feature discovery for trecvid broadcast news video story segmentation. In *TrecVid Workshop*, Gaithersburg, 2004.

[168] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.