UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
MOHAMED AMINE BOURENANE

UN OUTIL POUR L'INDEXATION DES VIDÉOS PERSONNELLES PAR LE CONTENU

AOÛT 2009

Sommaire

Suite aux avancées technologiques réalisées récemment, il y a eu une explosion dans la quantité de séquences vidéo disponibles et leur accessibilité. Ceci a été grandement motivé par la baisse des prix des dispositifs d'acquisitions tels que les caméras numériques, l'augmentation de la capacité des supports de stockage tels que les disques durs, et le développement des technologies permettant le partage de vidéos telles que l'Internet et la téléphonie mobile. Cette abondance de données n'a cependant pas que des impacts positifs. En effet, sans techniques appropriées de stockage, de recherche et d'extraction, toutes ces vidéos seront difficilement exploitables. Les systèmes d'indexation et de recherche des vidéos se présentent donc comme la solution à ce problème. Certains de ces systèmes exploitent le texte qui entoure les vidéos afin de les indexer. Cependant, comme la plupart des séquences existantes ne sont pas annotés avec du texte, le seul recours qui reste serait de se baser sur leur contenu. C'est dans cette perspective que s'inscrit notre recherche, i.e. le développement d'un système pour la recherche des séquences vidéo, basé sur leur contenu.

Il serait très prétentieux de notre part de vouloir développer un système qui indexe tous les types de séquences vidéo. Nous avons donc décidé de concentrer nos efforts sur le développement d'un système qui indexe les vidéos personnelles. Notre choix a été motivé par 1) la quantité phénoménale de vidéos personnelles disponibles aujourd'hui, et qui ne cesse de croître, 2) le fait que les systèmes existants se sont surtout intéressés à d'autres types de vidéos et d'applications telles que le résumé automatique des bulletins d'information ou encore l'extraction des faits saillants des matchs de soccer, tennis, etc.

Dans ce mémoire, nous allons commencer par examiner la nécessité de développer des systèmes pour l'indexation des vidéos personnelles et étudier les questions auxquelles doit répondre tout système, telles que le découpage des vidéos, l'extraction des caractéristiques, la mesure de la similarité, la formulation de la requête et la présentation des résultats. Cette étude va nous démontrer que le secret de la réussite de tout système est le développement de caractéristiques capables de représenter le contenu des vidéos avec fidélité. Forts de cette constatation, nous allons consacrer le second chapitre à une étude détaillée sur les caractéristiques. En particulier, on étudiera les caractéristiques de la couleur telles que les moments et les histogrammes, celles de la texture telles que la matrice de cooccurrence, celles de la forme telles que le pourcentage des points de contour, et celles du mouvement. Cette étude va nous permettre de déterminer parmi les caractéristiques existantes celles qui donnent les meilleurs résultats lors de la recherche. Elle va nous permettre également d'appliquer une nouvelle caractéristique à l'indexation de la vidéo, en l'occurrence l'histogramme de la couleur aux alentours des points de contour. Finalement, elle va nous permettre de mesurer à quel point il est important de bien choisir l'image clé (celle qui sera utilisée pour représenter la séquence vidéo) et de développer une nouvelle façon de l'extraire.

Lors du troisième chapitre, nous allons décrire en détail l'architecture et le fonctionnement du système de recherche que nous avons développé. Nous aborderons des questions telles que les caractéristiques adoptées, les mesures de similarité utilisées, la sélection des caractéristiques, la formulation de la requête et la présentation des résultats.
Les expériences conduites dans le dernier chapitre démontreront l'efficacité de notre système et la pertinence des caractéristiques que nous avons adoptées.

Table des matières

Liste des figures

Liste des abréviations

PAL	Phase Alternate Line.
SECAM	Sequential Color with Memory.
NTSC	National Television System(s) Committee.
MPEG	Moving Picture Experts Group.
BD	Base de donnèes.
RGB	Red, Green, Blue.
HSV	Hue, Saturation, Value.
CMY	Cyan, Magenta, Yellow.
CIELab	Espace de couleur ,
CIE	Commission Internationale d'Éclairage.
EMD	Earth mover distance.
GLCM	Gray level Coocurence Matrice.
TREC	Text retrieval conference.
OCR	Optical Character Recognition.

Remerciements

Toute la grâce est pour Dieu.

C'est avec un grand plaisir que j'exprime ma profonde gratitude au professeur Mohamed Lamine Kherfi, dont le soutien sans faille et les précieux conseils ont joué un rôle crucial lors de mon travail scientifique. Je suis persuadé que sans son aide, ce mémoire n'aurait jamais vu le jour.

J'exprime aussi ma reconnaissance aux professeurs Mhamed Mesfioui et François Meunier qui ont fait l'honneur de juger mon travail.

Je remercie tous les membres du département de mathématiques et d'informatique et spécialement madame Chantal Guimond.

Chapitre 1

Généralités et état de l'art

1.1 Introduction

Les avancées récentes en technologies de multimédia permettent la capture et le stockage des vidéos avec des ordinateurs relativement peu coûteux. En outre, les nouvelles possibilités offertes par Internet ont rendu une grande quantité de vidéos publiquement disponibles. Cependant, sans techniques appropriées de stockage, de recherche et d'extraction, toutes ces vidéos sont difficilement exploitables. Les utilisateurs veulent pouvoir faire une recherche à l'intérieur de la vidéo, au lieu de la visionner entièrement pour trouver l'information qu'ils recherchent. Par exemple, un utilisateur analysant une vidéo d'un documentaire recherchera des événements spécifiques en se basant sur des caractéristiques visuelles, textuelles, sonores ou sémantiques. Ainsi, la recherche de la vidéo basée sur le contenu devient un problème important à résoudre, et par conséquent, le besoin d'outils qui peuvent manipuler le contenu visuel des vidéos. Ce chapitre présente une synthèse des travaux existants le plus souvent cités dans la littérature. Il examine alors les divers problèmes liés à la conception et la mise en œuvre d'un système de recherche de la vidéo basée sur le contenu, tel que la segmentation, l'extraction des caractéristiques et la formulation de la requête. Les systèmes de recherche de la vidéo basés sur le contenu apparaissent comme une prolongation logique des systèmes de recherche d'images ou d'audio basés sur le contenu. Cependant, il y a un certain nombre de facteurs spécifiques à la vidéo et qui doivent être pris en considération. Ces facteurs sont principalement liés à l'information temporelle présente dans la vidéo. L'information temporelle induit le concept de mouvement des objets présents à l'intérieur de la vidéo. Un autre problème associé à la

recherche de la vidéo est la complexité des systèmes de requête. La recherche par l'exemple typiquement utilisée dans les systèmes de recherche d'images basée sur le contenu, exige de l'utilisateur de montrer au système un ou plusieurs documents semblables à ce qu'il recherche. Bien que ceci semble normal pour des images, ce processus de requête est plus compliqué à adapter dans le contexte des documents vidéo.

En ce qui concerne les systèmes existants qui permettent la recherche de la vidéo par le contenu, ils se sont intéressés à des vidéos spécialisées telles que les vidéos de sport et les bulletins d'informations. Cependant, à notre connaissance, il n'y a pas de travaux qui se sont intéressés aux vidéos personnelles, bien que ce type de vidéos soit omni présent et leur nombre est en pleine expansion. Par conséquent, développer un système qui s'occupe de ce genre de vidéos est devenu crucial. Nous nous sommes donc attaqués à cette problématique en développant un outil qui permet à la fois d'organiser ces vidéos et de les localiser suite à une requête formulée par l'utilisateur.

Dans ce chapitre, nous allons aborder la nécessité d'avoir des systèmes de recherche pour la vidéo, puis dans la section 1.3 nous allons donner quelques exemples de systèmes de recherche existants, ensuite dans la section 1.4 nous allons présenter les questions relatives à la recherche de la vidéo telles que la segmentation de la vidéo, l'extraction de l'image clé et des caractéristiques, et la formulation de la requête. Finalement, nous présentons une conclusion du chapitre.

1.2 La nécessité d'avoir des systèmes de recherche pour la vidéo

Les systèmes de recherche de la vidéo basée sur le contenu visent à aider l'utilisateur à retrouver une séquence vidéo dans une base de données potentiellement grande. Trois cas principaux peuvent être distingués :

- L'utilisateur a en tête une séquence vidéo spécifique et sait qu'elle existe dans la base de données. L'utilisateur pourra décrire avec précision la séquence cible et peut voir dès le premier regard si une séquence suggérée correspond à son besoin.
- L'utilisateur a une vidéo spécifique en tête et ne sait pas si elle existe dans la base de données. Le problème ici est de fournir un outil précis de recherche de sorte que l'utilisateur puisse rapidement prendre une décision si la vidéo cible est dans la BD ou non.
- L'utilisateur recherche une vidéo simplement en se rapportant à son thème ou à un certain événement qui se produit dans la séquence vidéo, ex. tous les buts marqués pendant un match de football.

1.2.1 Pourquoi développer des systèmes de recherche de la vidéo

La vidéo améliore significativement l'expérience de communication et de formation. Quand elle est pertinemment liée aux textes, aux diagrammes et aux images, la vidéo fournit le réalisme, l'intérêt et les détails non disponibles dans d'autres types de médias. Les attributs de la vidéo sont critiques dans beaucoup de secteurs incluant la formation médicale, la maintenance technique, les démonstrations de produit et divers types de formation à distance. Deux goulots d'étranglement empêchent la vidéo d'être une partie intégrante de l'expérience informatique d'aujourd'hui : le coût et le temps pour indexer la vidéo, et la difficulté de la recherche et l'accès aux contenus de la vidéo.

Jusqu'à maintenant, pour le grand public, les usages liés aux photos et aux vidéos numériques restent proches de ceux liés à la photo papier et à la vidéo analogique. Or, le gain d'accessibilité offert par le support numérique n'est réellement intéressant que si nous sommes capables d'appliquer la recherche d'informations à ses contenus. A titre d'exemple, au moment de la réalisation d'un nouveau documentaire, nous avons souvent besoin d'y inclure des passages extraits d'émissions existantes. Cependant, retrouver ces passages est une tâche fastidieuse et longue qui requiert parfois plusieurs jours de

visualisation séquentielle. Un moyen de réaliser cette recherche sur les contenus est de créer un index pour accéder rapidement à ces vidéos. Par exemple, lorsque nous cherchons une information précise dans un livre, sans connaitre à priori le numéro de la page, nous pouvons nous référer à la table des matières, ou bien à l'index des mots-clés. Pour un document vidéo, l'idée est la même, mais au lieu de renvoyer l'utilisateur à un numéro de page, nous pouvons être renvoyés à un numéro d'image, à un numéro de plan ou à des coordonnées dans une image donnée, etc. De plus, la recherche ne s'effectue pas obligatoirement à partir d'un mot-clé, mais peut être basée sur une autre image, une couleur, une caractéristique sémantique, etc.

1.2.2 Les applications des systèmes de recherche de la vidéo

Les utilisations potentielles des systèmes de recherche de la vidéo basée sur le contenu incluent :

- La télédiffusion

  En général, les chaînes de télévision gardent un archive complet de leur diffusion et avec le temps cet archive devient gigantesque. Les journalistes utilisent ce dernier pour préparer de nouveaux documentaires ou émissions en cherchant dans les archives les anciens reportages liés au thème voulu. Un système de recherche de la vidéo facilitera énormément leur tâche.

- La sécurité

  La vidéosurveillance est devenue un outil important utilisé pour assurer la sécurité dans les villes, aéroports, métros, etc. Après un acte criminel, les services de sécurité ont souvent besoin de visionner les séquences de vidéosurveillance pour pouvoir identifier les suspects. La rapidité de l'extraction des données est un atout majeur dans ce cas précis.

- La formation à distance

  De nos jours, les avancées dans les télécommunications et les technologies multimédias ont permis la formation multimédia aux étudiants éloignés, via Internet. Par conséquent, la vidéo est largement utilisée dans la formation à distance. Puisque chaque vidéo peut couvrir beaucoup de sujets, c'est critique pour un environnement de formation à distance d'avoir les capacités de recherche dans une vidéo basée sur son contenu pour répondre aux besoins de divers individus.

- L'archivage de la vidéo

  Ces dernières années, il y a eu l'apparition d'un nombre grandissant de sites Internet qui offrent le partage de la vidéo en ligne, par exemple le site YouTube. Les utilisateurs de ces sites ont besoin d'un outil pour pouvoir localiser les vidéos voulus en un temps acceptable.

- Les applications médicales

  La vidéo dans le domaine médical est essentiellement utilisée pour des usages pédagogiques, par exemple, pour enregistrer les opérations cliniques, pour présenter les détails des maladies, leurs symptômes, les comparaisons et les chirurgies. Un outil de recherche de la vidéo permettrait à ceux qui enseignent la médecine d'économiser beaucoup de temps, et aidera à perfectionner ceux qui sont dans les endroits isolés.

1.3 Exemples de systèmes de recherche de la vidéo

Dans cette section, nous allons donner quelques détails sur les systèmes ou prototypes existants destinés à la recherche de la vidéo.

1.3.1 La bibliothèque de vidéo numérique Informedia

Le développement du projet Informedia a commencé en 1994, à l'université de Carnegie Mellon à Pittsburgh aux États-Unis. Le but du projet été l'extraction du contenu visuel et audio de la vidéo pour permettre la recherche par le contenu. Ce projet a passé par plusieurs phases de développement. Dans la première phase, les chercheurs ont combiné l'extraction des caractéristiques visuelles et la compréhension du langage naturel pour la transcription automatique, la segmentation et l'indexation de la vidéo [1]. Dans la deuxième phase, ils ont amélioré l'extraction des caractéristiques visuelles et sonores pour faire le résumé, la visualisation et la présentation de la vidéo [2, 3,4].
En 2004, les développeurs de ce système ont participé aux tâches de l'extraction des caractéristiques sémantiques et à la recherche manuelle, interactive et automatique de la compétition TRECVID [5]. Notons que les tests de ce système se sont concentrés surtout sur les vidéos de bulletins d'informations.

1.3.2 Le système de vidéo numérique Fischlar

Ce système a été développé au centre de traitement de vidéo numérique de l'université de Dublin en Irlande [6]. Ses développeurs se sont intéressés principalement au développement de techniques de navigation basées sur le contenu dans les bases de données de vidéo de bulletins d'informations. Cela comprend la recherche, la navigation, le filtrage, la lecture et le résumé de la vidéo. Dans TRECVID 2004 [7], les développeurs de ce système ont participé à la tâche de recherche interactive des vidéos. Pour ce faire, ils ont inclus dans leur système les techniques de reconnaissance optique de caractères (OCR), la reconnaissance de la parole, la détection de mouvement et la reconnaissance des visages.

1.3.3 Le système iBase

Ce système a été développé au Collège Imperial de Londres [8]. L'objectif global du projet est de faciliter la navigation, la recherche et la visualisation des vidéos d'une base de données de bulletins d'informations. Pour ce faire, les développeurs de ce système ont extrait des caractéristiques de couleur, de texture et de texte pour faire de la recherche par l'exemple. Dans TRECVID 2004 et 2005 [9, 10], les développeurs ont fait des tests pour la détection des plans, l'extraction des caractéristiques de haut niveau et la recherche de la vidéo. Ils ont détecté les plans en utilisant l'histogramme de la couleur, ensuite ils ont extrait des caractéristiques de couleur, de texture et de texte pour faire de la recherche.

Nous avons présenté très brièvement quelques systèmes ou projets qui traitent le domaine de la recherche de la vidéo par le contenu. Pour avoir des informations sur d'autres systèmes existants, voir [11, 12, 13, 14].
Dans la section suivante, nous allons voir les problèmes liés à la recherche de la vidéo par le contenu.

1.4 Les questions relatives à la recherche de la vidéo

Nous avons vu dans la section précédente qu'il y a eu ces dernières années quelques tentatives pour réaliser des outils pour la recherche de la vidéo; mais quels sont les principaux problèmes qu'il faut résoudre pour développer de tels systèmes?
Pour faire de la recherche dans une BD de vidéos, il faut en premier lieu indexer son contenu. Pour ce faire, il faut extraire, représenter et organiser efficacement le contenu de ces vidéos. La figure 1.1 donne une vue d'ensemble du processus de prétraitement des vidéos afin de les indexer. Ce prétraitement inclut le découpage des vidéos, l'annotation manuelle et l'extraction des caractéristiques de bas et de haut niveau. Les caractéristiques de bas niveau sont celles de la couleur, de la texture, des contours, de la

forme, du mouvement, du texte et de l'audio. Les caractéristiques de haut niveau sont les concepts sémantiques présents dans la vidéo.

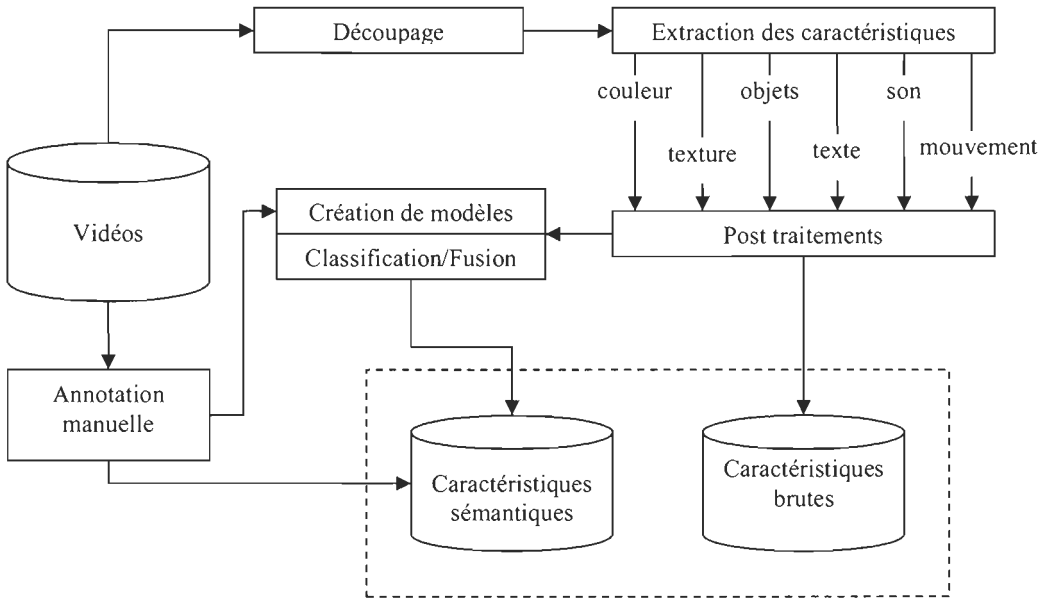


Figure 1.1 : Phase de prétraitement des vidéos.

Selon la figure 1.1, la première étape consiste à segmenter (découper) les vidéos afin de les rendre plus faciles à analyser. La deuxième étape consiste à extraire les caractéristiques de ces vidéos découpés comme la couleur, la texture, le mouvement, la parole, le texte, etc. Les caractéristiques extraites peuvent être sauvegardées directement dans la BD comme des caractéristiques brutes de la vidéo analysée, ou elles peuvent subir plus de traitements pour extraire les informations sémantiques qui seront sauvegardées à leur tour dans la BD. Les vidéos peuvent également être annotées de façon manuelle. Cette annotation servira à la fois à la création des modèles et à l'extraction des caractéristiques sémantiques.

Une fois le prétraitement terminé (figure 1.1), nous pouvons entamer le processus de recherche. Ce processus est illustré dans la figure 1.2.

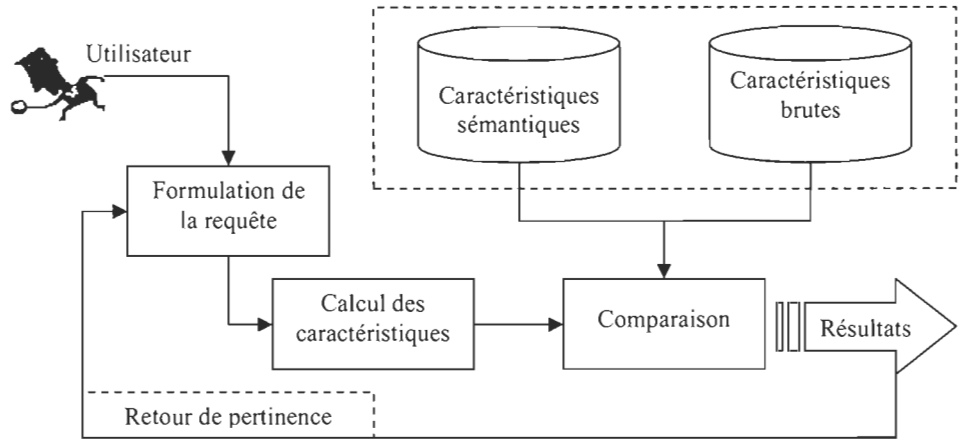


Figure 1.2 : Phase de la recherche.

En premier lieu, l'utilisateur doit formuler une requête qui peut inclure différents types de données (l'image, la vidéo, le son et le texte). Le système analyse cette requête pour calculer les caractéristiques nécessaires, puis fait une comparaison avec les vidéos existantes dans la BD. Finalement, le système retourne à l'utilisateur les vidéos qui ressemblent le plus à sa requête.

Dans ce qui suit, nous allons donner plus de détails sur les étapes du prétraitement et la recherche de la vidéo.

1.4.1 Découpage de la vidéo

Une vidéo est physiquement constituée de plans et sémantiquement de scènes. Nous pouvons définir un plan comme une suite continue d'images enregistrées au cours d'une même prise, alors qu'une scène est une partie d'un acte déterminée par l'entrée ou la sortie d'un ou de plusieurs acteurs [15]. Pour rendre l'analyse et l'indexation de la vidéo plus facile, cette dernière doit être découpée en unités logiques telles que des plans et des scènes. En général, chaque plan est l'effet d'un mouvement de caméra, un mouvement d'objets ou un effet d'édition du film. Puisque les images dans un plan sont

fortement corrélées le long de la dimension temporelle, très peu d'images représentatives ou images clés sont choisies pour récapituler chaque plan. Cependant, avec cette approche, une vidéo de longueur typique sera représentée par des milliers d'images clés. Par conséquent, les chercheurs ont proposé des méthodes pour identifier les images sémantiquement importantes et modéliser les rapports inter plans. Comme illustré dans la figure 1.3, le partitionnement d'une vidéo peut se faire à 4 niveaux différents de granularité :

- Niveau-Image : chaque image est traitée séparément. Aucune analyse temporelle à ce niveau.
- Niveau-Plan : un plan est un ensemble d'images contiguës, toutes acquises par un enregistrement continu de la caméra.
- Niveau-Scène : une scène est un ensemble de plans contigus ayant une signification sémantique commune.
- Niveau-Vidéo : la vidéo est traitée comme un seul ensemble.

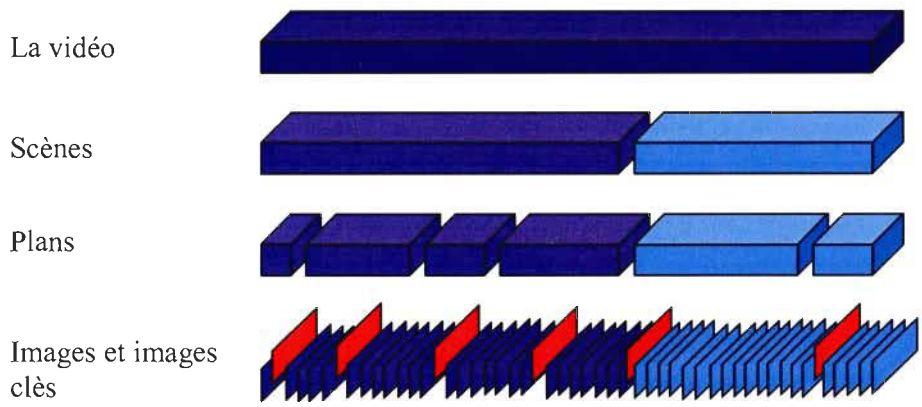


Figure 1.3 : Structure hiérarchique dans une vidéo.

Pendant le montage d'un document vidéo, le réalisateur inclut des transitions entre les différents plans et scènes. Dans ce qui suit, nous allons éclaircir le concept de transition et étudier les différentes techniques pour les détecter afin de découper les vidéos en plans.

1.4.1.1 Transition entre les plans

Les transitions entre les plans sont les moments de passage entre les plans de la caméra et qui mènent le lecteur de la vidéo à partir d'un plan à l'autre. Ils sont ajoutés pendant la postproduction. Il y a deux types de transitions que nous pouvons avoir entre les plans :

- les transitions de plan brusque (discontinu), également désignées sous le nom de coupure;
- les transitions de plan progressif (continu), qui peuvent être du type estompé (*fades*), de dissolution (*dissolve*), de balayage (*wipe*) ou de poussé (*push*).

Ces transitions sont définies comme suit :

a. Coupure (*shot cut*): un changement brusque d'un plan à un autre (Figure 1.4-a) ;
b. Apparaître en fondu (*fade-in*) : le plan apparaît graduellement à partir de sa première image (Figure 1.4-b) ;
c. Disparaître en fondu (*fade-out*) : le plan disparaît graduellement (Figure 1.4-c) ;
d. Dissolution (*dissolve*) : le plan courant disparaît graduellement tandis que le prochain plan apparaît graduellement (Figure 1.4-d) ;
e. Balayage (*wipe*) : le prochain plan est indiqué par une frontière mobile sous forme d'une ligne ou d'un motif (Figure 1.4-e) ;
f. Poussée (*push*) : le prochain plan est introduit en balayant l'écran de façon horizontale (Figure 1.4-f).

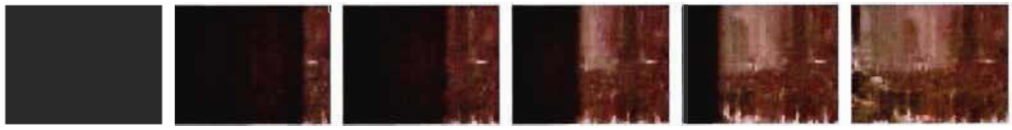
La détection de l'une de ces transitions pendant l'analyse de la vidéo permet le découpage en différents plans. Dans ce qui suit, nous allons étudier les différentes techniques existantes pour la détection de coupure entre les plans.

(a) L'effet coupure (*shot cut*)

(b) L'effet apparaître en fondu (*fade-in*)

(c) L'effet disparaître en fondu (*fade-out*)

(d) L'effet dissolution (*dissolve*)



(e) L'effet essuyer (*wipe*)

(f) L'effet pousser vers la gauche (*A push to the left*)

Figure 1.4 : Différents exemples de transition de plan.

1.4.1.1.1 La détection de coupure entre les plans

Cette section résume les approches existantes pour la détection de coupure entre les plans. Au moment de la coupure, une image est remplacée par une autre. Par conséquent, la détection doit produire un certain nombre de plans tels que :

- toutes les images dans le même plan exposent des caractéristiques semblables;
- les images appartenant à différents plans ont des caractéristiques différentes.

Différentes caractéristiques et métriques ont été proposées pour la détection de coupure entre les plans. Elles ont été analysées dans plusieurs études comparatives [16, 17, 18, 19]. Dans ce qui suit, nous allons présenter les différentes approches utilisées pour la détection de coupure entre les plans.

a. Comparaisons au niveau pixel

   La manière la plus simple de mesurer la différence entre deux images est de comparer les intensités des pixels entre les deux images [20]. Selon cette méthode, il y a une coupure entre deux images qui se succèdent dans une vidéo, si le changement de la moyenne des intensités des pixels est supérieur à un seuil fixé. L'inconvénient de cette méthode est sa sensibilité aux mouvements des objets et de la caméra, car en utilisant le changement de la moyenne, il est impossible de faire la différence entre un grand changement dans une petite région de l'image et un petit changement dans une grande région. Zhang et al. [21] ont proposé une amélioration qui consiste à déterminer le pourcentage des pixels qui ont changé considérablement entre deux images. Leur méthode utilise un filtre moyen 3x3 pour réduire le bruit et l'effet du mouvement de la caméra. Bien qu'elle apporte une amélioration, cette méthode est toujours sensible au mouvement des objets et de la caméra.

b. Comparaisons au niveau global

Quelques méthodes ont été proposées pour pallier au problème du mouvement de la caméra et des objets. Ces méthodes comparent les caractéristiques globales de chaque image au lieu de comparer chaque pixel individuellement. Nagasaka et Tanaka [22] ont proposé l'utilisation de l'histogramme à niveau de gris pour comparer deux images. Toutefois, la méthode n'était pas robuste en présence de bruit momentané, comme le flashe d'un appareil photo ou le mouvement d'un grand objet. Nagasaka et Tanaka [23] ont également proposé une méthode basée sur la comparaison de l'histogramme de la couleur. Ils ont proposé d'utiliser un code couleurs de 6 bits obtenus en prenant les deux bits les plus significatifs de chaque composante RGB ce qui donne un code à 64 couleurs. Ils utilisent la loi de Chi deux χ_2 pour mesurer la différence entre deux distributions liées. Selon Gargi et al. [24], Nagasaka et Tanaka [22] et Lienhart [25], une simple comparaison entre les histogrammes de la couleur (RGB ou YUV), avec chaque bande quantifiée à 2^b valeurs différentes, est une méthode efficace pour détecter les frontières des plans.

c. Comparaisons basées sur les blocs

Une faiblesse des comparaisons de niveau global est qu'elles peuvent manquer la coupure entre deux plans où seule la distribution spatiale du contenu change. Zhang et al. [22] proposent de partitionner l'image en régions (blocs) puis faire la comparaison des blocs correspondants dans deux images successives. Les blocs sont comparés sur la base des caractéristiques statistiques du deuxième ordre de leurs valeurs d'intensité. Nagasaka et Tanaka [22] proposent également de diviser chaque image en quatre régions et de comparer les histogrammes de couleur des régions correspondantes. Ueda et al. [26] proposent une autre approche en augmentant le nombre de blocs à 48 et en déterminant la mesure de différence entre deux images comme le nombre total de blocs avec une différence d'histogramme supérieur à un seuil donné. Selon Otsuj et al. [27], la

méthode d'Ueda et al. [26] est plus efficace que celle de Nagasaka et Tanaka [22]. Cependant, le fait que les blocs soient plus petits dans l'approche d'Ueda et al. [26] signifie aussi que cette méthode est plus sensible au mouvement des objets et de la caméra [28]. Cela met en évidence le problème de choisir une échelle appropriée pour la comparaison entre les contenus de deux images.

d. Approches basées sur le mouvement

Plusieurs méthodes ont tenté d'éliminer la différence entre deux images causées par le mouvement des objets et de la caméra avant de faire la comparaison. Plusieurs auteurs [17, 29, 30] ont proposé des méthodes qui font la comparaison entre les images partitionnées en blocs afin d'obtenir des mesures de similarités basées sur le mouvement. La différence principale entre ces approches est la méthode utilisée pour combiner les mesures des blocs afin d'obtenir une caractéristique globale de l'image. Vlachos [31] a utilisé une méthode qui emploie la corrélation de phase afin d'obtenir une mesure de similarité entre deux images. Cette méthode se caractérise par son invariance aux changements dans l'illumination globale du contenu de l'image. Fernando et al. [32] ont exploité le fait que les vecteurs de mouvement sont de nature aléatoire pendant une coupure de plan. La méthode calcule le vecteur de mouvement moyen entre deux images et la distance Euclidienne par rapport au vecteur moyen pour tous les vecteurs de mouvement. Ils déduisent qu'il y a une coupure s'il y a une grande augmentation dans la distance Euclidienne.

e. Approches basées sur les contours

Zabih et al. [33] ont proposé une méthode pour détecter les coupures de plan en vérifiant la distribution spatiale des contours sortants et entrants. Cette méthode a exploité le fait que les contours d'objets dans l'image avant une coupure de plan ne peuvent être trouvés dans le même emplacement après la coupure. Bien que cette méthode ait illustré la viabilité de la caractéristique de contour pour détecter

un changement de la décomposition spatiale entre deux images, sa performance était décevante comparée avec d'autres métriques plus simples qui sont moins gourmandes en ressource machine [34, 35, 36].

Nous allons maintenant donner une brève idée sur les méthodes proposées pour détecter les transitions graduelles dans une séquence vidéo.

1.4.1.1.2 La détection des transitions graduelles

Moins de travaux ont été réalisés sur la détection des transitions progressives. Contrairement aux coupures franches, la différence entre les images pendant une transition progressive est petite. Pour cette raison, il peut être difficile de distinguer entre les changements provoqués par le mouvement de la caméra et des objets et ceux provoqués par une transition progressive. Par conséquent, si on essaie de détecter toutes les transitions progressives, il en résulte beaucoup de fausses alarmes. Nous pouvons dire que la détection précise des transitions graduelles est encore un problème non résolu. Lienhart [35] a présenté un algorithme de détection d'effet fondu réalisant un taux de détection de 69 % tout en ramenant le taux de fausse alarme à 68 %.
Alors, le but de cette section est de passer en revue les travaux existants sur la détection des transitions graduelles et d'aborder les questions sous-jacentes. Des aperçus utiles sont également présentés dans certaines recherches [19, 28]. Nous pouvons mentionner trois groupes d'approches pour détecter les transitions graduelles.

a. Approches basées sur les histogrammes

   Une des premières méthodes proposées est la technique de comparaison jumelée proposée par Zhang et al. [21]. Cette méthode compare les différences d'histogrammes avec deux seuils : un seuil inférieur a été utilisé pour détecter les petites différences qui se produisent pendant la durée de la transition graduelle, tandis qu'un seuil plus haut a été utilisé pour la détection de coupure entre les

plans et les transitions graduelles. Les auteurs ont essayé de résoudre le problème des fausses alarmes par la détection de modèle de mouvement dans l'image introduit par le mouvement de la caméra. Bien que cet algorithme ait aidé à réduire le nombre de fausses alarmes, il a échoué à détecter les transitions avec des mouvements de la caméra avant, pendant ou après la transition.

b. Approches basées sur les contours

Pendant une dissolution, les contours des objets disparaissent graduellement tandis que les contours des nouveaux objets deviennent graduellement apparents. Pendant un effacement, les contours disparaissent graduellement, tandis que pendant une apparition en fondu les contours émergent graduellement. Zabih et al. [33] ont prolongé leur méthode de détection des coupures entre les plans pour détecter les transitions progressives. Ils ont rapporté que le taux de détection des transitions progressives avec cette méthode est bon, mais selon d'autres auteurs [17, 25], le taux de fausses alarmes étaient souvent inacceptables. Parmi les raisons des fausses alarmes est le fait que l'algorithme ne compense que pour les mouvements en translation; un mouvement en zoom conduit directement à une fausse alarme. Lienhart [25] a exploité la perte de contraste des contours pendant une dissolution pour détecter la transition graduelle. Pour ce faire, il a capturé et amplifié la relation entre les contours les plus forts et les plus faibles. Le but de cette méthode a été de résoudre le problème de mouvement de la caméra et des objets rencontré par la méthode de Zabih et al. [33]. Cependant, Lienhart [25] a rapporté que le taux de fausses alarmes reste toujours très haut. Une autre méthode intéressante est la détection des transitions progressives par l'analyse de tranche temporelle [37, 38]. La vidéo est représentée comme un volume 3-D qui se compose d'un ensemble de tranches 2-D. Ces tranches ont été alors utilisées pour extraire un indicateur qui peut être utilisé pour capturer la similarité entre les vidéos. Chaque tranche contient les régions de couleur et de texture

uniformes et les bordures de ces régions sont utilisées pour détecter la présence de transitions de plan.

c. Approches basées sur la variance

Une autre méthode pour détecter les transitions progressives est d'analyser le comportement temporel de la variance des intensités de pixels dans chaque image. Ceci a été proposé pour la première fois par Alattar [39]. Ensuite d'autres auteurs [40, 36, 41] ont proposé des modifications à cette méthode. Alattar [39] a exploité le fait que la courbe de variance dans un effet de dissolution idéale a une forme parabolique. Ainsi, détecter les effets de dissolution revient à détecter ce modèle dans la série de temps. Même si ces modèles ont une bonne performance, ils sont handicapés par la supposition faite à propos des transitions, car cette dernière ne se généralise pas aux séquences vidéo réelles. Pour résoudre ce problème, Nam et Tewfik [42] ont proposé une technique pour estimer la courbe de transition actuelle par l'utilisation de la technique d'adaptation de courbe polynomiale *B-Spline*. D'autres auteurs [25, 36, 41] ont proposé des approches pour la détection des transitions de type fondu.

1.4.1.2 Découpage dans le domaine compressé

Patel et Sethi [43] ont développé un algorithme de détection des coupures directement dans le domaine compressé. Leur méthode se sert des coefficients de la transformée cosinus discrète pour détecter les coupures en calculant la variance moyenne et les histogrammes des images MPEG-1. Alternativement, nous pouvons nous servir des indications fournies par la taille du fichier et les vecteurs de mouvements. Deardorff et al. [44] ont étudié la variation de la taille de fichier du mouvement JPEG pour détecter les coupures de plan. La dynamique du volume de fichier suppose que les images qui appartiennent au même plan sont régulières dans le contenu et la dynamique. Par conséquent, un changement brusque de la taille de fichier de deux images consécutives peut indiquer une coupure de plan potentielle. Deng et Manjunath [45] ont étudié la

dynamique de mouvement des images-P et des images-B. Puisqu'il y a une discontinuité du mouvement en conséquence à un changement soudain entre deux images consécutives, les auteurs prédisent qu'il y a une baisse significative des vecteurs de mouvement avant. En particulier, les images-B contiendront plus des vecteurs de mouvement en arrière. Par conséquent, les coupures peuvent être facilement détectées par le seuil des vecteurs de mouvement avant. Bien que ces approches puissent détecter les coupures de plan, elles tendent à produire de fausses alarmes dues aux transitions progressives. Shen et al. [46] suggèrent une méthode qui applique l'histogramme de distance de Hausdor et l'algorithme de fusion multi passages pour remplacer les évaluations de mouvement. En outre, ils décrivent un algorithme pour produire la carte de contour directement dans le domaine fréquentiel pour accélérer la vitesse de calcul.

Nous avons présenté une vue d'ensemble sur les différentes méthodes existantes pour le découpage des vidéos. D'après nos recherches, nous pouvons conclure qu'il reste beaucoup à faire dans le domaine de la détection des transitions graduelles. Dans la section prochaine, nous allons présenter quelques méthodes utilisées pour résumer un plan d'une vidéo.

1.4.2 Sélection des images clés

Le nombre d'images par seconde dans une vidéo prises avec une ancienne caméra mécanique varie entre 6 et 8, alors que ce nombre augmente à 120 ou plus pour une vidéo prise avec une caméra professionnelle de dernière génération. Les standards PAL et SECAM spécifient le nombre d'images par seconde à 25, alors que NTSC le spécifie à 29.97. Au fait, il suffit de 10 images par seconde pour réaliser l'illusion d'une image qui se déplace. De cela, nous pouvons conclure qu'il est inutile d'entrer dans des calculs compliqués et longs pour toutes les images d'un plan afin de correctement en extraire les caractéristiques visuelles. En effet, il serait par la suite impossible de conserver et d'utiliser cette information qui est par ailleurs redondante. Le processus de

simplification de la vidéo consiste donc en la sélection d'une ou de plusieurs images représentatives des plans. Idéalement, les images clés doivent capturer le contenu sémantique du plan. Une image clé est choisie parmi chaque plan pour l'indexation afin de ramener les problèmes de recherche de la vidéo à des problèmes de recherche d'image. Il faut noter que le choix des images clés est subjectif et il dépend souvent de l'application. Afin d'avoir une recherche efficace de la vidéo, les images choisies devraient pouvoir représenter le contenu entier de la séquence vidéo [47]. Dufaux [48] a proposé une méthode d'extraction de l'image clé basée sur la détection du visage humain dans la vidéo. Cette méthode analyse toutes les images de la vidéo puis elle sélectionne l'image clé en se basant sur la somme des visages détectés.

Il y a eu récemment beaucoup de travaux liés au problème du choix des images clés [49, 50, 51]. Deux approches principales ont été suggérées :

- avec la détection explicite des transitions de plan;
- sans la détection des transitions de plan.

Deux auteurs [52, 53] ont proposé une approche dans laquelle la première image de chaque plan est choisie comme une image clé. Ceci n'est pas toujours satisfaisant puisqu'il peut exister des changements importants dans un plan en raison du mouvement des objets ou de la caméra. Ardizzone et Cascia [54] ont suggéré que le nombre des images clés doit être en rapport avec la longueur du plan. Une fois les images clés choisies, cette approche peut suréchantillonner une vidéo dans le cas où la durée du plan est grande, mais contient peu de changements dans le contenu. Zhang et al. [55] ont suggéré d'extraire les images clés en utilisant des mesures de similarité semblables à celles utilisées dans la détection des coupures de plan. Kim et Park [56] ont proposé que la similarité entre les différentes vidéos soit évaluée par la distance de Hausdorff modifiée entre les ensembles des images clés de chaque vidéo. Chang et al. [47] ont proposé une approche pour déterminer un ensemble minimum d'images clés pour un plan, de telles manières que la distance qui sépare une image de l'image clé est inférieure à un certain seuil. Xiong et al. [57] ont suggéré une méthode compacte pour choisir les images clés, appelées recherche et diffusion (*Seek and Spread*) ou SS. Cette

méthode recherche les images clés séquentiellement et ensuite étend l'intervalle représentatif des images clés aussi loin que possible. Cependant, cette approche est plus appropriée aux plans avec des effets panoramiques. Gresle et Huang [58] ont proposé de sélectionner comme image clé l'image avec la différence temporelle minimum entre deux maxima locaux. De cette façon, les endroits où la caméra reste plus longtemps sont choisis. Vermaak et al. [59] ont proposé une approche qui localise les images qui comportent le maximum d'informations et qui sont distinctes au maximum. Wolf [60] a utilisé le flux optique pour identifier des minimums locaux de mouvement dans un plan pour identifier les images clés.

Maintenant que nous avons vu les techniques sur le découpage de la vidéo et le choix des images clés, nous allons voir très brièvement les différentes caractéristiques visuelles qui peuvent être extraites du contenu de la vidéo. Nous allons donner plus de détails sur les caractéristiques dans le deuxième chapitre.

1.4.3 Extraction des caractéristiques

Les caractéristiques sont le cœur de l'indexation et de la recherche de la vidéo. Ce sont les informations que l'on extrait de la vidéo de telle sorte qu'elles représentent d'une manière appropriée son contenu. Elles sont conservées dans un index, et utilisées pour faire la recherche. Nous pouvons classifier les caractéristiques par caractéristique de bas niveau ou brutes, et caractéristiques de haut niveau ou sémantiques selon leur complexité et utilisation des sémantiques [61].

1.4.3.1 Caractéristiques de bas niveau

Les caractéristiques de bas niveau les plus couramment utilisées pour décrire la vidéo sont la couleur, la texture, la forme et le mouvement.

1.4.3.1.1 La couleur

La couleur est une caractéristique très utilisée pour la représentation des images. Elle est invariante à la translation et à la rotation, et change légèrement en cas de changements de l'angle de prise d'image ou d'échelle. Le système de couleur le plus employé est le RGB. D'autres systèmes tels que le HSV, le L*a*b*, ou le L*u*v* sont aussi couramment utilisés. La couleur est généralement décrite par un histogramme calculé dans un des divers espaces de couleur existant. Lors de la recherche, l'histogramme de la requête est comparé aux histogrammes du reste de la base de données en utilisant une mesure de similarité comme l'intersection d'histogramme ou la distance Euclidienne. L'inconvénient majeur de l'histogramme est qu'il ne contient pas d'informations spatiales. Ainsi, pour pallier à cet inconvénient, plusieurs méthodes ont été proposées comme le découpage de l'image en zones d'intérêt par Stricker et Dimai [62] ou l'étude de la corrélation spatiale des couleurs (les correlogrammes) par Huang et al. [63].

1.4.3.1.2 La texture

La texture est une information de plus en plus utilisée en indexation d'images et de la vidéo. Elle permet de combler un vide que la couleur est incapable de faire, notamment lorsque les distributions de couleur sont très proches.

Selon Tuceryan et Jain [64], il y a quatre types d'approches d'analyse de la texture : les approches statistiques, géométriques, spectrales et par modélisation. Dans la première catégorie, nous trouvons les matrices de cooccurrence [65]. Dans la seconde, il y a les descripteurs de Tamura [66] qui caractérisent la granularité, la direction et le contraste. Le troisième type d'approches contient les ondelettes avec les filtres de Gabor qui permettent de capturer les fréquences et les directions principales [67, 68]. Finalement, la dernière catégorie inclut la décomposition de Wold [69] qui caractérise la périodicité, la direction et le désordre; ainsi que les modèles autorégressifs simultanés et multi

résolutions qui modélisent la texture à différents niveaux de granularité en fonction du voisinage des pixels [70].

1.4.3.1.3 La forme

La forme est utilisée pour caractériser les objets à l'intérieur de l'image. Généralement, la forme est décrite par des caractéristiques globales comme la taille (le périmètre et la superficie), l'excentricité et les moments ou par des caractéristiques plus précises comme les coins, les points de contours et la transformée de Fourier.
Pour décrire la forme, Hu [71] a proposé un ensemble de sept moments invariants aux translations, aux rotations et aux changements d'échelle. Une amélioration de certaines caractéristiques invariantes aux transformations linéaires a ensuite été proposée par Reiss [72]. Ainsi, de nombreuses applications de recherche d'images par le contenu utilisent l'analyse de la forme [73, 74, 74, 75].

1.4.3.1.4 Le mouvement

Le mouvement est une caractéristique spécifique à la vidéo. Il est utilisé pour caractériser les déplacements des objets à l'intérieur de la séquence d'images qui constituent le plan, et pour caractériser les mouvements de la caméra. À partir des images qui constituent un plan, le mouvement de la caméra peut être estimé, ensuite le mouvement des objets peut être déterminé.
Parmi les méthodes les plus utilisées pour caractériser le mouvement, nous trouvons la technique de calcul du flux optique. Les techniques pour calculer le flux optique sont nombreuses : les premières ont été proposées dans les années 80 [77, 78] et de nos jours, ce sujet intéresse toujours les chercheurs [79, 80, 81]. Cependant, les deux méthodes les plus couramment utilisées sont celles de Horn et Schunck [82] et de Lucas et Kanade [83]. Toutefois, la plupart des méthodes utilisent directement les vecteurs de mouvement fournis par la compression MPEG-1 ou MPEG-2.

Le mouvement est rarement utilisé directement pour l'indexation, mais plutôt pour la détection du contenu sémantique. Un des principaux exemples est celui de Leonardi et al. [84] qui traite la détection des événements sportifs.

1.4.3.2 Caractéristiques de haut niveau

Également connues sous le nom de caractéristiques logiques, dérivées ou sémantiques, les caractéristiques de haut niveau impliquent des degrés divers de sémantique représentés dans les images, la vidéo, et l'audio. Nous pouvons distinguer deux types de caractéristiques sémantiques

a. Les caractéristiques objectives

   Elles concernent l'identification des objets dans les images et l'action dans la vidéo. Un exemple de requête est « Trouvez une séquence vidéo contenant une baleine ». Pour répondre à des requêtes à ce niveau, le processus de recherche exige normalement une connaissance antérieure des objets.

b. Les caractéristiques subjectives

   Elles sont des caractéristiques abstraites. Elles décrivent la signification et le but des objets ou des scènes. Nous pouvons subdiviser les caractéristiques en événements (par exemple, le jour de l'indépendance), en types d'activité (par exemple, le dessin), la signification émotive (par exemple, un sourire), le religieux (par exemple, une prière), etc. L'interprétation complexe et le jugement subjectif peuvent être demandés à un expert dans le domaine d'application pour établir le lien entre le contenu de la vidéo et les concepts abstraits.

Généralement, afin d'extraire les caractéristiques sémantiques, nous devons en premier faire une annotation des vidéos de la BD. L'annotation d'une vidéo consiste à associer à celle-ci des informations textuelles. Ces informations, qu'on appelle aussi métadonnées, sont des mots-clés qui se rapportent essentiellement aux contenus sémantiques de la

vidéo annotée. L'annotation des vidéos se fait soit d'une façon manuelle, automatique ou semi-automatique.

a. L'annotation manuelle

Comme son nom l'indique, l'annotation se fait manuellement par un utilisateur qui est chargé d'attribuer à chaque vidéo un ensemble de mots-clés. Cependant, cette technique n'est pas appropriée pour une grande base de données, car c'est une tâche fastidieuse et difficile à faire. En plus, l'annotation faite par l'utilisateur peut être subjective, car elle dépend de son interprétation et jugement. Ainsi, cette technique demande un certain niveau d'expertise de la part de la personne annotant les vidéos.

b. L'annotation automatique

Dans cette technique, c'est un système informatique qui affecte aux vidéos les mots clés, en utilisant des techniques de classification [85, 86] ou d'intelligence artificielle [87]. Pour faire l'annotation automatique, il faut dans un premier temps, extraire les caractéristiques de bas niveau qui représentent bien le contenu de la vidéo. Ensuite, la plupart des approches utilisent l'annotation uniquement pour apprendre à faire la correspondance entre les caractéristiques de bas niveau et les concepts sémantiques. Hauptmann et al. [5] ont extrait de chaque vidéo 16 caractéristiques de bas niveau (visuelles, sonores et textuelles). Ensuite, ils ont assigné aux vidéos des caractéristiques de haut niveau (dix concepts sémantiques) suivant leurs caractéristiques de bas niveau, en utilisant une technique de classification. Il faut noter que cette méthode atteint vite ses limites, dès que le contenu de la vidéo devient très complexe (par exemple, le mouvement des objets et de la caméra).

c. L'annotation semi-automatique

Cette solution est un compromis entre les avantages et les inconvénients de l'annotation manuelle et automatique. Elle combine ces deux méthodes afin de

surmonter leurs problèmes respectifs. Ainsi, l'utilisateur doit annoter une vidéo donnée, ensuite, à l'aide d'un algorithme, le système propage les mots-clés au reste des vidéos qui ont un contenu visuel similaire. Un exemple de cette méthode est proposé par Song et al. [88].

Maintenant que nous avons présenté toutes les questions relatives à la recherche de la vidéo et une revue de la littérature, nous allons aborder dans la section suivante le problème de la formulation de la requête dans les systèmes de recherche de la vidéo par le contenu.

1.4.4 Formulation de la requête

Pour qu'un utilisateur recherche une vidéo dans une base de données, il doit en premier formuler sa requête au système. Généralement, il peut la faire soit par une requête textuelle, par une vidéo exemple ou par la navigation dans la BD de vidéos [89]. Chaque approche représente un moyen utile d'accéder à une base de données de vidéos. Nous allons présenter dans ce qui suit les avantages et les problématiques de trois approches.

- La requête textuelle

  C'est une méthode pratique pour les utilisateurs qui veulent rechercher des vidéos basée sur des concepts sémantiques. Par exemple, l'utilisateur recherche une vidéo simplement en se rapportant au concept sémantique qui se produit dans la séquence vidéo, p. ex. tous les buts marqués pendant un match de football. Cependant, les systèmes qui permettent l'extraction des caractéristiques sémantiques ne sont pas assez développés pour exploiter les avantages de cette méthode.

- La requête par l'exemple

  Cette méthode permet à un utilisateur d'indiquer sa requête via une vidéo exemple. Toutefois, la plupart des techniques existantes n'ont pas encore atteint le niveau qui permet à un utilisateur non expérimenté de préciser sa requête initiale de manière efficace quand il n'a pas de bons exemples à portée de main.

- La navigation dans la base de données

  C'est une méthode attrayante pour les utilisateurs non expérimentés qui n'ont pas de connaissance préalable du contenu de la base de données, et qui n'ont aucun concept précis de requête à l'esprit. Cependant, la navigation basée sur les concepts sémantiques n'est pas encore assez développée, en raison de l'absence de structure appropriée pour l'organisation de base de données orientée concept.

Dans la section suivante, nous allons présenter nos conclusions après ce survol des problèmes liés à la recherche de la vidéo par le contenu.

1.5 Conclusion

Ces dernières années, nous assistons à une explosion du nombre de vidéos numériques sauvegardées sur des supports de stockage personnel ou partagées dans des sites Internet. Cependant, sans techniques appropriées de stockage, de recherche et d'extraction, toutes ces vidéos sont difficilement exploitables. La solution à ce problème est le développement d'un système d'indexation et de recherche de la vidéo.

La plupart des systèmes existants se sont intéressés à des vidéos spécialisées telles que les vidéos de sport et les bulletins d'information. Cependant, à notre connaissance, il n'y a pas de travaux qui se sont intéressés aux vidéos personnelles. Nous avons donc eu l'idée de développer un outil qui permet à la fois d'organiser ces vidéos et de les localiser à la suite d'une requête formulée par l'utilisateur.

Dans ce chapitre, nous avons présenté un ensemble de problèmes liés à l'indexation et à la recherche de la vidéo et nous avons fait un survol des méthodes existantes pour les résoudre. D'après notre étude, les caractéristiques sont la base de tout système de recherche de la vidéo. En effet, le problème de recherche de la vidéo est presque toujours ramené à un problème de comparaison entre caractéristiques. Par conséquent, si l'on veut développer un système qui est précis et efficace, on doit passer par l'adoption et/ou le développement de bonnes caractéristiques. Ainsi, nous allons conduire dans le deuxième chapitre une étude détaillée des caractéristiques visuelles, ensuite nous allons utiliser ces caractéristiques dans notre système.

Chapitre 2

Les caractéristiques de la vidéo

2.1 Introduction

Afin d'indexer le contenu visuel de la vidéo, il faut extraire des caractéristiques pour le représenter. Les plus couramment utilisées sont celles de la couleur comme les moments et les histogrammes de la couleur, les caractéristiques des contours, les caractéristiques de la texture et la caractéristique du mouvement qui est spécifique à la vidéo.
Dans ce chapitre, nous allons présenter brièvement les caractéristiques les plus fréquemment utilisées en recherche de la vidéo, leurs avantages et inconvénients et les méthodes de leur extraction.

2.2 La couleur

La couleur est cet aspect de lumière visible, par lequel un être humain distingue entre différentes répartitions spectrales de l'énergie de lumière. En utilisant son système de vision, l'être humain interprète les couleurs grâce aux quantités de lumière de longueurs d'onde variées que les objets autour de lui émettent ou réfléchissent. En fait, la plupart des couleurs sont dues non à des mélanges de longueurs d'onde, mais à des soustractions. La lumière blanche du soleil étant partiellement absorbée par des pigments qui absorbent certaines longueurs d'onde et ne laissent passer que leur complément, ce qui produit la sensation de couleur. L'être humain peut différencier jusqu'à environ deux millions de couleurs différentes. La figure 2.1 montre le spectre visible pour l'œil humain.

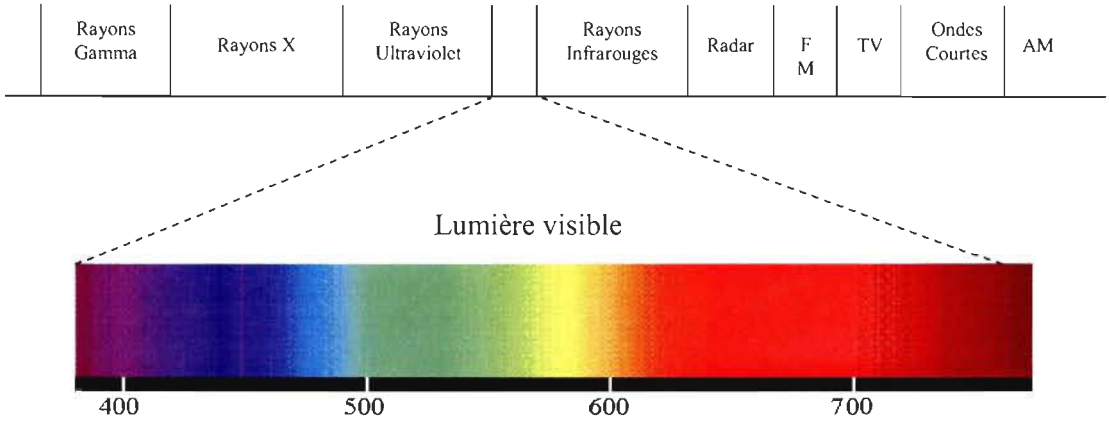


Figure 2.1: Le spectre visible.

De nos jours, presque la totalité des images et des vidéos présente dans les différents médias sont en couleur, car elles sont plus réalistes et satisfaisantes à l'œil humain. La nécessité de produire, de stocker et de transmettre des documents colorés (image ou vidéo) a conduit à imaginer des systèmes cohérents pour représenter fidèlement les couleurs qui les composent. Ces systèmes sont appelés les espaces de couleurs. Ils sont appelés ainsi, car la variation des différentes couleurs peut être représentée dans un espace tridimensionnel, où chaque point (dans cet espace) représente une couleur différente. Ce nuage de points de couleur différente constitue un espace de couleur. En conséquence, l'espace de couleur est une notation par laquelle nous pouvons spécifier les couleurs, c'est-à-dire la perception humaine du spectre électromagnétique visible.

Plusieurs espaces de couleurs ont été employés pour la représentation de couleur basée sur les concepts perceptuels. Nous pouvons citer les espaces RGB, CMY, HSV, CIELab, etc. Il n'y a aucun accord sur le meilleur espace de couleur. Cependant, ses caractéristiques désirables sont la perfection, l'uniformité, la compacité, et il doit être orienté utilisateur. La perfection signifie qu'il doit inclure toutes les couleurs différentes perceptibles. L'uniformité signifie que la proximité mesurée parmi les couleurs doit être directement rapprochée de la similitude perceptuelle ou psychologique entre ces

couleurs. La compacité signifie que chaque couleur présente une différence perceptrice des autres couleurs.

Dans ce qui suit, nous allons donner plus de détails sur les espaces de couleur RGB, HSV, CMY, CIELab et YUV.

2.2.1 L'espace RGB

C'est l'espace de couleur de base. Il est très utilisé dans les systèmes de télévision et les applications informatiques. La représentation des couleurs dans cet espace donne un cube appelé « cube de Maxwel », comme illustré dans la figure 2.2. Le système de couleur RGB est un système de couleur additif, c.-à-d. que les couleurs sont obtenues par le mélange des trois couleurs de base qui sont le rouge, le bleu et le vert. La représentation numérique la plus fréquente de cet espace de couleur est des valeurs allant de 0 à 255.

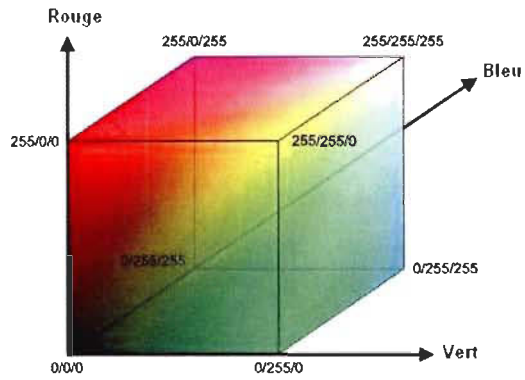


Figure 2.2: L'espace de couleur RGB.

L'avantage de l'espace de couleur RGB est qu'il est conceptuellement simple. En contrepartie, il a l'inconvénient d'être perceptuellement non uniforme, c.-à-d. il n'y a pas de corrélation entre la différence perçue entre deux couleurs différentes, et la distance Euclidienne qui sépare ces deux couleurs. De plus, l'espace de couleur RGB ne

tient pas compte des particularités de la perception visuelle des couleurs, il n'est pas indépendant du matériel utilisé, et il n'est pas très intuitif pour les utilisateurs non initiés.

2.2.2 L'espace HSV

C'est un espace dérivé de l'espace RGB, le plus souvent, utilisé dans des applications informatiques de graphisme. Il a été formellement décrit en 1978 par Alvy Ray Smith [90]. Les couleurs dans cet espace sont représentées selon des notions de teinte (*Hue*), de pureté (*Saturation*) et de luminosité (*Value*). La teinte caractérise la couleur en elle-même. Sa valeur varie entre 0 et 360° et elle correspond à la position de la couleur dans un cercle où se trouvent six couleurs primitives : rouge, jaune, vert, cyan, bleu et magenta et les différentes couleurs intermédiaires. La saturation caractérise la pureté de la couleur. Elle indique la quantité de gris dans la couleur. Sa valeur varie entre 0 et 100 %. La luminosité correspond à la brillance perçue de la couleur. Elle varie entre 0 et 100 %. La figure 2.3 montre une illustration de l'espace couleur HSV.

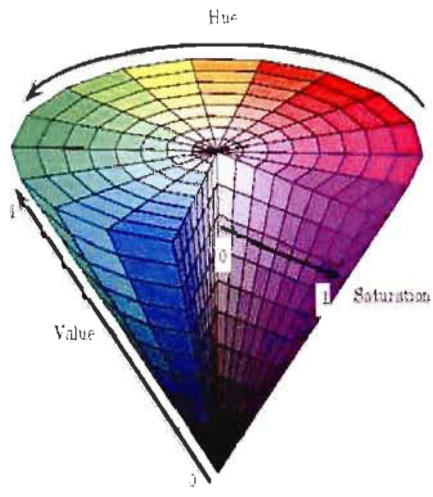


Figure 2.3 : L'espace de couleur HSV.

Les valeurs des composantes HSV sont obtenues à partir d'une transformation non linéaire des couleurs RGB, selon les relations mathématiques suivantes [91] :

$$H = \begin{Bmatrix} \theta & si & B \le G \\ 360^\circ - \theta & si & B > G \end{Bmatrix} \quad \text{où} \qquad \theta = \cos^{-1}\left(\frac{0.5(R-G)+(R-B)}{\sqrt{(R-G)^2+(R-B)(G-B)}} \right)$$

$$S = 1 - \left(\frac{3}{R+G+B} \right) \min(R,G,B)$$

$$V = \frac{1}{3}(R+G+B)$$

L'avantage de l'espace HSV est qu'il est très intuitif. Il est basé sur une approche de la couleur plus naturelle à la perception humaine. En contrepartie, il hérite des inconvénients de l'espace RGB vis-à-vis de sa dépendance du matériel utilisé et sa non-linéarité. De plus, il est difficile à faire des opérations arithmétiques dans cet espace de couleur, car habituellement il y a une discontinuité de la teinte aux alentours de 360°.

2.2.3 L'espace CMY

L'espace de couleur CMY est directement déduit de l'espace RGB. Il est souvent utilisé par les imprimantes couleur. Les composantes de cet espace sont C pour Cyan, M pour Magenta et Y pour Jaune (*Yellow*). CMY est basé sur l'absorption des couleurs, et c'est la raison pour laquelle il est considéré comme un système soustractif. La figure 2.4 montre une illustration de l'espace de couleur CMY.

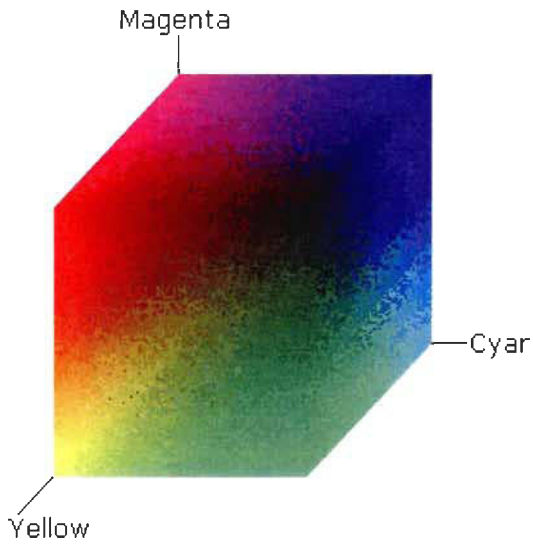


Figure 2.4 : L'espace de couleur CMY.

On peut considérer que cet espace est l'inverse de RGB vu que les formules de transformation sont les suivantes :

$$\begin{cases} C = 1 - R \\ M = 1 - G \\ Y = 1 - B \end{cases}$$

L'un des inconvénients de ce modèle est le fait qu'il ne permet pas de générer la couleur noire de façon exacte. Pour pallier à ce problème, on a proposé l'extension CMYK où la composante K code le noir (*Black*). Un autre inconvénient résulte du fait que l'écran et l'imprimante utilisent deux modèles différents pour la représentation de la couleur, en l'occurrence RGB et CMY. De ce fait, l'imprimante ne reproduit pas fidèlement les couleurs de l'écran.

2.2.4 L'espace CIELab

L'espace Lab, appelé également CIELab, a été introduit par la Commission Internationale d'Éclairage (CIE) en 1976. Sa propriété principale est son uniformité comparée à l'espace RGB. En effet, deux couleurs perceptuellement semblables peuvent

être loin l'une de l'autre dans l'espace RGB et vice versa. Contrairement à cela, dans l'espace CIELab, deux couleurs perceptuellement semblables sont toujours proches alors que deux couleurs perceptuellement différentes sont toujours loin l'une de l'autre. Cet espace repose sur la perception de la couleur par l'œil humain. Ainsi, la composante *L* représente la luminosité; sa plage de valeur varie entre 0 % pour la couleur noire et 100 % pour la couleur blanche. Les deux autres composantes *a* et *b* décrivent la couleur. La composante *a* permet de parcourir l'axe de couleur rouge-vert, pour une gamme de couleurs allant du vert (-128) au rouge (+127), et la composante *b* parcourt l'axe de couleur jaune-bleu, pour une gamme de couleurs allant du bleu (-128) au jaune (+127). La figure 2.5 présente une illustration de cet espace de couleur.

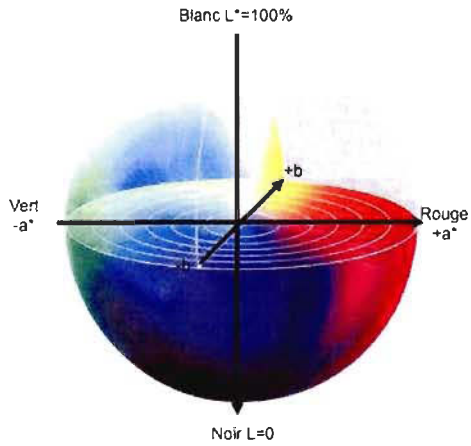


Figure 2.5: L'espace de couleur CIELab.

L'espace de couleur Lab est issu de l'espace de couleur CIE XYZ. Donc, pour obtenir les composantes L*a*b* à partir de RGB, nous devons commencer la transformation vers XYZ comme suit :

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.618 & 0.177 & 0.205 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.056 & 0.944 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Puis, les équations de passage de l'espace XYZ vers l'espace L*a* b* sont les suivantes:

$$L* = 116 \times \left(\frac{Y}{Y_n} \right)^{1/3} - 16 \qquad \text{Pour} \quad \frac{Y}{Y_n} > 0.008856$$

$$L* = 903.3 \frac{Y}{Y_n} \qquad \text{Pour} \quad \frac{Y}{Y_n} \leq 0.008856$$

$$a* = 500 \times \left[\left(\frac{X}{X_n} \right)^{1/3} - \left(\frac{Y}{Y_n} \right)^{1/3} \right] \qquad b* = 200 \times \left[\left(\frac{Y}{Y_n} \right)^{1/3} - \left(\frac{Z}{Z_n} \right)^{1/3} \right]$$

où X_n, Y_n et Z_n sont les coordonnées du blanc de référence : X_n= 94.81, Y_n= 100 et Z_n= 107.3 (sous l'illuminant standard D65 et pour une incidence inférieure à 10 °).

Bien que l'espace CIELab a l'avantage de modéliser la vision humaine, et il est indépendant vis-à-vis du matériel utilisé, il présente par nature un inconvénient quand il s'agit d'estimer avec précision les caractéristiques chromatiques. Cela a pour effet que les variations de couleurs sur les axes *a* ou *b* sont cinq fois moins visibles que les variations de luminance dans les applications de traitement d'images.

2.2.5 L'espace YUV

L'espace de couleur YUV est utilisé principalement dans les systèmes de diffusion télévisuelle PAL et NTSC. Il code une image colorée ou une vidéo en prenant en considération la perception humaine des couleurs. La composante *Y* de cet espace représente la luminance, en d'autres termes, le niveau de luminosité, alors que les composantes *U* et *V* représentent la chrominance qui décrit la couleur. La figure 2.6 présente une illustration de l'espace de couleur YUV.

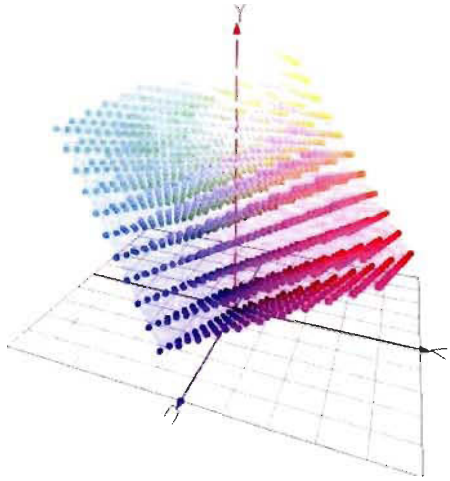


Figure 2.6 : L'espace de couleur YUV.

Nous pouvons obtenir la valeur des composantes YUV à partir de RGB selon la formule suivante :

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} +0.299 & +0.587 & +0.114 \\ -0.147 & -0.289 & +0.436 \\ +0.615 & -0.515 & -0.100 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \qquad \text{où } R,G,B \in [0,1]$$

Cet espace de couleur a plusieurs avantages liés à son utilisation dans le domaine de la télévision. Premièrement, il s'agit de sa compatibilité avec les anciens modèles de télévision analogique noir et blanc. Deuxièmement, il est adapté aux compressions digitales ou analogiques. Par contre, son inconvénient est sa dépendance du matériel utilisé.

2.3 Les moments de la couleur

Nous savons de la théorie des probabilités qu'une distribution d'une variable aléatoire peut être caractérisée de façon unique par ses moments. Ces moments peuvent être d'un

ordre quelconque positif, quoique les plus couramment utilisés soient les moments d'ordre un, deux, trois et quatre.

a. Le moment d'ordre un correspond à la moyenne des valeurs d'une distribution.
b. Le moment d'ordre deux est l'écart-type. C'est une mesure qui permet de caractériser la dispersion des valeurs d'une distribution par rapport à la moyenne.
c. Le moment d'ordre trois est le coefficient de dissymétrie (*skewness*). C'est une mesure qui permet de caractériser l'asymétrie d'une distribution.
d. Le moment d'ordre quatre correspond au coefficient d'aplatissement (*kurtosis*). C'est une mesure qui permet de caractériser l'aplatissement de la distribution d'une variable aléatoire. Autrement dit, elle mesure la disposition des masses de probabilités autour de leur centre.

Les figures 2.7 et 2.8 montrent des illustrations des moments d'ordre un, deux, trois et quatre.

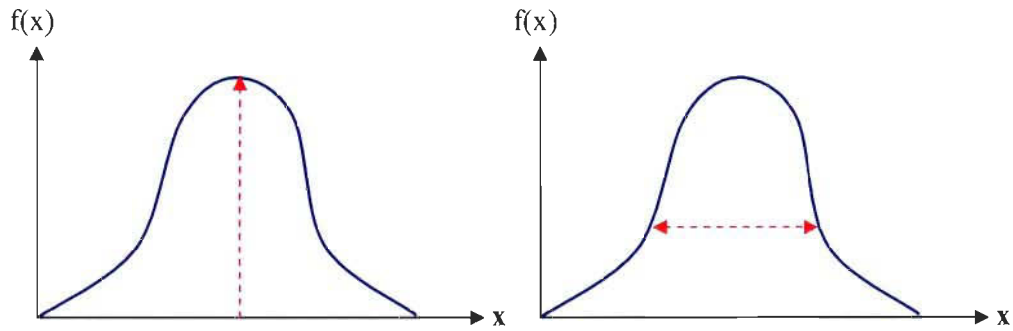
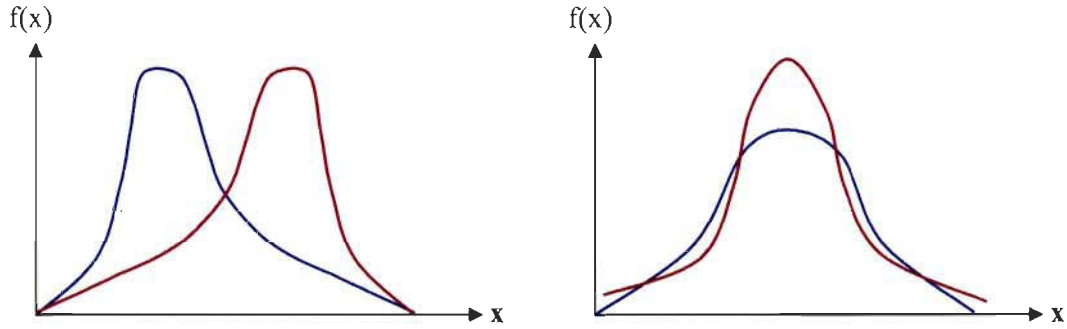


Figure 2.7 : Illustration des moments d'une distribution : (a) la moyenne, (b) l'écart-type.



(a) (b)

Figure 2.8 : Illustration des moments d'une distribution : (a) le coefficient de dissymétrie (b) le coefficient d'aplatissement.

La base de l'utilisation des moments de la couleur comme caractéristiques qui représentent une image repose sur le principe que la distribution des couleurs dans une image peut être interprétée comme une distribution de probabilité. Selon Stricker et Orengo [92], les moments qui peuvent caractériser le mieux une image sont les moments de la couleur d'ordre un, deux et trois.

Le moment d'ordre un représente la couleur moyenne de l'image. Le moment d'ordre deux caractérise le contraste d'une image. Plus la variance des couleurs est grande, plus l'image est contrastée. Le moment d'ordre trois caractérise la quantité de lumière dans une image. Une image avec un coefficient de dissymétrie positif a tendance à apparaître plus sombre et plus brillante qu'une image semblable avec un coefficient de dissymétrie inférieur.

La formule générale pour calculer le moment d'ordre h (h= 1, 2, etc.), d'une image de dimension NxM, est comme suit :

$$M_{h,b} = \left(\frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(P_b(i,j) - M_{1,b} \right)^h \right)^{\frac{1}{h}}$$

$$M_{1,b} = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} P_b(i,j)$$

où $P_b(i,j)$ est l'intensité de la couleur au point qui a les coordonnées (i,j) dans l'image, et b (b= 1, 2, 3) est le numéro de la bande de couleur.

Les avantages de l'utilisation des moments de la couleur comme caractéristique sont leurs tailles réduites, leur vitesse d'extraction, et leurs invariances à la rotation, à l'échelle et à la translation. En contrepartie, ils ont quelques inconvénients. Premièrement, ce sont des caractéristiques globales qui ne permettent pas de caractériser les détails à l'intérieur de l'image. Ainsi, deux images différentes peuvent avoir les mêmes valeurs de moments, alors que deux images semblables peuvent avoir des valeurs différentes de moments. Deuxièmement, les moments de la couleur ne sont pas tous invariants au changement d'illumination dans l'image.

2.4 L'histogramme de la couleur

L'histogramme de la couleur est une représentation de la distribution des couleurs dans une image. Il est produit en découpant d'abord les bandes de l'espace de couleur utilisé dans un certain nombre de cases, puis en comptant le nombre de pixels dans chaque case. Formellement, l'histogramme de couleurs est défini comme suit :

$$h_{A,B,C}[a,b,c] = N.\ Prob\{A=a,\ B=b,\ C=c\}$$

où A, B et C représente les bandes de couleur dans l'espace de couleur choisie (RGB, HSV, etc.), et N est le nombre de pixels dans l'image. La figure 2.9 illustre un exemple d'une image et ses différents histogrammes de la couleur, c.-à-d. un histogramme pour chaque bande de couleur.

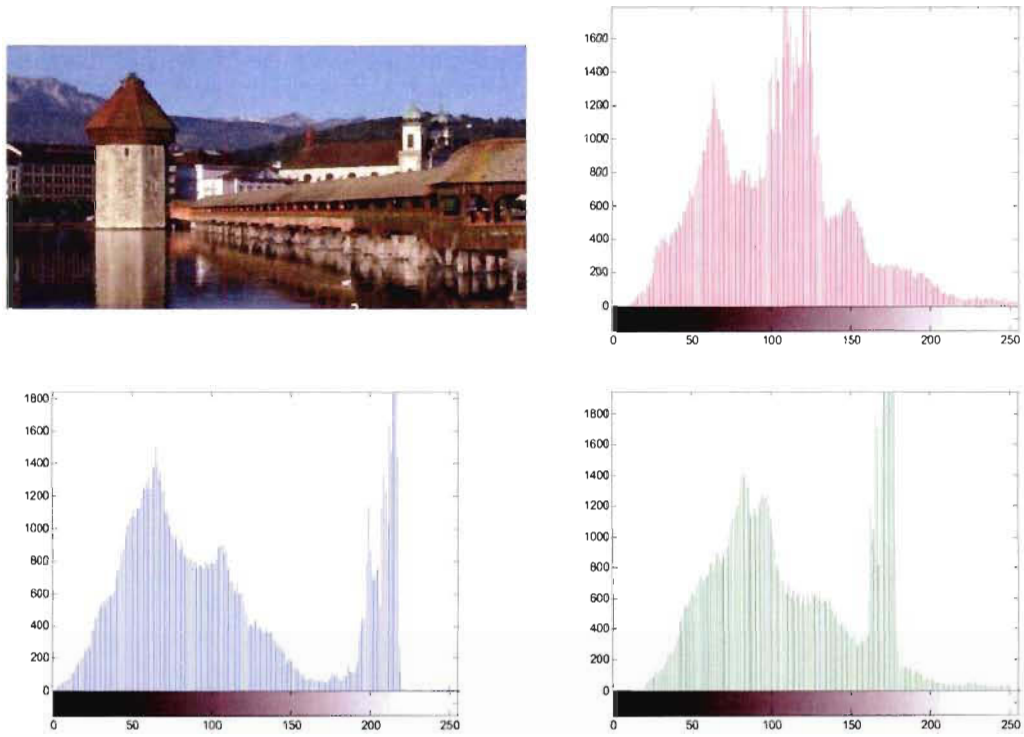


Figure 2.9 : Les différents histogrammes d'une image couleur.

L'histogramme de la couleur est couramment utilisé dans la recherche d'images et de la vidéo par le contenu en raison de ses nombreux avantages. Micheal Swain et Dana Ballard [9] sont parmi les premiers à l'avoir utilisé en recherche d'images, en 1991. Parmi les avantages de son utilisation, nous pouvons citer les suivantes :

- L'extraction de l'histogramme est facile et rapide.
- Il est invariant à plusieurs transformations comme la translation, la rotation, le changement d'échelle et le point de vue de l'image.
- Généralement, il représente bien le contenu de l'image.
- Différentes mesures de similarité peuvent y être appliquées.

Par contre, afin que l'histogramme soit utilisé efficacement, il faut d'abord régler un certain nombre de questions.

- L'histogramme de la couleur ne contient pas d'informations spatiales. En d'autres termes, il ne donne aucune information sur l'emplacement des objets dans l'image. En effet, l'histogramme nous informe sur les couleurs présentes dans l'image et la proportion occupée par chacune. Cependant, il ne fournit aucune information sur la couleur d'une zone en particulier de l'image, ni sur l'endroit où une couleur est présente dans l'image, ni sur le fait qu'une couleur correspond à une seule région ou à des régions disjointes. La figure 2.10 illustre bien ce problème.

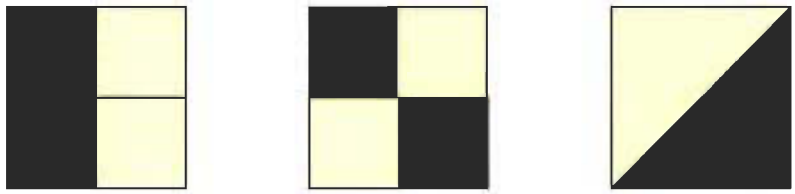


Figure 2.10 : Des images perceptuellement différentes avec des histogrammes de la couleur identique.

Parmi les solutions proposées pour résoudre ce problème, Hadjidemetriou et al. [94] ont utilisé l'histogramme de l'image ensemble, avec les différences entre les histogrammes de la même image à différentes résolutions pour encoder les informations spatiales. Cependant, l'efficacité de cette méthode dépend de la forme et de la texture de l'image.

- L'histogramme de la couleur n'est pas invariant à l'illumination. Dans l'espace de couleur RGB, la distribution des couleurs dans une image change proportionnellement avec l'illumination. Ainsi, l'histogramme de la même image change selon son degré d'illumination. Parmi les solutions proposées pour résoudre ce problème, il y a l'utilisation d'un pourcentage de couleur des pixels voisins [95] ou l'utilisation des moments invariants [96, 97].

- L'histogramme de la couleur a une grande dimension. Sans quantification, il est une caractéristique de dimension très élevée. Par exemple, si nous avons dans chaque bande de couleur 256 intensités différentes, nous obtenons un histogramme de 256^3 cases, dont la plupart sont vides. Ceci constitue un handicap majeur : la recherche n'est plus précise, le temps de recherche devient inacceptable, et la mémoire nécessaire est énorme. Une solution évidente pour réduire la dimension de l'histogramme de la couleur est de réduire la gamme des couleurs. Cela peut être fait, parce que l'œil humain ne fait pas la différence entre des couleurs proches. Par exemple, nous pouvons remarquer qu'il est très difficile de distinguer la différence entre les couleurs de la première et de la deuxième case dans la Figure 2.11.

Figure 2.11 : La différence perceptuelle entre les couleurs.

Ainsi, au lieu de diviser chaque bande en 256 couleurs différentes, on peut la diviser en n intervalles (couleurs différentes) où n est beaucoup plus petit que 256 (ex. $n = 8$). À la fin, on se retrouve avec un histogramme de n^3, ce qui est beaucoup plus petit que 256^3. Il faut bien sûr trouver un découpage judicieux qui assure que chaque couleur tombe dans une case à part. Parmi les solutions pour résoudre ce problème de dimension, Wong et al. [98] ont proposé de réduire le nombre des couleurs utilisées à extraire l'histogramme de la couleur en produisant une palette commune. Ils ont produit cette palette en regroupant les couleurs perceptuellement semblables. Deng et al. [99] ont proposé une autre méthode basée sur l'observation qu'un petit nombre de couleurs est habituellement suffisant pour caractériser l'information de couleur dans une région de l'image. Ils ont groupé les couleurs dans une région donnée dans un certain nombre de couleurs représentatives. Puis, ils ont extrait le vecteur

caractéristique à partir des couleurs représentatives et leur distribution dans les régions. Kherfi et al. [100] ont proposé la division de chaque bande de l'espace de couleur RGB en trois cases. Ainsi, ils ont réduit radicalement la dimension du vecteur caractéristique de l'histogramme de la couleur de 2^{24} à 27 cases.

- L'histogramme de la couleur a l'inconvénient des similarités entre les cases. En d'autres termes, lors de la quantification d'une image, des pixels avec des couleurs qui sont perceptuellement très semblables peuvent être placés dans des cases d'histogramme différentes, mais voisines. Cela peut mener que, la différence entre deux histogrammes est beaucoup plus grande que la différence perceptuelle entre deux images. Parmi les solutions proposées pour résoudre ce problème, c'est l'ajout à chaque case de l'histogramme les couleurs qui se situent à sa frontière. Ainsi, El-Feghi et al. [101] ont proposé une méthode basée sur la contribution de la couleur de chaque pixel dans l'image à toutes les cases de l'histogramme, à travers l'utilisation des fonctions d'ensemble flou.

Maintenant que nous avons présenté les avantages et les problèmes qu'il faut régler pour utiliser l'histogramme de la couleur comme caractéristique en recherche d'images et de la vidéo, nous allons donner plus de détails sur quelques mesures de similarité utilisées avec les histogrammes.

2.4.1 Les mesures de similarité

Plusieurs formules de distance pour mesurer la similarité entre les histogrammes de la couleur ont été utilisées. Nous pouvons citer les mesures ci-dessous.

2.4.1.1 La distance Euclidienne

La distance Euclidienne entre deux histogrammes de la couleur h et g est calculée pour chaque bande de couleur comme suit :

$$D(h,g) = \sqrt{\sum_{i=0}^{n-1}(h(i) - g(i))^2}$$

où n est le nombre de cases de l'histogramme.

Cette mesure compare les cases identiques dans les histogrammes respectifs, où toutes les cases contribuent à parts égales à la distance.

2.4.1.2 La distance de Mahalanobis

La distance de Mahalanobis entre deux histogrammes de couleur h et g est calculée pour chaque bande de couleur comme suit :

$$D(h,g) = \sqrt{(h-g)^T P^{-1}(h-g)}$$

où P est la matrice de covariance.

Cette mesure prend en compte la corrélation entre les deux histogrammes, et elle est invariante au changement d'échelle.

2.4.1.3 L'intersection d'histogramme

La distance entre deux histogrammes de couleur h et g est calculée pour chaque bande de couleur comme suit :

$$D(h,g) = \frac{\sum_{A}\sum_{B}\sum_{C}\min(h(a,b,c), g(a,b,c))}{\min(|h|,|g|)}$$

où |h| et |g| sont les magnitudes des histogrammes, lesquelles sont égales au nombre d'échantillons, et a, b et c représentent les bandes de couleur.

2.4.1.4 La distance quadratique

La distance quadratique entre deux histogrammes de la couleur h et g est calculée pour chaque bande de couleur comme suit :

$$D(h,j) = \sqrt{(h-g)^T A^{-1}(h-g)}$$

où h et g sont considérés comme des vecteurs de dimension K, et A = $[a_{ij}]$ est une matrice de dimension KxK, avec a_{ij} une distance quelconque mesurant la similitude entre la classe i et la classe j.

$$a_{ij} = 1 - d_{ij} / \max(d_{ij})$$

où d_{ij} est la distance L_2 entre la couleur i et j dans l'espace de couleur.

2.4.1.5 La distance EMD (*Earth mover distance*)

La distance EMD entre deux histogrammes est le travail minimal pour rendre les deux histogrammes identiques, en transportant le contenu des colonnes qui diffèrent d'un histogramme à l'autre. La distance entre deux histogrammes de la couleur h et g est calculée pour chaque bande de couleur comme suit :

$$EMD(h,g) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}$$

où

- n est le nombre de cases de l'histogramme h;
- m est le nombre de cases de l'histogramme g;
- $d_{ij} = d(h_i, g_j)$ est la distance entre h_i et g_j, c.-à-d. la distance entre la case i de l'histogramme h et la case j de l'histogramme g;
- f_{ij} est la quantité de masse transportée de la case i de l'histogramme h vers la case j de l'histogramme g, ou vice versa.

2.5 La texture

La texture existe partout dans la nature. Elle se réfère aux propriétés tenues et aux sensations causées par la surface externe d'objets reçus par le sens du toucher. Elle est aussi parfois utilisée pour décrire les sensations non tactiles. Littérairement, c'est une répétition spatiale d'un même motif dans différentes directions de l'espace. La texture peut avoir plusieurs qualités, elle peut être décrite comme grossière, fine, lisse, tachetée, granuleuse, marbrée, régulière ou irrégulière. La figure 2.12 montre quelques types de textures existantes dans la nature.

Figure 2.12 : Différents modèles de texture.

En traitement d'images, la texture est une région de l'image qui a des caractéristiques cohérentes et homogènes, formant un tout pour un observateur. Elle est composée de petits éléments répétitifs. Généralement, la répétition peut impliquer des variations locales d'échelle, d'orientation, ou d'autres caractéristiques géométriques et optiques des

éléments. Une définition plus formelle de la texture a été donnée par Haralick [102]. Il décrit la texture comme un phénomène à deux dimensions. La première dimension décrit les éléments de base ou primitives, qui sont les motifs à partir desquels est formée la texture. La deuxième dimension décrit l'organisation spatiale de ces primitives.
Dans la nature, nous distinguons deux grandes classes de textures selon le niveau de perception visuelle :

a. Les macro-textures : ce sont des textures régulières, formées de motifs répétitifs et périodiques, spatialement placées selon une règle précise. Le mur de brique représente un bon exemple de ce type de texture. Les méthodes les plus adaptées pour décrire ce type de textures sont les approches fréquentielles ou structurelles.
b. Les micro-textures : ce sont des textures composées de primitives (microscopiques) distribuées de manière aléatoire. L'herbe ou le gazon représente un bon exemple de ce type de texture. Les méthodes les plus adaptées pour décrire ce type de textures sont les approches probabilistes.

Cependant, ces deux classes de texture se complètent naturellement, car une texture n'est jamais strictement périodique ni totalement aléatoire.

2.5.1 Les différentes méthodes d'analyse de la texture

Le but de l'analyse de la texture est de formaliser ses caractéristiques par des paramètres mathématiques qui serviraient à l'identifier. Pour cela, différentes méthodes ont été proposées dans la littérature. Nous pouvons classifier ces méthodes en quatre différentes approches :

a. Les approches statistiques : ces approches étudient les relations entre un pixel et ses voisins. Elles sont utilisées pour caractériser les structures sans régularité apparente. La méthode la plus fréquemment citée est la méthode de la matrice de cooccurrence. [102, 103]

b. Les approches structurelles : ces approches décrivent la texture en définissant les propriétés et les règles de placement des éléments de la texture. Cependant, ces méthodes semblent être limitées dans l'aspect pratique puisqu'elles ne peuvent décrire que des textures très régulières. Les méthodes de l'histogramme invariant et celle de la décomposition morphologique en sont des exemples [102, 104].

c. Les approches basées sur les modèles : ces approches produisent un modèle empirique de chaque pixel dans l'image, basé sur une moyenne pondérée des intensités des pixels voisins. Les paramètres estimés des modèles d'image sont utilisés comme des descripteurs de caractéristiques de la texture. Les modèles autorégressifs, fractals et champs aléatoires Markoviens sont des exemples de ces approches [105-110].

d. Les approches basées sur les transformations : ces approches convertissent l'image dans une nouvelle forme en utilisant les propriétés fréquentielles de celle-ci. Le succès de ces techniques repose sur le type de transformation utilisée pour extraire les caractéristiques de la texture. Nous pouvons citer le filtre de Gabor [111, 112], le spectre de Fourier [113] et la transformation en ondelettes [114, 115, 116] comme exemples.

Selon la littérature [117, 118, 119], la méthode de la matrice de cooccurrence donne de bons résultats, c'est pourquoi la section suivante donne plus de détails sur cette méthode.

2.5.1.1 La méthode de matrice de cooccurrence

La méthode de la matrice de cooccurrence à niveau de gris (GLCM) a été proposée par Haralick et al. [120] en 1973. Elle permet de mesurer la fréquence d'apparition d'un motif formé de deux pixels séparés par une distance déterminée d dans une direction Θ par rapport à l'horizontale. Plus précisément, l'élément $P(a,b)_{d,\Theta}$ de la matrice de

cooccurrence représentent le nombre de points de niveau de gris *a* ayant pour voisin un point de niveau de gris *b* à la distance *d*.

$$d = (dx , dy)=(d \cos \Theta , \ d \ \sin \Theta)$$

où *d* est la distance entre deux points dans l'image, et Θ est l'angle entre ces deux points.

L'exemple ci-dessous (figure 2.13) présente une image à 4 niveaux de gris, et ses 4 matrices de cooccurrence dans quatre différentes directions (0°, 45°, 90°, 135°). La taille de la matrice de cooccurrence est égale à N^2, où N correspond au nombre de niveaux de gris.

Image

0	0	2	1	0
1	0	0	1	1
0	1	0	1	1
0	0	2	2	2
1	2	2	3	3

0°

	0	**1**	**2**	**3**
0	6	4	2	0
1	5	4	2	0
2	2	2	6	1
3	0	0	1	2

45°

	0	**1**	**2**	**3**
0	4	7	1	0
1	7	2	2	0
2	1	2	4	1
3	0	0	1	0

90°

	0	**1**	**2**	**3**
0	6	6	3	0
1	6	6	2	0
2	3	2	2	2
3	0	0	2	0

135°

	0	**1**	**2**	**3**
0	8	1	3	0
1	1	6	3	0
2	3	3	0	2
3	0	0	2	0

Figure 2.13 : Une image 5x5 avec 4 niveaux de gris et ses 4 matrices de cooccurrence.

La réussite de cette méthode repose sur le bon choix des paramètres qui sont : la taille de la matrice sur laquelle s'effectue la mesure, et la distance *d* qui sépare les deux pixels du motif. Ainsi, pour rendre cette méthode plus efficace, il faut limiter le nombre de calculs. Le plus souvent la taille de la matrice est réduite en choisissant un N égale à 8, 16 ou 32. La distance *d* est fixée à 1, et les valeurs de la direction Θ à 0°, 45°, 90°, 135°.

La matrice de cooccurrence n'est pas utilisée directement en traitement d'images, car elle contient une grande quantité d'informations difficilement manipulables. En 1973, Haralick et al. [120] ont proposé quatorze mesures pour caractériser la texture dite indices de la texture. L'objectif des indices est de résumer l'information contenue dans la matrice de cooccurrence et de permettre une meilleure discrimination des différents types de textures. Les caractéristiques de texture les plus souvent utilisées selon [121] sont :

a. La moyenne

$$f1 = \sum_i \sum_j p(i,j)$$

où $p(i,j)$ correspond aux éléments de la matrice de cooccurrence.

b. La variance

$$f2 = \sum_i \sum_j (i-\mu)^2 p(i,j)$$

où μ est la moyenne calculée ci-dessus, et $p(i,j)$ correspond aux éléments de la matrice de cooccurrence.

Cette mesure caractérise la distribution des niveaux de gris autour de la valeur moyenne f_1 calculée précédemment. Une forte valeur caractérise une texture fine.

c. L'énergie

$$f3 = \sum_i \sum_j p(i,j)^2$$

où $p(i,j)$ correspond aux éléments de la matrice de cooccurrence.

L'énergie caractérise l'homogénéité de l'image. Si la valeur de l'énergie est grande, cela veut dire qu'il y a beaucoup de transitions de niveaux de gris dans l'image.

d. La corrélation

$$f4 = \sum_i \sum_j (i-\mu_x)(j-\mu_j)p(i,j)$$

où μ_x et μ_y sont respectivement les moyennes des lignes et des colonnes de la matrice de cooccurrence, et *p(i,j)* correspond aux éléments de la matrice.

Cet indice mesure la corrélation de la distribution des niveaux de gris dans l'image.

e. L'entropie

$$f5 = -\sum_i \sum_j p(i,j) \log p(i,j)$$

où *p(i,j)* correspond aux éléments de la matrice de cooccurrence.

L'entropie est une mesure de complexité de l'image. Elle permet de caractériser le degré de grandeur des granules dans l'image.

f. Le contraste

$$f6 = \sum_i \sum_j (i-j)^2 p(i,j)$$

où *p(i,j)* correspond aux éléments de la matrice de cooccurrence.

Cette mesure permet de caractériser la netteté de la texture. Le contraste est élevé lorsque les variations des niveaux de gris sont importantes.

g. L'homogénéité

$$f7 = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i.j)$$

où *p(i,j)* correspond aux éléments de la matrice de cooccurrence.

Cet indice est l'inverse du contraste. Une forte valeur caractérise une texture homogène.

h. Le *cluster shade*

$$f8 = \sum_i \sum_j (i+j-2\mu)^3 p(i,j)$$

où μ est la moyenne calculée en a, et *p(i,j)* correspond aux éléments de la matrice de cooccurrence.

i. Le *cluster prominence*

$$f9 = \sum_i \sum_j (i + j - 2\mu)^4 \, p(i,j)$$

où μ est la moyenne calculée en a, et *p(i,j)* correspond aux éléments de la matrice de cooccurrence.

La méthode de la matrice de cooccurrence est largement utilisée en traitement d'image, car elle est facile à implémenter et en général, donne de bons résultats en recherche d'images. Cependant, elle n'est pas adaptée pour étudier la forme détaillée des éléments de la texture.

2.6 Les contours

Dans les paragraphes précédents de ce chapitre, nous avons parlé de la couleur et de la texture, qui sont des caractéristiques intéressantes pour représenter les images et leurs contenus. Cependant, il y a des caractéristiques propres aux objets, telles que la forme, qui ne peuvent pas être représentées uniquement par la couleur et la texture. La détection de contour est très utilisée comme une étape de prétraitement afin de détecter les objets et trouver les limites des régions. En effet, un objet peut être localisé à partir de l'ensemble des points de son contour. De plus, trouver cet ensemble permet d'obtenir une information sur la forme de l'objet, et de réduire de manière significative la quantité de données et d'éliminer les informations qu'on peut juger moins pertinentes, tout en préservant les propriétés structurelles importantes de l'image. D'une façon simplifiée, le contour représente les frontières des objets ou des régions dans une image. La figure 2.14 présente une image et l'image correspondante après la détection des contours.

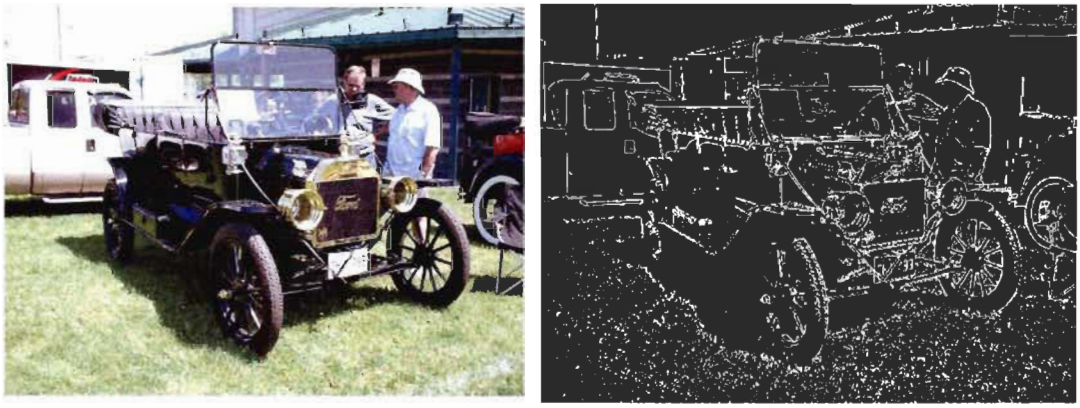
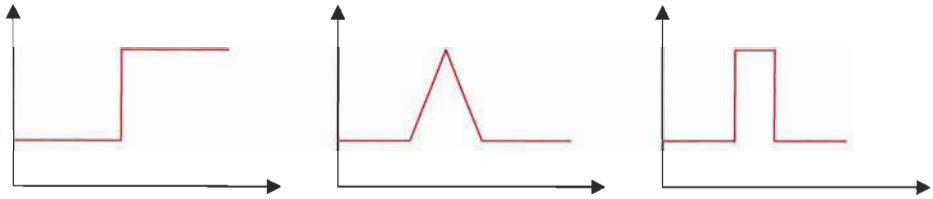


Figure 2.14 : La détection des contours d'une image.

Techniquement, la détection de contours consiste à repérer les points d'une image qui correspondent à un changement brusque de l'intensité des couleurs. En général, ces changements reflètent des événements importants à l'intérieur de l'image. Par exemple, des discontinuités dans l'éclairage d'une scène, dans l'orientation d'une surface, dans les propriétés d'un matériel ou dans la profondeur.

2.6.1 La détection de contours

Les contours dans une image sont caractérisés par des discontinuités de la fonction d'intensité dans les images. La figure 2.15 présente quelques types de contours dans le cas idéal.



(a) (b) (c)

Figure 2.15 : Quelques types de contours dans le cas idéal :
(a) marche, (b) pointe, (c) toit.

Cependant, à cause de la lentille de la caméra, le changement brusque des niveaux de gris est généralement remplacé par un changement progressif sur une courte distance. La figure 2.16 illustre cela.

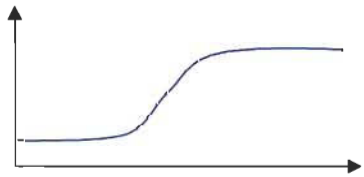


Figure 2.16: Contour avec lentille.

Le point de contour correspond au point d'inflexion dans la fonction image. Pour trouver ce point, deux méthodes sont généralement utilisées. Mathématiquement, au point d'inflexion :

a. La première dérivée de la fonction image est un Max positif ou un Min négatif. Cette propriété a donné naissance à la méthode de détection par Gradient.
b. La seconde dérivée de la fonction image est nulle et la troisième dérivée n'est pas nulle. Cette propriété est à la base de la méthode de détection par Laplacien.

La figure 2.17 donne un exemple d'une fonction image et de sa première et seconde dérivée.

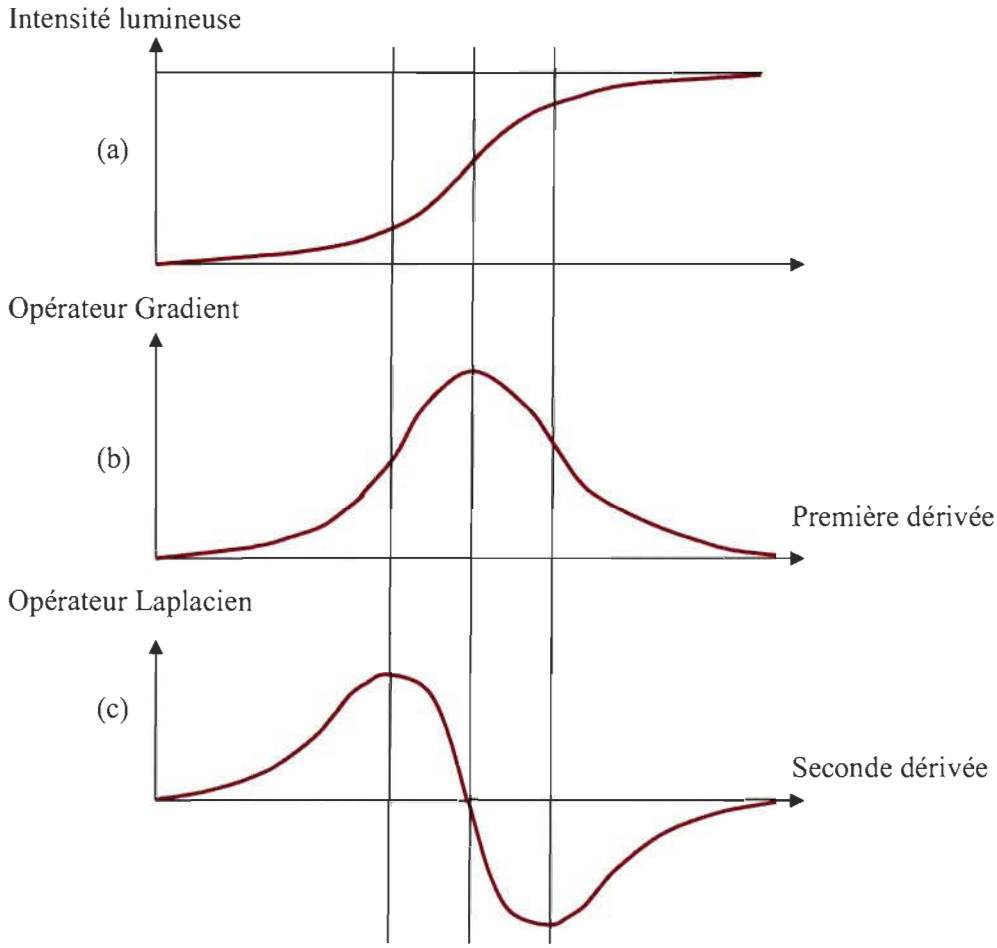


Figure 2.17 : Une fonction image et sa première et seconde dérivée. (a) La fonction image, (b) la première dérivée, et (c) la seconde dérivée.

Dans la partie suivante, nous allons présenter très brièvement les méthodes de la détection par Gradient et par Laplacien.

2.6.1.1 Le détecteur Gradient

Le principe de la détection de contours par l'utilisation du Gradient consiste à calculer d'abord le Gradient de l'image dans les deux directions orthogonales, c.-à-d. à calculer la dérivée partielle de la fonction image par rapport à x et à y. Ensuite, il faut trouver les

maximas du module du Gradient qui vont dans le même sens que l'orientation de ce dernier.

Donc, en premier, nous devons calculer le Gradient de la fonction image f en tout point de coordonnées (x,y). Il est donné par :

$$\nabla f(x,y) = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} \right)$$

La dérivée partielle de l'image par rapport à x (appelé f_x) est le résultat de la convolution de l'image f avec un masque Gx créé pour cet effet.

$$f_x = f * Gx$$

De même, la dérivée partielle par rapport à y (appelée fy) est le résultat de sa convolution avec un autre masque Gy créé pour cela.

$$f_y = f * Gy$$

Gx et Gy peuvent généralement être obtenus de deux façons :

- En utilisant un masque Gaussien : Gx est la dérivée partielle de ce masque par rapport à x, et Gy est la dérivée partielle de ce masque par rapport à y.
- En utilisant un masque prédéfini : parmi les masques les plus utilisés, nous citons les masques de Roberts, Prewitt, Sobel, Kirch et Robinson. Ci-dessous un exemple du masque de Sobel :

$$G_x = 1/4 \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y = 1/4 \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

La seconde étape consiste à calculer le module du Gradient m et son orientation Θ. Le module permet de mesurer l'importance du contour, ou plus précisément l'amplitude de la différence d'intensité existante dans l'image. Quant à l'orientation, elle mesure la direction des contours qui, en tout point de l'image, sont orthogonaux à la direction Θ. Le module du Gradient m et sa direction Θ sont respectivement donnés par :

$$m(x,y)=\sqrt{\left(\frac{\partial f(x,y)^2}{\partial x}+\frac{\partial f(x,y)^2}{\partial y}\right)}$$

$$\Theta(x,y)=\arctan\left(\frac{\partial f(x,y)}{\partial x}\Big/\frac{\partial f(x,y)}{\partial y}\right)$$

Pour repérer les points de contour, il faut rechercher les maximas du module du Gradient dans le sens de l'orientation, ou faire un seuillage adéquat et garder que les points où le module du Gradient est supérieur à un certain seuil. Pour plus de détails, voir l'article de Torre et Poggio [122].

2.6.1.2 Les détecteurs de passage par zéro du Laplacien

Cette méthode est basée sur la propriété qui stipule que la deuxième dérivée de l'image (son Laplacien) passe par zéro aux points de contour. Théoriquement, le Laplacien d'une fonction image *f(x,y)* est défini comme suit :

$$\nabla^2 f(x,y)=\frac{\partial^2 f(x,y)}{\partial x^2}+\frac{\partial^2 f(x,y)}{\partial y^2}$$

En pratique, le Laplacien d'une fonction image *f(x,y)* est le résultat de la convolution de la fonction image par un masque crée pour cet effet. Le masque peut être obtenu de deux manières différentes :

- En utilisant la seconde dérivée d'un masque Gaussien.
- En utilisant un masque prédéfini. Ci-dessous un exemple du masque :

$$G=\begin{bmatrix}0 & 1 & 0\\ 1 & -4 & 1\\ 0 & 1 & 0\end{bmatrix}$$

La détection de contours consiste à localiser les passages par zéro du résultat de la convolution. Autrement dit, les points pour lesquels le Laplacien change de signe. Puis, il faut faire un Seuillage des passages par zéro de fortes amplitudes, car cette méthode est particulièrement sensible au bruit en raison de la double dérivation. Pour plus de détails, voir [123].

2.7 Le mouvement

La vidéo est constituée d'une séquence d'images. Toutefois, l'extraction des caractéristiques visuelles que nous avons décrites dans les paragraphes précédents n'est pas suffisante pour indexer la vidéo. Cela est dû à la nécessité de décrire le contenu dynamique à l'intérieur de la vidéo. Ce dynamisme est le résultat des changements qui se produisent à travers le temps à l'intérieur du contenu des images qui constituent la vidéo. Ces changements sont le résultat du mouvement de la caméra, du mouvement des objets à l'intérieur de la vidéo, ou d'une combinaison des deux. Les mouvements de la caméra les plus courants sont : la translation, la rotation et le zoom. La figure 2.18 présente une illustration de ces différents types de mouvement de la caméra.

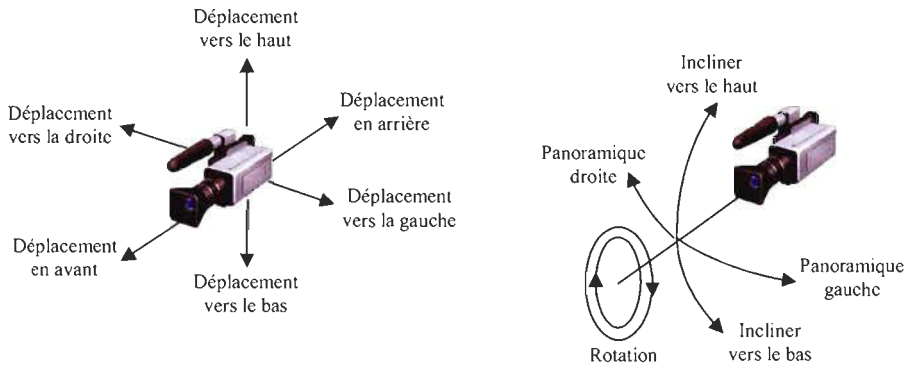


Figure 2.18 : Les mouvements courants de la caméra.

Donc, nous pouvons dire que le mouvement est une information pertinente pour l'indexation de la vidéo. Nous allons présenter dans ce qui suit les différentes caractéristiques du mouvement et les méthodes utilisées pour les extraire.

2.7.1 Les différentes caractéristiques du mouvement

Les différentes caractéristiques du mouvement dans une vidéo sont incluses dans le standard MPEG-7 [124]. Ces caractéristiques sont :

a. L'activité du mouvement (*Motion activity*) : Cette caractéristique capture la notion d'intensité du mouvement d'une façon globale dans le plan, en utilisant les différents champs de vecteurs de mouvement extraits d'un plan [125, 126, 127]. Elle inclut les attributs suivants :
   - L'intensité de l'activité : le niveau d'activité peut être haut ou bas selon le sujet de la vidéo. Par exemple, une vidéo d'une course automobile comporte un mouvement intense alors que le mouvement dans une vidéo de diner familial est peu intense.
   - La direction de l'activité : elle indique la direction dominante de l'activité dans le plan.
   - La distribution spatiale de l'activité : elle indique le nombre et la taille des régions actives dans les images qui constituent le plan.
   - La distribution temporelle de l'activité : elle indique la variation de l'activité toute au long de la durée du plan.

b. Le mouvement de la caméra (*Camera motion*) : cette caractéristique indique le type de mouvement de la caméra dans le plan (translation, zoom, rotation, etc.), son amplitude et sa localisation temporelle dans le plan. Les méthodes utilisées pour extraire cette caractéristique reposent sur des modélisations paramétriques 2D ou 3D des mouvements de la caméra [128, 129, 130] ;

c. Les paramètres de déformation (*Warping parameters*) : on estime les paramètres d'un modèle mathématique qui représente le panorama par une application. Cette application permet, à partir d'une seule image du panorama, de trouver les autres

images de ce même panorama. Notons que le panorama est un mouvement de la caméra du type zoom, affine, etc. [131, 132, 133, 134] ;

d. La trajectoire du mouvement (*Motion trajectory*) : elle décrit les déplacements des objets dans le temps. En général, c'est les positions successives dans le temps du centre de gravité d'un objet [130, 135];

e. Le mouvement paramétrique (*Parametric motion*) : cette caractéristique permet d'extraire les objets ayant des mouvements similaires et qui subissent des rotations ou des déformations. Elle utilise le même modèle paramétrique de mouvement que les paramètres de déformation [131, 132, 133, 134].

2.8 Conclusion

Il existe une multitude de caractéristiques utilisées pour la représentation du contenu visuel de la vidéo. Nous avons présenté dans ce chapitre les différents espaces de couleur et les caractéristiques les plus couramment utilisées en recherche de la vidéo, leurs avantages et inconvénients et les méthodes de leurs extractions. Cela va nous permettre de choisir les caractéristiques qui représentent bien le contenu de la vidéo et les méthodes de leurs extractions, ainsi que développer de nouvelles caractéristiques.

Chapitre 3

Notre travail

3.1 Introduction

Les systèmes de recherche de la vidéo existants se sont intéressés à des vidéos spécialisées telles que les vidéos de sport et les bulletins d'informations. Cependant, à notre connaissance, il n'y a pas de travaux qui se sont intéressés aux vidéos personnelles. Or, nous avons vu dans le premier chapitre que ce type de vidéos est omni présent et que leur nombre est en pleine expansion. Par conséquent, développer un système qui s'occupe de ce genre de vidéos est devenu crucial. Nous nous sommes donc attaqués à cette problématique en développant un outil qui permet à la fois d'organiser ces vidéos et de les localiser suite à une requête formulée par l'utilisateur. Par ailleurs, nous avons développé une nouvelle méthode pour l'extraction de l'image clé. Notons que le choix de cette image est crucial, puisque c'est elle qui remplace le plan lors de l'extraction des caractéristiques. Par conséquent, un bon choix de l'image clé contribue considérablement à l'amélioration des performances du système. Nous avons aussi développé une nouvelle caractéristique qui représente bien le contenu de l'image. C'est une combinaison de l'histogramme de la couleur et des points de contour. Nous l'avons appelé l'histogramme de la couleur aux alentours des points de contours.

Dans ce chapitre, nous allons présenter l'architecture de notre système, les modules qui le constituent, les caractéristiques des vidéos que nous avons extraites et les méthodes de leur extraction, ainsi que la méthode de comparaison que nous avons utilisée lors de la recherche. À la fin, nous allons expliquer le fonctionnement de l'interface de notre système.

3.2 Architecture de notre système

Notre but est de créer un système d'indexation et de recherche de la vidéo par le contenu. Comme illustré dans la figure 3.1, notre système comprend plusieurs modules. Ces modules sont : le module de la segmentation temporelle, le module de l'extraction des caractéristiques, le module de l'extraction des images clés, le module de la formulation de la requête, le module du calcul des descripteurs et le module de la comparaison des vidéos.

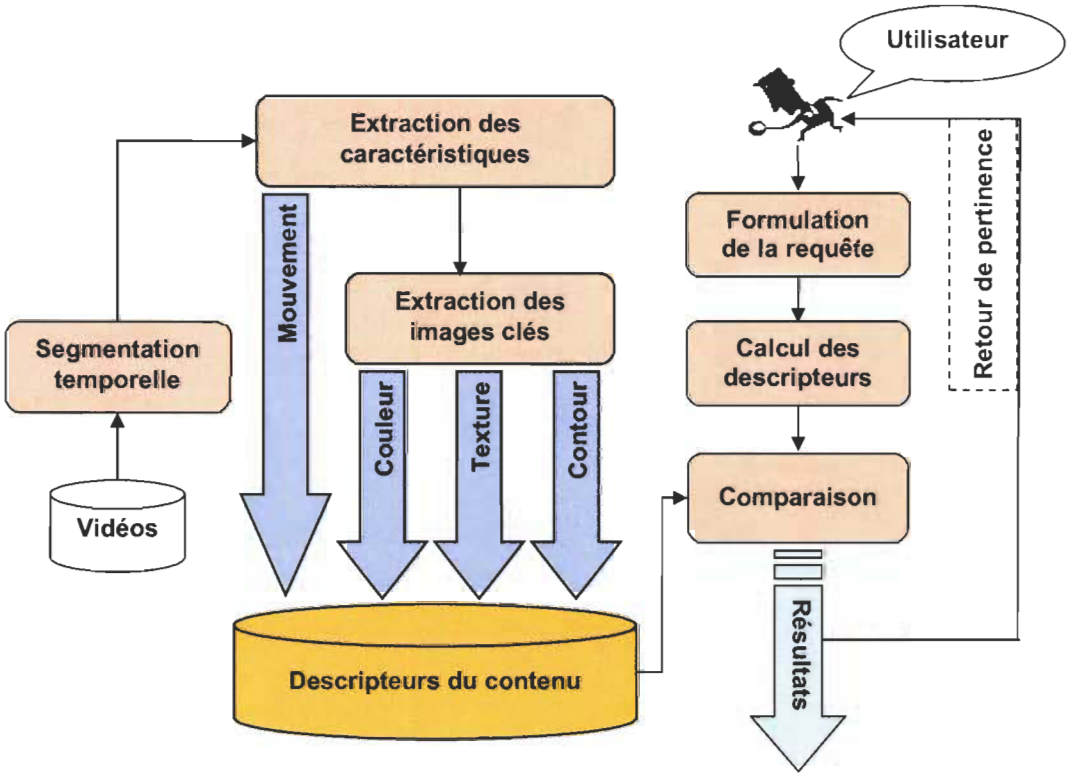


Figure 3.1: Architecture du système d'indexation de la vidéo par le contenu.

En premier, les vidéos de la base de données sont découpées en « plan » ou « *shot* » par le module de la segmentation temporelle. Cette étape est nécessaire pour rendre l'analyse et l'indexation des vidéos moins complexes. Ensuite, pour les besoins de l'indexation, le module de l'extraction des caractéristiques s'occupe du calcul des

caractéristiques visuelles, sonores, textuelles et sémantiques de chaque segment vidéo dans la base de données. Dans notre cas, nous avons limité notre système aux caractéristiques visuelles, car l'extraction des caractéristiques sonores fait appel à des techniques de traitement du son qui dépassent le cadre de notre spécialité. Pour chaque plan, ce module extrait la caractéristique du mouvement de la vidéo, en plus d'une image clé qui la représente. Cette image est ensuite utilisée pour extraire les caractéristiques visuelles représentatives du plan. Ces caractéristiques sont les moments de la couleur, l'histogramme de la couleur, la texture, l'histogramme de la couleur aux alentours des points de contour et le pourcentage des points de contour. Les valeurs des caractéristiques que nous avons extraites sont normalisées puis sauvegardées dans un fichier pour usage ultérieur.

Quand un utilisateur veut faire une recherche, il doit choisir une vidéo via le module de la formulation de la requête. Ensuite, il doit sélectionner les caractéristiques visuelles qui sont utilisées pour la recherche des vidéos similaires à sa requête, puis lancer la recherche. Le module qui fait la comparaison calcule la distance entre la vidéo requête de l'utilisateur et les vidéos de la base de données. À la fin, les résultats de la comparaison sont affichés à l'utilisateur. Si ce dernier n'est pas satisfait, il peut choisir une des vidéos résultantes comme requête pour raffiner sa recherche.

Dans ce qui suit, nous allons donner plus de détails et expliquer le fonctionnement de chaque module de notre système d'indexation et de recherche de la vidéo.

3.2.1 Le module de prétraitement

L'opération de segmentation temporelle, d'extraction des caractéristiques et d'extraction des images clés est une tâche lente qui consomme beaucoup de ressources machine. Cela est dû à la quantité considérable de données à traiter. Par exemple, un fichier vidéo en format PAL d'une durée d'une minute est composé de 1500 images, où chaque image peut contenir jusqu'à plusieurs millions de pixels. Un des moyens pour résoudre ce problème est d'effectuer le prétraitement en « *offline* », c.-à-d. l'effectuer avant même de

permettre à l'utilisateur d'utiliser le système. Une fois les caractéristiques extraites, elles seront sauvegardées pour être utilisées au moment de la recherche. Dans ce qui suit, nous allons détailler chacune des étapes du prétraitement.

3.2.1.1 La segmentation temporelle

Ce module permet de découper les vidéos de la base de données en unités logiques élémentaires, appelées plan ou « *shot* ». Comme nous l'avons expliqué dans le premier chapitre, une vidéo est physiquement constituée de plans et sémantiquement de scènes. L'objectif est de rendre l'analyse et l'indexation des vidéos plus simple. Puisque cette tâche est compliquée, et elle n'est pas le but de notre recherche, nous avons découpé les vidéos manuellement pour les besoins de nos expérimentations en utilisant un logiciel d'édition de la vidéo.

3.2.1.2 L'extraction des caractéristiques

Afin d'indexer les vidéos de la base de données, il faut extraire les caractéristiques de leur contenu. Dans notre cas, nous nous sommes focalisés sur les caractéristiques visuelles de ces vidéos. Ainsi, après le découpage des vidéos en plans, nous avons analysé et extrait leurs caractéristiques. Suivant l'étude que nous avons faite dans le deuxième chapitre sur les caractéristiques les plus utiles en recherche de vidéos, nous avons choisi d'extraire les caractéristiques suivantes : les moments de la couleur, l'histogramme de la couleur, la texture, l'histogramme de la couleur aux alentours des points de contour, le pourcentage des points de contour et le mouvement. Notons que nous avons extrait le mouvement à partir du plan, et le reste des caractéristiques à partir de l'image clé.

Dans ce qui suit, nous allons présenter comment nous avons extrait l'image clé et les caractéristiques que nous avons développées et implémentées dans notre moteur de recherche.

3.2.1.3 L'extraction de l'image clé

Une vidéo est constituée d'une séquence d'images qui se succèdent à une vitesse prédéfinie selon le format de la vidéo, et une bande sonore. L'analyse puis l'indexation de chaque image dans une vidéo rendent n'importe quel moteur de recherche de la vidéo inefficace, à cause de la quantité d'information qu'il faut gérer. La solution que nous proposons est d'extraire pour chaque segment de la vidéo, appelée plan, une image clé qui représente son contenu visuel. L'objectif est de ramener le problème de la recherche de la vidéo en un problème de recherche d'images.

Idéalement, l'image clé doit capturer le contenu sémantique du plan qu'elle représente. Cela reste très subjectif, car c'est très difficile de définir la notion d'image clé d'une vidéo. Prenons un exemple. Nous avons un plan d'une durée de cinq minutes dont le contenu montre des personnes qui dansent dans une fête de mariage, et au milieu de la séquence vidéo, il y a une personne qui subitement tombe par terre. Quelle est l'image qui représente la sémantique de la vidéo? Une image des personnes qui dansent ou une image de la personne qui tombe? L'une des solutions qui permettent d'aboutir à un résultat satisfaisant consiste à compléter l'analyse du contenu visuel de la vidéo par une analyse du contenu sonore et une annotation manuelle ou automatique.

Afin d'extraire l'image clé d'un plan, nous calculons d'abord pour chaque image qui constitue la séquence vidéo, la moyenne de la couleur rouge, verte et bleu. Nous l'appelons *Moy(R, G, B)*. Ensuite, nous calculons la moyenne générale des couleurs RGB de la vidéo. Ci-dessous la formule utilisée pour l'extraction de la moyenne générale des couleurs d'une vidéo.

$$Moy(R,G,B) = \frac{\sum_{i=1}^{n} Moy(R_i, G_i, B_i)}{n}$$

où n est le nombre d'images dans la séquence vidéo, et i représente les différentes images qui constituent cette vidéo.

Finalement, l'image clé de notre séquence est l'image la plus proche de la moyenne de cette séquence. En d'autres termes, il faut d'abord calculer la distance entre chaque image membre de la séquence et la moyenne de cette séquence (en termes de vecteurs (R, G, B)), puis choisir l'image ayant la distance la plus petite. La figure 3.2 montre une illustration de notre solution. Les signes « + » dans l'espace RGB représentent chacune une image de la séquence vidéo traitée. Le cercle représente le centre du nuage de points ou la moyenne des couleurs RGB de la séquence vidéo. Les coordonnées des points sont les moyennes des couleurs RGB.

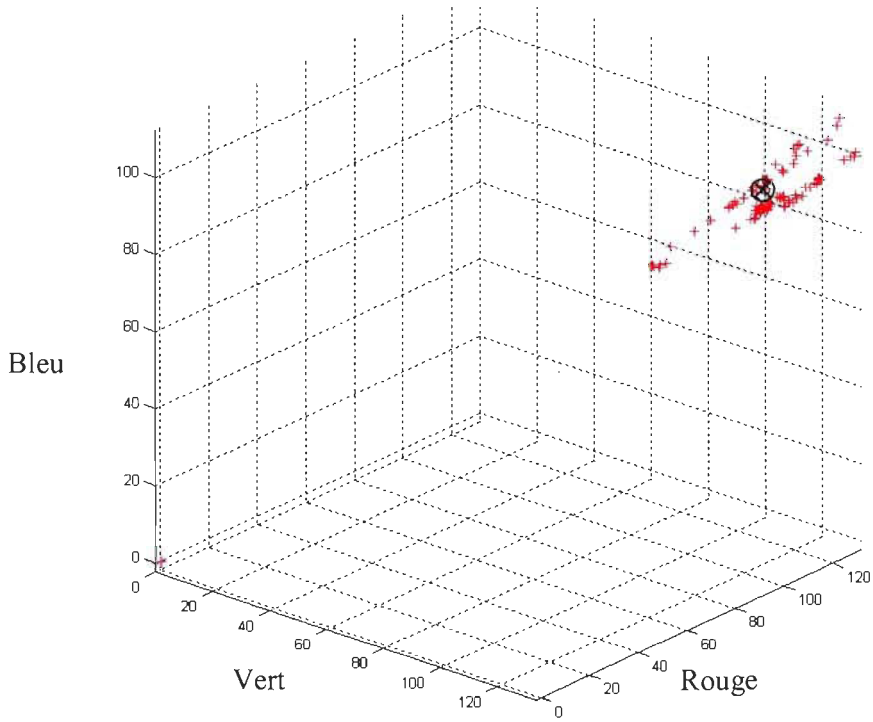


Figure 3.2 : Une représentation des images d'une séquence vidéo et de son image clé.

Maintenant que nous avons réduit l'analyse des vidéos déjà découpées à une analyse d'images, nous allons voir comment analyser ces images et extraire leurs caractéristiques visuelles pour les indexer.

3.2.1.4 Les caractéristiques

Les caractéristiques que nous avons implémentées sont :

1. Les moments de la couleur :

   Les premières caractéristiques que nous avons implémentées dans notre système sont les moments de la couleur d'ordre 1 jusqu'à 6. Notons que les trois premiers moments sont très utilisés en statistique; ils représentent respectivement la moyenne, l'écart-type et le coefficient de dissymétrie (*skewness*). Les moments sont calculés à partir de l'image clé de chaque vidéo. Pour ces caractéristiques, nous avons opté pour l'utilisation de l'espace de couleur RGB, car ça ne demande aucune transformation des couleurs, c'est rapide à calculer et donne généralement de bons résultats. Ainsi, pour chaque bande de couleur, respectivement R, G et B, nous avons calculé les moments d'ordre h où h va de 1 jusqu'à 6 selon la formule suivante :

$$M_{h,b} = \left(\frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(P_{b\backslash}(i,j) - M_{1,b} \right)^h \right)^{\frac{1}{h}}$$

$$M_{1,b} = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} P_b(i,j)$$

   où $P_b(i,j)$ est l'intensité de la couleur au point qui a les coordonnées (i,j) dans l'image, N et M sont respectivement le nombre de lignes et de colonnes dans l'image, et b (b= 1, 2, 3) est le numéro de la bande de couleur.

2. L'histogramme de la couleur :

   Comme nous avons vu dans le deuxième chapitre, l'histogramme de la couleur a été largement utilisé dans la recherche d'images et de la vidéo, c'est pourquoi nous l'avons utilisé comme deuxième caractéristique. Nous avons calculé l'histogramme de la couleur des images clés des vidéos. En premier, en utilisant

l'espace de couleur RGB, puis le HSV pour améliorer les performances de la recherche. Formellement, l'histogramme de la couleur est défini comme suit :

$$h_{A,B,C}[a,b,c] = N.\ Prob\{A=a,\ B=b,\ C=c\}$$

où A, B et C représentent les bandes de couleur dans l'espace de couleur choisie (RGB ou HSV), et N est le nombre de points dans l'image.

L'histogramme dans son état brut comporte trop d'informations. Afin de réduire la dimension de l'histogramme, nous avons suivi la solution proposée par Kherfi et al. dans [100]. Nous avons décomposé l'espace de couleur en 27 sous-espaces, en divisant les intensités dans chaque bande de couleur en trois parties égales. Le résultat est un vecteur de 27 cases seulement. Les figures 3.3 et 3.4 montrent respectivement une illustration du découpage de l'espace RGB et HSV.

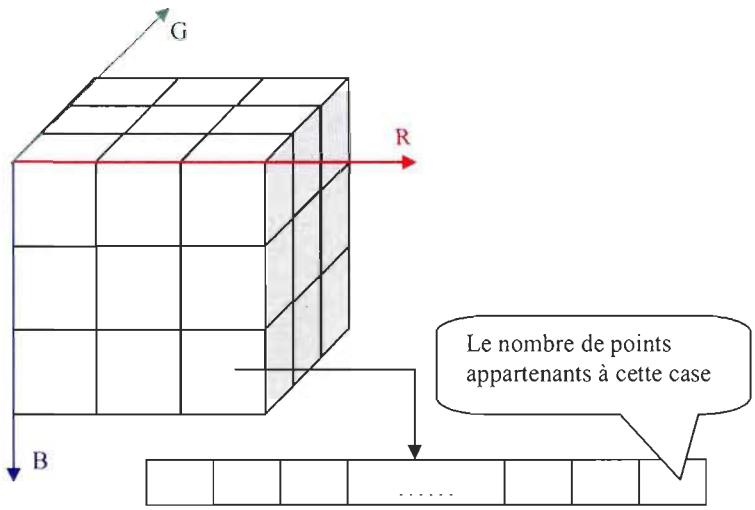


Figure 3.3 : L'histogramme de la couleur dans l'espace RGB.

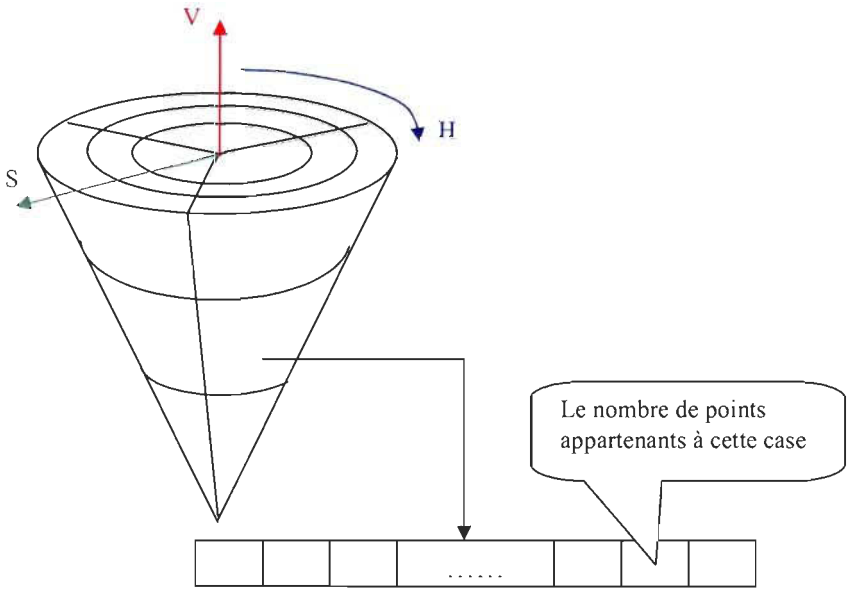


Figure 3.4 : L'histogramme de la couleur dans l'espace HSV.

3. La texture :

   Bien que les caractéristiques que nous avons déjà implémentées (les moments de la couleur et l'histogramme de la couleur) représentent bien les couleurs de l'image, elles ne sont pas suffisantes pour représenter tout son contenu visuel. C'est la raison pour laquelle nous avons ajouté l'analyse de la texture dans les images clés. Nous avons implémenté la méthode de la matrice de cooccurrence à niveau de gris pour extraire les indices de la texture, car elle est très utilisée en recherche d'images, et en général donne de bons résultats [117]. Les indices de la texture que nous avons extraite de l'image clé de chaque vidéo sont la moyenne, la variance, l'énergie, la corrélation, l'entropie, le contraste, l'homogénéité, le *cluster shade* et le *cluster prominence.*

   Le calcul de la matrice de cooccurrence nécessite le choix d'une distance et d'un angle de déplacement. Il a été noté par plusieurs chercheurs [118, 119] que la distance d'un pixel combinée avec des angles respectifs de 0°, 45°, 90° et 135° donne de bons résultats. C'est la solution que nous avons adoptée, ce qui nous

donne à la fin quatre matrices de cooccurrence pour chaque image clé. La deuxième difficulté à laquelle nous avons fait face est le nombre élevé de niveaux de gris différent. En effet, avec seulement 256 niveaux de gris différent, nous aurons des matrices de 256x256 cases chacune, ce qui les rend inutilisables à cause de leur taille élevée. Pour remédier à ce problème, nous avons sous-échantillonné les niveaux de gris en les regroupant dans 16 niveaux de gris différent seulement. Ci-dessous les formules que nous avons utilisées pour le calcul des indices de la texture, où *p(i,j)* correspond aux éléments de la matrice de cooccurrence.

a. La moyenne

$$f1 = \sum_i \sum_j p(i,j)$$

b. La variance

$$f2 = \sum_i \sum_j (i-\mu)^2 p(i,j)$$

où μ est la moyenne calculée ci-dessus.

c. L'énergie

$$f3 = \sum_i \sum_j p(i,j)^2$$

d. La corrélation

$$f4 = \sum_i \sum_j (i-\mu_x)(j-\mu_j)p(i,j)$$

où μ_x et μ_y sont respectivement les moyennes des lignes et des colonnes de la matrice de cooccurrence.

e. L'entropie

$$f5 = -\sum_i \sum_j p(i,j)\log p(i,j)$$

f. Le contraste

$$f6 = \sum_i \sum_j (i-j)^2 p(i,j)$$

g. L'homogénéité

$$f7 = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i.j)$$

h. Le *cluster shade*

$$f8 = \sum_i \sum_j (i+j-2\mu)^3 \, p(i,j)$$

où μ est la moyenne calculée avec la formule a présentée dans la page précédente.

i. Le *cluster prominence*

$$f9 = \sum_i \sum_j (i+j-2\mu)^4 \, p(i,j)$$

où μ est la moyenne calculée avec la formule a présentée dans la page précédente.

4. L'histogramme de la couleur aux alentours des points de contour :

Les caractéristiques de la couleur et de la texture ne donnent pas assez d'information sur la nature des objets à l'intérieur de l'image. En effet, deux images totalement différentes peuvent avoir des caractéristiques de couleur et de texture presque similaires. Pour remédier à ce manque, nous avons implémenté une caractéristique qui mélange la détection des contours et l'histogramme de la couleur afin de différencier entre les objets à l'intérieur de l'image clé. Voici les étapes de l'extraction de cette caractéristique à partir de l'image clé d'une vidéo :

a. Extraire les contours par la méthode du détecteur Gradient en utilisant le masque de Sobel. Les vidéos traitées n'ont pas tout le même degré de contraste : certaines sont très contrastées alors que d'autres sont très floues. Ceci nous a conduits à utiliser un seuil variable pour la détection des contours. Nous avons déterminé ce seuil de telle manière que le nombre des points de contours représente 10 % du total des points de l'image résultante. Ce pourcentage a été déterminé empiriquement. Ainsi, nous commençons par

extraire le contour avec un seuil de 0%, puis d'une manière itérative, nous augmentons le seuil de 1% à chaque fois afin de réduire le pourcentage des points de contours jusqu'à 10%.

b. Considérer l'entourage des points de contour. Pour ce faire, nous avons augmenté l'épaisseur du contour d'un pixel de chaque côté.

c. Calculer l'histogramme de la couleur dans l'espace RGB, en ne prenant en compte que les points retenus à l'étape b.

La figure 3.5 montre une illustration des trois étapes de l'extraction de cette caractéristique.

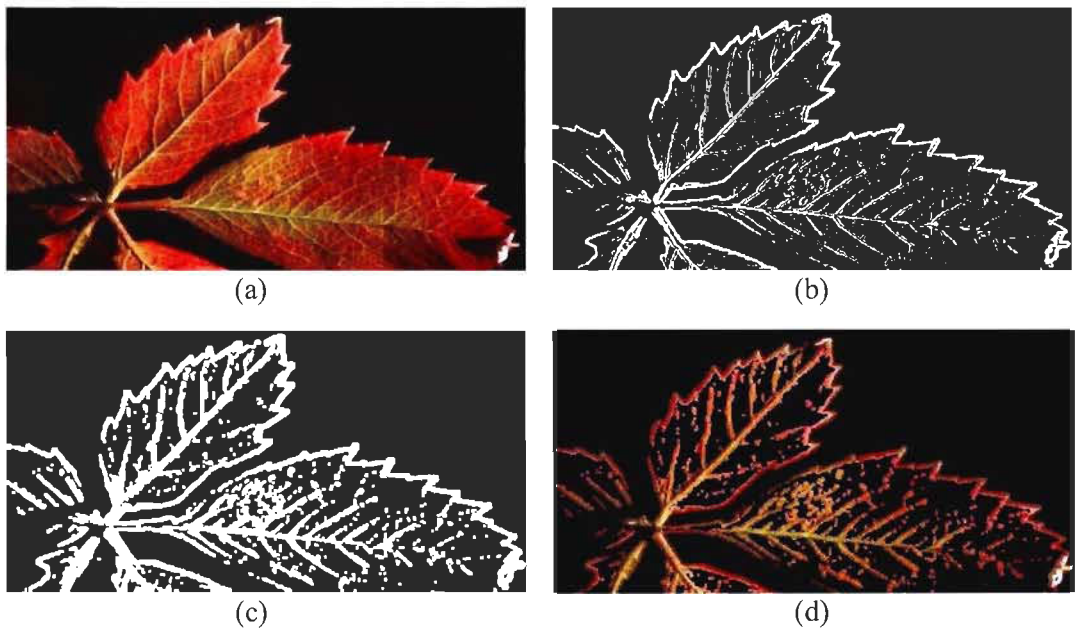


Figure 3.5 : Extraction de l'histogramme de la couleur au alentour des points de contour : (a) l'image originale, (b) l'image du contour, (c) l'augmentation du contour, (d) l'image considérée pour l'extraction de l'histogramme.

5. Le pourcentage des points de contour :

   Nous avons implémenté cette caractéristique pour essayer de refléter le nombre d'objets dans l'image clé. En effet, une image avec beaucoup de points de contour est susceptible de contenir beaucoup d'objets, et vice versa.

   Nous avons calculé cette caractéristique à partir de l'image clé de la manière suivante :

   a. Extraire le contour par la méthode du détecteur Gradient en utilisant le masque de Sobel. Nous avons fixé le seuil à 0.05 d'une façon empirique.
   b. Calculer le pourcentage des points de contour par rapport à la taille de l'image.

6. Le mouvement :

   La différence entre l'image et la vidéo est l'existence du mouvement. Ce mouvement peut résulter du mouvement de la caméra, du mouvement des objets à l'intérieur de la vidéo, ou d'une combinaison des deux. Donc, nous pouvons dire que le mouvement est une information spécifique et pertinente pour l'indexation de vidéo. Cela dit, notre objectif est d'extraire une caractéristique qui décrit le mouvement dans les vidéos personnelles. Ces dernières peuvent traiter des sujets différents et très variés. Par exemple, les vidéos peuvent être d'une fête d'anniversaire ou d'un mariage, d'une journée à la plage, d'un match de soccer dans le quartier ou d'un diner à la maison. Le mouvement dans ces genres de vidéo n'est pas cohérent. Essayer de le modéliser ou d'analyser la trajectoire de tous les objets à l'intérieure des vidéos, en vue d'extraire une caractéristique est une tache fastidieuse et couteuse en ressource machine. La solution que nous avons proposée et qui peut être appliquée à tous les genres de vidéo est de quantifier l'intensité du mouvement dans les vidéos. De cette manière, la caractéristique du mouvement nous informera sur la quantité du mouvement dans la vidéo.

Pour extraire la caractéristique du mouvement dans les vidéos, nous avons suivi l'algorithme suivant :

a. Diviser chaque image appartenant au plan en blocs de 64 x 64 points. Nous avons choisi la taille du bloc d'une façon empirique pour optimiser le temps de calcul et la pertinence des résultats de la recherche des vidéos.
b. Pour chaque bloc de chaque image, calculer la moyenne des couleurs RGB (chaque bande à part).
c. Définir une séquence de blocs comme étant l'ensemble de tous les blocs de toutes les images du plan, qui apparaissent à la même position.
d. Calculer, pour chaque séquence de blocs, la variance entre les moyennes calculées à l'étape b.
e. Calculer la moyenne des variances calculées à l'étape d.

Par conséquent, pour chaque plan nous avons extrait un descripteur de mouvement constitué de trois valeurs. Une valeur pour chaque bande de couleur.

3.2.1.5 Normalisation des données

Les systèmes de recherche de vidéos utilisent les distances entre les vecteurs de caractéristiques pour calculer la similarité entre les vidéos. Une étape importante entre l'extraction des caractéristiques et le calcul des distances est la normalisation des valeurs des caractéristiques. La normalisation est le processus de changement d'échelle, ou la redistribution des données pour améliorer la visualisation ou la comparaison. Cette normalisation est nécessaire, car l'ordre de grandeur et l'intervalle des valeurs des caractéristiques peuvent varier énormément d'une caractéristique à l'autre. Par conséquent, une mesure de distance comme la distance Euclidienne accorde implicitement plus de poids aux caractéristiques dont l'intervalle des valeurs est grand qu'à celles avec un petit intervalle.

Donc, notre but est de ramener les valeurs de toutes les caractéristiques dans un intervalle de [0,1]. Étant donné la limite inférieure $X_{\min}$ et la limite supérieure $X_{\max}$ des valeurs d'un vecteur caractéristique X, nous appliquons la transformation linéaire suivante pour normaliser ces valeurs :

$$x_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

où X_{norm} est le vecteur caractéristique normalisé dont l'intervalle des valeurs est [0, 1].

3.2.2 La formulation de la requête

Idéalement, un système d'indexation et de recherche de la vidéo doit permettre à son utilisateur d'exprimer sa requête d'une manière facile et implicite. Généralement, l'utilisateur peut rechercher dans la base de données des vidéos un personnage, un objet, un événement ou un texte à l'intérieur de la vidéo. La manière la plus implicite pour un utilisateur est d'exprimer sa requête par un texte qui définit le contenu des vidéos qu'il recherche. Par la suite, le système analyse la requête et traduit cela en caractéristiques qui décrivent la vidéo requête. Le problème qui se pose à ce niveau, c'est comment traduire une requête textuelle en un concept visuel? Par exemple : quelles sont les caractéristiques visuelles qui d'une fête de mariage ? Et comment ne pas confondre les caractéristiques d'une fête de mariage avec celles d'un autre type d'événement?

La deuxième manière pour un utilisateur d'exprimer sa requête, c'est de choisir une vidéo parmi les vidéos de la base de données. Après cela, le système analyse la vidéo requête, extrait les caractéristiques de cette dernière et rapporte toutes les vidéos qui lui ressemblent. C'est la solution que nous avons adoptée pour notre système d'indexation et de recherche de la vidéo.

3.2.3 La sélection des caractéristiques

Dans notre système, nous avons extrait une multitude de caractéristiques visuelles pour décrire les vidéos de la base de données. Nous avons utilisé ces caractéristiques lors de la comparaison afin de trouver les vidéos qui ressemblent à la requête de l'utilisateur. Puisqu'un de nos buts est de trouver la combinaison de caractéristiques qui donne les meilleurs résultats, nous avons permis à l'utilisateur de sélectionner les caractéristiques qui entrent dans la comparaison. L'utilisateur a également la possibilité de sélectionner une combinaison de caractéristiques prédéterminées, que nous avons sélectionnées d'une façon empirique. Ces caractéristiques sont :

- Les moments de la couleur d'ordre 1 et 2.
- L'histogramme de la couleur dans l'espace RGB et HSV.
- La texture en utilisant la matrice de cooccurrence (la moyenne, la variance, l'homogénéité et l'entropie).
- L'histogramme de la couleur aux alentours des points de contour.
- Le pourcentage des points de contour.
- Le mouvement.

Nous avons trouvé que la combinaison que nous proposons donne de bons résultats lors de la recherche. La figure 3.10 montre la boite de dialogue qui permet à l'utilisateur de l'outil que nous avons développé le choix des caractéristiques qui entre dans la comparaison.

3.2.4 La comparaison

C'est l'étape de comparaison entre les caractéristiques de la vidéo requête et les vidéos de la base de données. L'objectif est de trouver les vidéos qui ressemblent le plus à la vidéo requête, en utilisant une mesure de similarité adéquate. Étant donné que la distance Euclidienne est une mesure simple et très populaire, nous avons choisi d'utiliser comme mesure de similarité une somme pondérée de distances Euclidiennes entre les

caractéristiques déjà extraites. Même si le choix de la distance influence les résultats de la recherche, nous avons opté pour cette solution car notre travail ne couvre pas la comparaison entre les distances existantes.

Premièrement, nous avons calculé la distance Euclidienne entre chaque caractéristique de la vidéo requête et sa vis-à-vis de chaque vidéo de la base de données. La distance Euclidienne entre deux vecteurs V_1 et V_2 est calculée comme suit :

$$D(V_1, V_2) = \sqrt{\sum_{i=1}^{n} (V_1(i) - V_2(i))^2}$$

où n est la taille du vecteur caractéristique.

Deuxièmement, nous avons normalisé les distances calculées à l'étape précédente. Troisièmement, nous avons combiné ces distances dans une somme pondérée qui nous a donné la distance globale entre notre requête et chaque vidéo de la BD. Notons que les poids des caractéristiques ont été déterminés de façon empirique selon la méthode expliquée dans la section 4.5.2 du quatrième chapitre. Finalement, nous avons trié les vidéos de la BD de façon ascendante, et les vidéos ayant les distances les plus petites ont été retournées à l'usager.

Donc, en utilisant la combinaison des caractéristiques prédéterminées que nous avons trouvées empiriquement, la formule du calcul de la distance est comme suit :

$$D = \sum_{i=1}^{5} normD_i + 4(norm(D_6)) + 3(norm(D_7)) + \frac{2}{4}\left(norm\left(\sum_{i=8}^{11} D_i \right) \right)$$

où

- D_1 est la distance Euclidienne de la caractéristique « moment de la couleur d'ordre 1 »;
- D_2 est la distance Euclidienne de la caractéristique « moment de la couleur d'ordre 2 »;

- D_3 est la distance Euclidienne de la caractéristique « histogramme de la couleur RGB »;
- D_4 est la distance Euclidienne de la caractéristique « histogramme de la couleur HSV »;
- D_5 est la distance Euclidienne de la caractéristique « mouvement » ;
- D_6 est la distance Euclidienne de la caractéristique « histogramme de la couleur aux alentours des points de contour »;
- D_7 est la distance Euclidienne de la caractéristique « pourcentage des points de contour » ;
- D_8 est la distance Euclidienne de la caractéristique « moyenne de la matrice de cooccurrence »;
- D_9 est la distance Euclidienne de la caractéristique « variance de la matrice de cooccurrence »;
- D_{10} est la distance Euclidienne de la caractéristique « homogénéité de la matrice de cooccurrence »;
- D_{11} est la distance Euclidienne de la caractéristique « entropie de la matrice de cooccurrence ».

Maintenant que nous avons expliqué toutes les étapes de fonctionnement de l'outil que nous avons développé pour l'indexation et la recherche de la vidéo, nous allons montrer dans la section suivante le fonctionnement de l'interface de cet outil.

3.3 Fonctionnement de l'interface de recherche

Nous avons développé notre application suivant l'architecture et les étapes de fonctionnement du système d'indexation et de recherche de la vidéo que nous avons présentés précédemment. Nous avons utilisé le langage Matlab pour implémenter notre

système. L'avantage de Matlab par rapport à d'autres langages de programmation est qu'il permet le développement et l'exécution rapide des opérations matricielles.
L'utilisation de notre application est très simple. La figure 3.6 montre une prise d'écran de la fenêtre principale. Cette dernière se compose d'un menu « Administrateur », un menu « Recherche », un panneau pour afficher un échantillon de la base de données, ainsi que les résultats de la recherche et un panneau pour visualiser les vidéos.

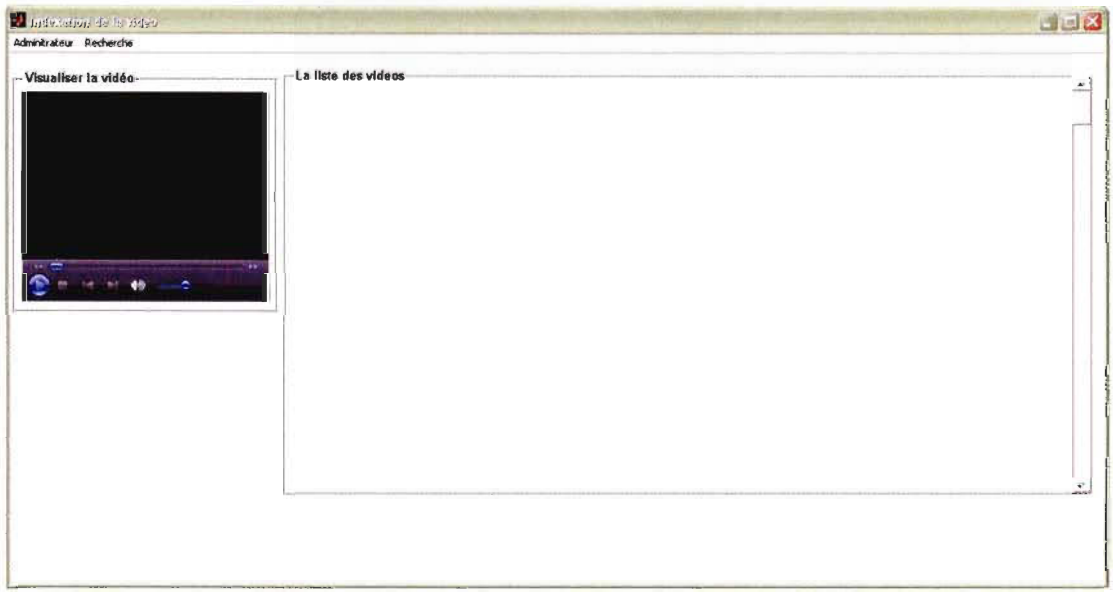


Figure 3.6 : La fenêtre principale du système d'indexation et de recherche de la vidéo.

Pour qu'un utilisateur puisse utiliser notre système, l'administrateur doit en premier faire le prétraitement de la BD en *offline*. Pour lancer le prétraitement ou la digestion d'une base de données, l'administrateur doit choisir le menu « Administrateur », puis sélectionner la commande « Calculer les caractéristiques ». Par la suite, il doit choisir le répertoire de la base de données de vidéos qu'il veut traiter, et le nom du fichier où les données du traitement seront sauvegardées. À la fin, il peut lancer la digestion de la BD.

La figure 3.7 montre une prise d'écran de la boite de dialogue qui permet le lancement du prétraitement.

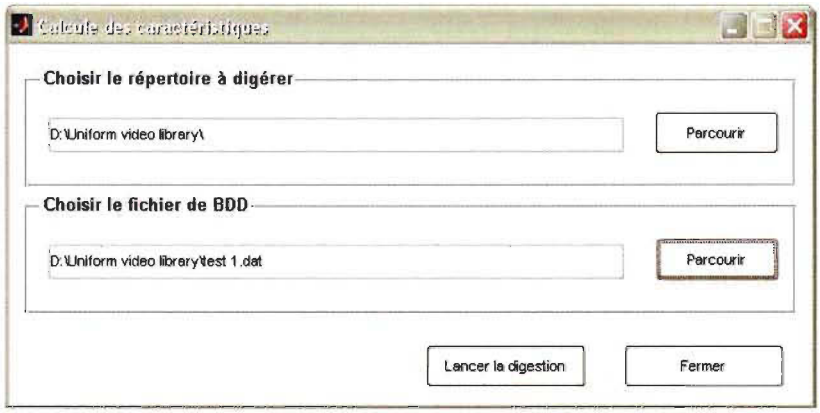


Figure 3.7 : La boite de dialogue qui permet le lancement de l'étape de prétraitement.

Pour faire une recherche, l'utilisateur de notre application doit en premier choisir la BD dans laquelle il veut effectuer des recherches. Cette BD doit être prétraitée préalablement. Pour ce faire, il doit choisir le menu « Recherche », puis sélectionner la commande « Initialiser ». La figure 3.8 montre une prise d'écran de la boite de dialogue qui permet l'initialisation du système via le choix du fichier où sont sauvegardées les données du prétraitement d'une base de données.

Figure 3.8 : L'initialisation de l'application avec les données d'une base de données déjà traitée.

Suite à l'initialisation du système, l'utilisateur peut voir un échantillon de la base de données dans le panneau « La liste des vidéos ». Dans le cas où il veut visualiser une vidéo, il lui suffit de double cliquer sur l'icône de la vidéo de son choix. La figure 3.9 montre une prise d'écran d'un échantillon d'une BD.

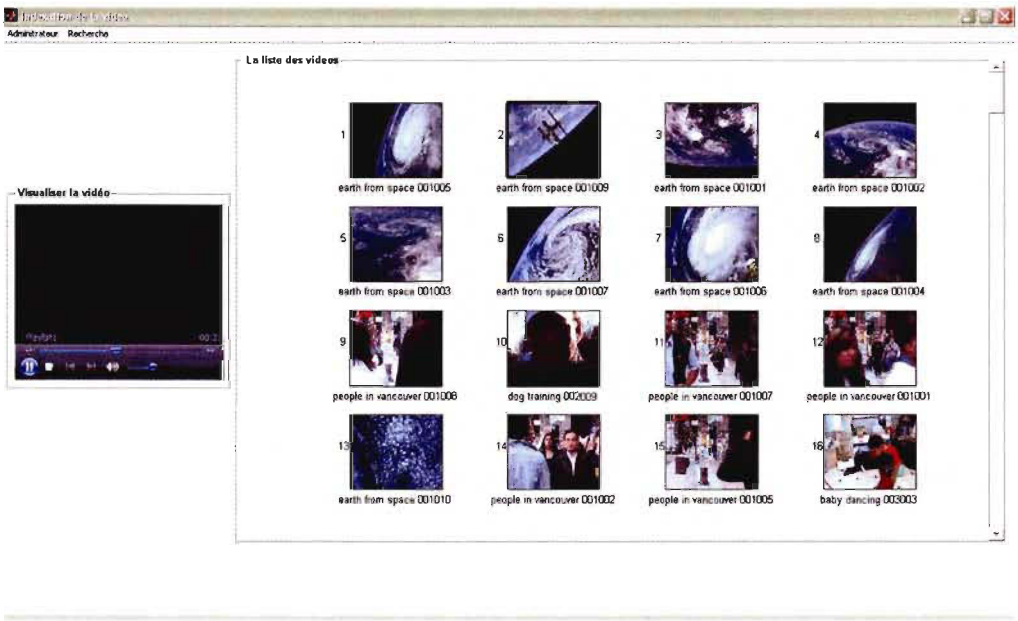


Figure 3.9 : Affichage d'un échantillon des vidéos de la base de données.

Si l'utilisateur veut lancer une recherche, il doit d'abord choisir une vidéo requête, en cliquant sur l'icône de la vidéo de son choix. Puis, il doit choisir le menu « Recherche », et sélectionner la commande « Rechercher ». Cela permet de lancer la boite de dialogue où l'utilisateur doit sélectionner les caractéristiques qui seront utilisées lors de la comparaison. Il a également la possibilité d'adopter l'ensemble de caractéristiques que nous avons sélectionnées, et ce, en cliquant sur le bouton « Mes caractéristiques ». D'après nos expériences, ces caractéristiques donnent de bons résultats. La figure 3.10 montre une prise d'écran de la boite de dialogue qui permet à l'utilisateur de choisir les caractéristiques.

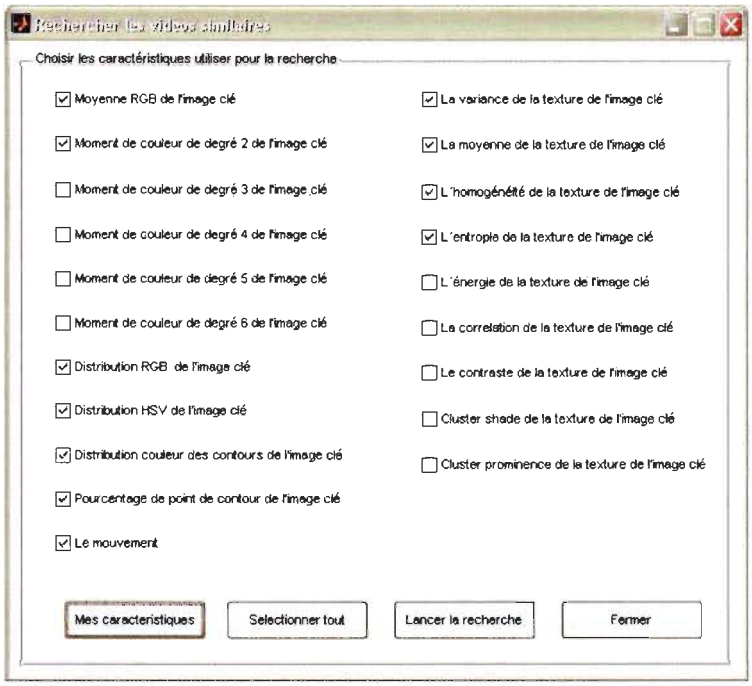


Figure 3.10 : La boite de dialogue qui permet le choix des caractéristiques.

Après la sélection des caractéristiques et le lancement de la recherche, les résultats s'affichent dans le panneau « La liste des vidéos ». Le numéro à côté des vidéos indique l'ordre des résultats. La figure 3.11 montre une prise d'écran des résultats obtenus suite à une interrogation de la BD.

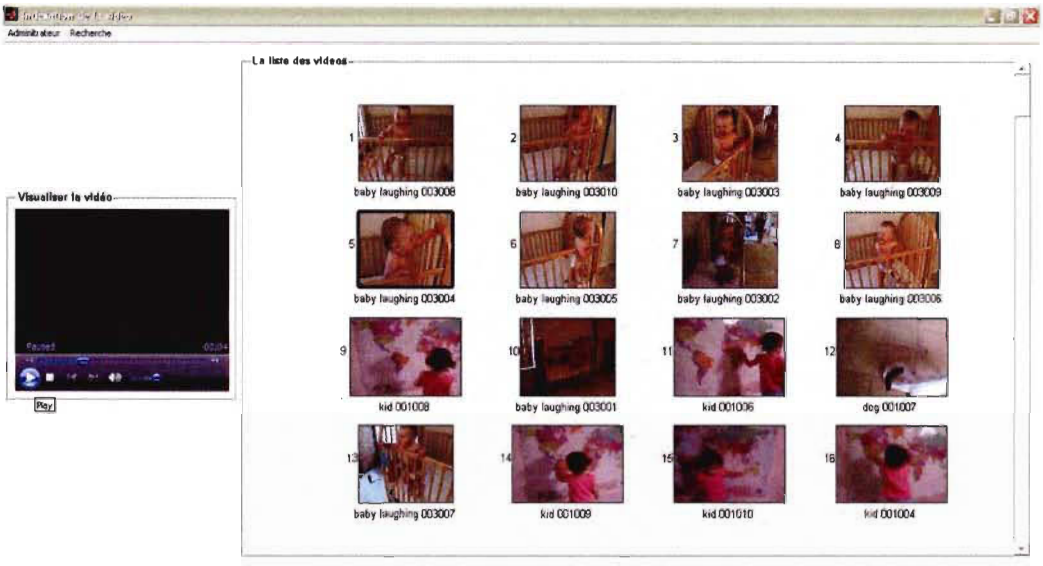


Figure 3.11 : La présentation des résultats après une recherche.

3.4 Conclusion

Le système d'indexation et de recherche des vidéos personnelles que nous avons développé est performant et efficace. Pour le développer, nous avons extrait des caractéristiques visuelles qui représentent bien le contenu de la vidéo. Ensuite, nous avons utilisé ces caractéristiques lors de la comparaison entre la vidéo requête et les vidéos de la base de données. Notons que l'extraction de l'image clé nous a permis de ramener le problème de la recherche de la vidéo en un problème de recherche d'images avec succès. Finalement, les résultats de la recherche sont présentés à l'utilisateur via une interface simple et conviviale.

Dans le chapitre suivant, nous allons présenter une évaluation détaillée de notre système.

Chapitre 4

Résultats expérimentaux

4.1 Introduction

Il y a diverses méthodes pour résoudre les problèmes liés à l'indexation et à la recherche de la vidéo par le contenu, chacune a ses avantages et ses inconvénients. C'est la raison pour laquelle il est indispensable de valider les résultats du système que nous avons conçu et implémenté sur une base de données relativement volumineuse, et qui reflète la réalité. Dans la section 4.2 de ce chapitre, nous allons exposer la base de données de vidéos que nous avons utilisée pour nos expérimentations. Ensuite, dans la section 4.3, nous allons aborder les indicateurs que nous avons utilisés pour mesurer la performance de notre outil, telle que la précision et le rappel. Les résultats des expériences sont présentés dans la section 4.4, et finalement, une conclusion à la section 4.5 termine le chapitre.

4.2 La base de données de vidéos

Le but de notre recherche est de créer un système d'indexation et de recherche de la vidéo destiné aux vidéos personnelles. Nous voulons dire par vidéo personnelle, toute collection de vidéos numériques archivée dans un support de sauvegarde pour un usage personnel. La source de ces vidéos peut être, par exemple, un appareil d'acquisition de vidéo numérique comme une caméra vidéo, une webcam ou un téléphone cellulaire muni de la fonction prise de vidéos. La particularité de ce genre de base de données est qu'elle contient des vidéos avec des sujets très variés. Par exemple, cela peut être une

vidéo d'une fête de mariage, des moments de vacances, des moments avec les membres de la famille, ou une capture d'un documentaire télévisé.

Afin de tester et d'évaluer les performances de notre application d'indexation et de recherche de la vidéo par le contenu, nous avons besoin d'une base de données de vidéos suffisamment grande, qui balaye les déférentes catégories de vidéos fréquemment rencontrées dans une BD personnelle. Pour cela, nous avons deux alternatives : avoir recours à une BD existante qui a été utilisée auparavant par un groupe de recherche, ou créer notre propre BD. Parmi les BD de vidéos existantes, nous pouvons citer :

- la bibliothèque libre de vidéos numériques (*Open Video Digital Library*) [136], développée à l'université de la Caroline du Nord à Chapel Hill. Elle contient 1800 vidéos, chacune d'une durée entre 1 et 10 minutes, totalisant 460 heures de vidéo.
- La BD de vidéos utilisée par la compétition TRECVID [137]. Elle contient 70 heures de vidéos, capturées de la chaine de télévision CNN et ABC par le consortium de données linguistiques (*Linguistic Data Consortium*).
- La BD de vidéos « MINERVA » créée par Wei Ren et al [138]. Elle contient 250 vidéos, chacune d'une durée de 2 minutes.

Nous avons décidé de ne pas utiliser ces bases de données, car aucune ne contient les catégories de vidéos que l'on rencontre dans une BD de vidéo personnelle. De plus, la seule BD que nous pouvons nous procurer est l'« *Open Video Digital Library* ». Par conséquent, il nous reste l'alternative de créer notre propre BD de vidéos personnelles, soit en produisant nos propres vidéos avec nos moyens, ou de collecter des vidéos gratuites des sites Internet. Nous avons opté pour la deuxième solution, parce que nous pouvons trouver sur Internet toutes sortes de vidéos gratuites, et ainsi avoir une BD de vidéos proche de la réalité.

Dans le prochain paragraphe, nous allons expliquer comment nous avons créé notre BD à partir de vidéos gratuites que nous avons collectées.

4.2.1 Collecte de données

Afin de créer notre BD de vidéos personnelles, nous avons commencé par sélectionner les catégories de vidéos que doit inclure la BD de test. En général, une personne veut archiver des vidéos des événements ou des sujets importants dans sa vie. Par exemple :

- Les fêtes : le mariage, l'anniversaire, le bureau, l'école, l'université, la garderie;
- La famille : la naissance, les activités d'enfants, le jardin, le BBQ, le diner, les animaux, les moments spéciaux;
- L'information : le documentaire et le vidéo blogue;
- La distraction : le concert, les blagues, le théâtre et le sport;
- Les vacances : le zoo, la plage, les montagnes et les endroits visités pendant les vacances;
- L'événement : la manifestation et le jour d'indépendance.

Après cela, nous avons collecté des vidéos gratuites qui appartiennent à chacune de ces catégories. Nous avons essayé d'avoir une collection de vidéos très variée qui reflète la réalité du terrain, en diversifiant les sujets des vidéos que nous avons collectés. Ainsi, la base de données collectée contient des vidéos liées aux différents sujets, qui soulignent différentes caractéristiques (la couleur, le mouvement, le nombre d'objets à l'intérieur des vidéos) et qui sont prises sous différentes conditions d'illuminations. En tout, nous avons collecté 30 vidéos, chacune d'une durée d'une vingtaine de minutes environ, qui couvrent les sujets suivants :

- six vidéos sur la famille;
- huit vidéos sur les animaux;
- quatre vidéos sur le sport;
- sept vidéos sur les fêtes;
- cinq vidéos sur d'autres sujets.

La figure 4.1 montre des icônes qui représentent les différentes familles de vidéos que nous avons utilisées.

Figure 4.1 : Images représentatives des vidéos de la base de données.

Nous allons maintenant aborder l'étape de la segmentation des vidéos.

4.2.2 Découpage des vidéos

Comme nous l'avons expliqué dans le troisième chapitre, nous avons segmenté nos vidéos en utilisant un logiciel d'édition. Dans notre cas, nous avons eu recours au logiciel « Adobe Première Pro». Cela est nécessaire pour réduire la complexité des vidéos. Ainsi, en utilisant ce logiciel, nous avons découpé les vidéos à l'endroit où il y a les caractéristiques d'un changement de plan, c.-à-d. les cas suivants :

- un changement brusque entre deux scènes;
- un effet d'édition de la vidéo, c.-à-d. un effet de transition entre deux plans;
- un mouvement de caméra avec un changement de l'arrière-plan.

Pour les besoins de nos tests, nous avons extrait 20 plans de chaque vidéo. Dans l'ensemble, la durée des plans varie entre 20 secondes et 2 minutes.

4.3 Vérité terrain

Afin d'évaluer les performances de notre système, il faut déterminer avec précision la pertinence des résultats de la recherche de chaque vidéo dans la base de données. C'est ce qu'on appelle la vérité terrain ou « *ground truth* ». Cependant, les critères des utilisateurs pour juger la pertinence des vidéos résultantes d'une recherche peuvent être très complexes en raison des caractéristiques spatio-temporelles compliquées des vidéos, des besoins divers et variables des utilisateurs, et de la possibilité que les utilisateurs interprètent différemment le contenu de la vidéo. Ainsi, les résultats d'une recherche peuvent être pertinents pour un utilisateur et pas pour un autre. Pour résoudre ce problème de concordance de jugement, nous avons classifié les vidéos de la BD manuellement. Pour ce faire, nous avons attribué un nom à chaque vidéo selon son contenu. Par exemple, nous avons attribué à toutes les vidéos qui contiennent des scènes d'une fête de mariage le nom « fête de mariage ». Dans le cas où il y'a plusieurs vidéos dans la même famille, nous avons ajouté un numéro de série aux noms des vidéos. Cela nous a permis de calculer automatiquement les mesures de performances. Si une vidéo

contient beaucoup d'objets et beaucoup de changement du contenu, la tâche de classification devient difficile, car cette vidéo peut appartenir à plusieurs familles en même temps. Par exemple, les vidéos de la famille « Aquarium », peuvent aussi appartenir à la famille « Poisson ». Donc, nous avons décidé de classifier chaque vidéo dans une seule famille afin d'éliminer ce problème. En tout, nous avons constitué 30 familles de vidéos, où chaque famille comprend 20 vidéos.

Dans le prochain paragraphe, nous allons présenter les mesures d'évaluation que nous avons utilisées dans le but de mesurer la performance de notre système.

4.4 Mesures d'évaluation

Afin d'évaluer la performance du système d'indexation et de recherche de la vidéo, nous devons définir les critères selon lesquels nous mesurons sa performance. Les mesures les plus courantes dans le domaine de la recherche de la vidéo et de l'image sont la précision et le rappel [139]. La précision (*Pr*) est la proportion des vidéos pertinentes dans l'ensemble des vidéos retournées comme résultat, c.-à-d. le rapport entre le nombre des vidéos pertinentes dans l'ensemble des vidéos trouvées et le nombre des vidéos trouvées. Le rappel (*Re*) est la proportion des vidéos pertinentes retournées comme résultat, c.-à-d. le rapport entre le nombre de vidéos pertinentes dans l'ensemble des vidéos trouvées et le nombre de vidéos pertinentes dans la base de données. Les formules pour calculer la précision et le rappel sont :

$$Pr = \frac{Ra}{A} \qquad\qquad Re = \frac{Ra}{R}$$

où :

- Ra est le nombre des vidéos pertinentes dans l'ensemble des vidéos retournées comme résultat;
- A est le nombre des vidéos retournées comme résultat;
- R est le nombre des vidéos pertinentes dans la base de données.

La figure 4.2 montre une illustration de la précision et du rappel.

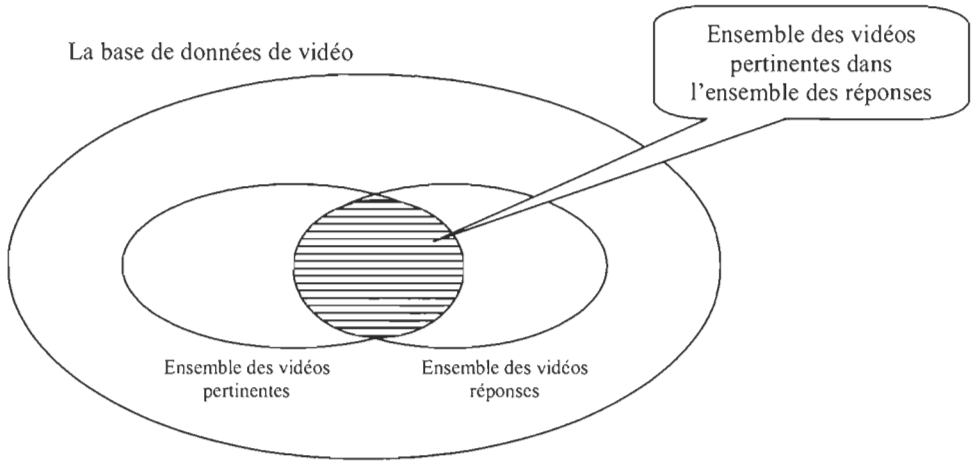


Figure 4.2: La précision et le rappel pour une recherche.

Les mesures de la précision et du rappel sont reliées entre elles, c'est pourquoi pour décrire la performance d'un système, on donne souvent une distribution du rappel et de la précision sous forme d'une courbe. Cependant, dans le domaine de la recherche de la vidéo et de l'image, le rappel dépend fortement du choix du nombre de vidéos retournées à l'utilisateur. Dans le cas où le nombre de vidéos pertinentes dans la BD est plus petit que le nombre de vidéos retournées à l'utilisateur, le rappel sera pénalisé. Pour remédier à cette lacune, nous avons opté pour la courbe précision-scope $Pr=f(Sc)$ [63], où le *Scope* (*Sc*) représente le nombre de vidéos retournées à l'utilisateur comme résultat de la recherche. En d'autres termes, la courbe $Pr=f(Sc)$ représente la précision du résultat de la recherche pour les différentes valeurs des nombres de vidéos retournées à l'utilisateur.

4.5 Expériences et évaluation

Nous avons effectué quatre expériences, chacune essaye de mesurer un aspect donné de notre système. Le but de la première est de mesurer la pertinence des images clés extraites de chaque vidéo. Dans la deuxième expérience, nous avons comparé les

résultats de la recherche en utilisant les différentes combinaisons de caractéristiques et les différents poids de pondération. Dans la troisième expérience, nous avons évalué les résultats de la recherche versus la classification que nous avons adoptée en utilisant la combinaison de caractéristiques et les poids que nous avons sélectionnés (chapitre 3, section 3.2.4). La dernière expérience consiste à valider nos résultats en demandant à des utilisateurs d'évaluer les résultats de la recherche. Nous avons obtenu les résultats de la deuxième et la troisième expérience automatiquement, en comparant le nom de la vidéo requête avec les noms des résultats. Nous avons considéré les résultats comme pertinents s'ils appartiennent à la même famille que la vidéo requête. Dans le cas où le contenu des vidéos est similaire, mais ces vidéos n'appartiennent pas à la même famille, nous avons considéré les résultats comme non pertinents. Cette méthode pénalise les résultats des expériences, mais nous a permis de faire des tests sur toutes les vidéos de la base de données.

4.5.1 Première expérience

L'objectif de cette expérience est d'évaluer la pertinence des images clés extraites de chaque vidéo. Pour ce faire, nous avons demandé à six utilisateurs de sélectionner au hasard 30 vidéos. À chaque fois, ils doivent évaluer si l'image clé générée par notre système est représentative de la vidéo sélectionnée. L'image clé peut être bonne, acceptable ou mauvaise. Il faut noter que cette évaluation reste très subjective, car elle est relative à l'interprétation du sujet de la vidéo par l'utilisateur. La figure 4.3 ci-dessous montre la moyenne des résultats de l'évaluation.

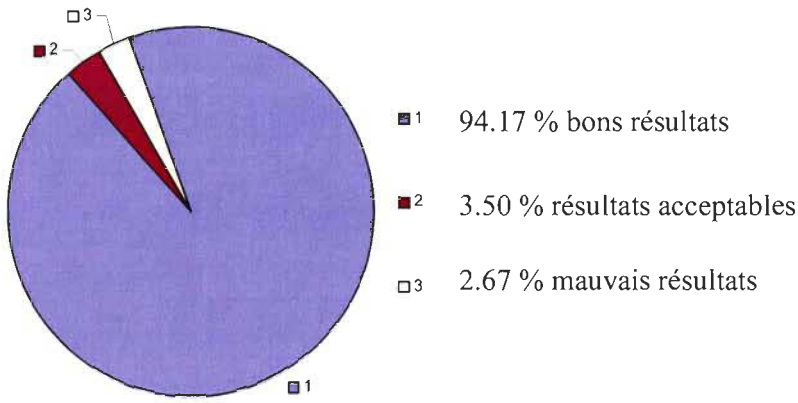


Figure 4.3 : Évaluation des images clés.

À partir de ces évaluations, nous pouvons dire que 95.9 % des images clés extraites des plans sont bonnes. Cependant, il faut noter que cela dépend du nombre d'objets et de la quantité de mouvement dans le plan.

4.5.2 Deuxième expérience

L'objectif de cette expérience est de trouver la bonne combinaison de caractéristiques parmi celles que nous avons implémentées, et les poids de pondération à affecter à chacune, lors du calcul de la distance entre les caractéristiques de la vidéo requête et les vidéos de la base de données.

Dans un premier temps, nous avons commencé par rechercher la bonne combinaison de caractéristiques sans se soucier des poids de pondération. Nous avons essayé différentes combinaisons de caractéristiques. Notons que nous nous sommes limités aux combinaisons qui semblent donner de bons résultats, vu que le nombre total de combinaisons est très élevé : pour vingt caractéristiques, il y'a 2^{20}-1 combinaisons possibles! Le choix de la bonne combinaison de caractéristiques peut également être fait de façon automatique. Pour ce faire, il suffit de calculer la précision moyenne de chaque

combinaison puis de choisir celle qui donne la précision moyenne la plus élevée. Cependant, si le calcul pour chaque combinaison prend environ 30 secondes, nous avons besoin d'environ 364 jours pour faire le choix automatique de la bonne combinaison.

Dans un second temps, nous avons recherché les bons poids de pondération en augmentant les poids à chaque fois de 0.5. Il faut noter que nous avons utilisé notre programme de comparaison automatique pour avoir les résultats de recherche de chaque vidéo dans la base de données.

Dans ce qui suit, nous allons présenter les différents tests et leurs résultats, en utilisant la courbe précision-scope *Pr=f(Sc)*. Pour chaque valeur de scope (de 1 jusqu'à 20), nous avons calculé la moyenne du pourcentage de bons résultats de chaque famille de vidéos dans la base de données. Par la suite, nous avons calculé la moyenne générale du pourcentage de bons résultats par rapport au scope de toutes les vidéos dans la base de données. Nous présentons pour chaque test la courbe précision-scope *Pr=f(Sc)* de cette moyenne.

a. Test 1 : faire une recherche en utilisant seulement les moments de la couleur dans le calcul de la distance. Nous avons testé les combinaisons suivantes :
    1. Test 1.1 : le moment d'ordre un;
    2. Test 1.2 : le moment d'ordre deux;
    3. Test 1.3 : le moment d'ordre trois;
    4. Test 1.4 : le moment d'ordre quatre;
    5. Test 1.5 : le moment d'ordre un et deux;
    6. Test 1.6 : le moment d'ordre un et trois;
    7. Test 1.7 : le moment d'ordre un, deux et trois;
    8. Test 1.8 : le moment d'ordre un, deux et quatre;
    9. Test 1.9 : le moment d'ordre un, deux et cinq;
    10. Test 1.10 : le moment d'ordre un, deux et six.

    Les figures 4.4 et 4.5 montrent les résultats de ces tests.

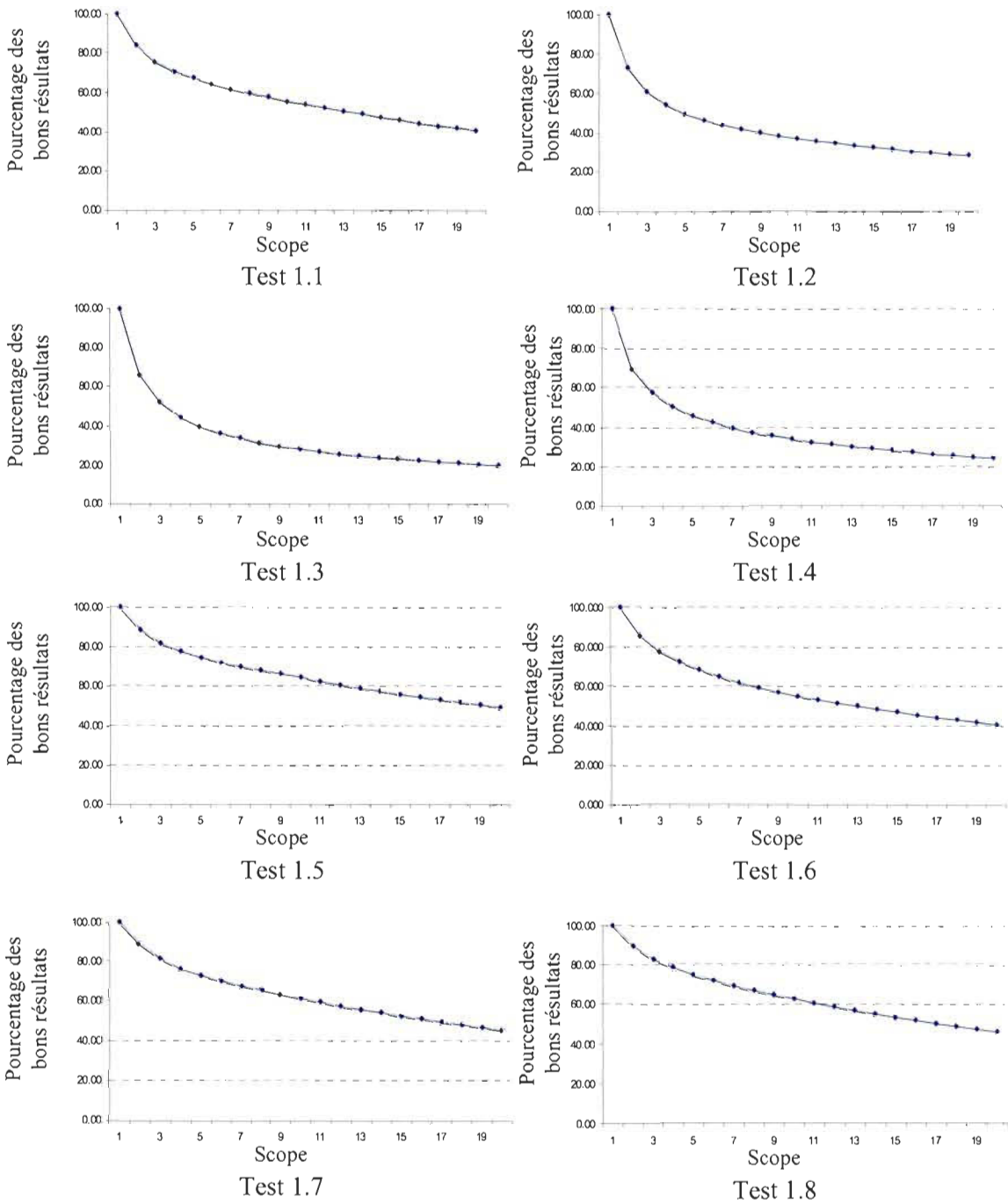


Figure 4.4 : Résultats de la recherche versus le scope du test 1.

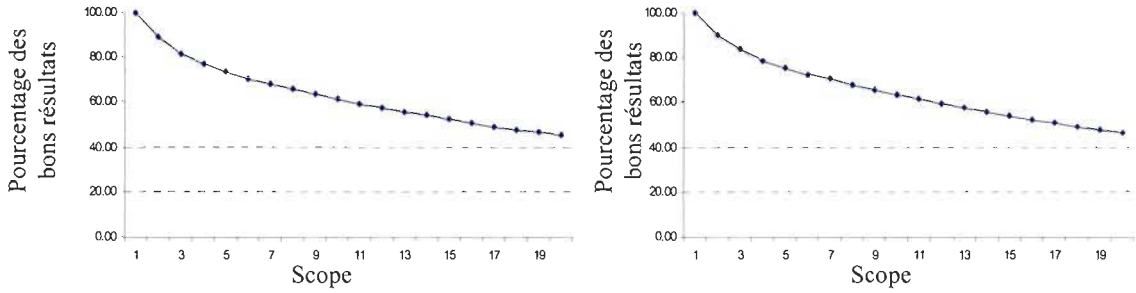


Figure 4.5 : Résultats de la recherche versus le scope du test 1 (suite).

b. Test 2 : nous avons testé les combinaisons de caractéristiques suivantes :
    1. Test 2.1 : l'histogramme RGB;
    2. Test 2.2 : l'histogramme HSV;
    3. Test 2.3 : l'histogramme de la couleur aux alentours des points de contour;
    4. Test 2.4 : le pourcentage des points de contour;
    5. Test 2.5 : le mouvement;
    6. Test 2.6 : l'histogramme RGB et HSV;
    7. Test 2.7 : l'histogramme RGB et aux alentours des points de contour ;
    8. Test 2.8 : l'histogramme HSV et aux alentours des points de contour ;
    9. Test 2.9 : l'histogramme RGB, HSV et aux alentours des points de contour;
    10. Test 2.10 : l'histogramme RGB, HSV, aux alentours des points de contour et le mouvement;

Les figures 4.6 et 4.7 montrent les résultats de ces tests.

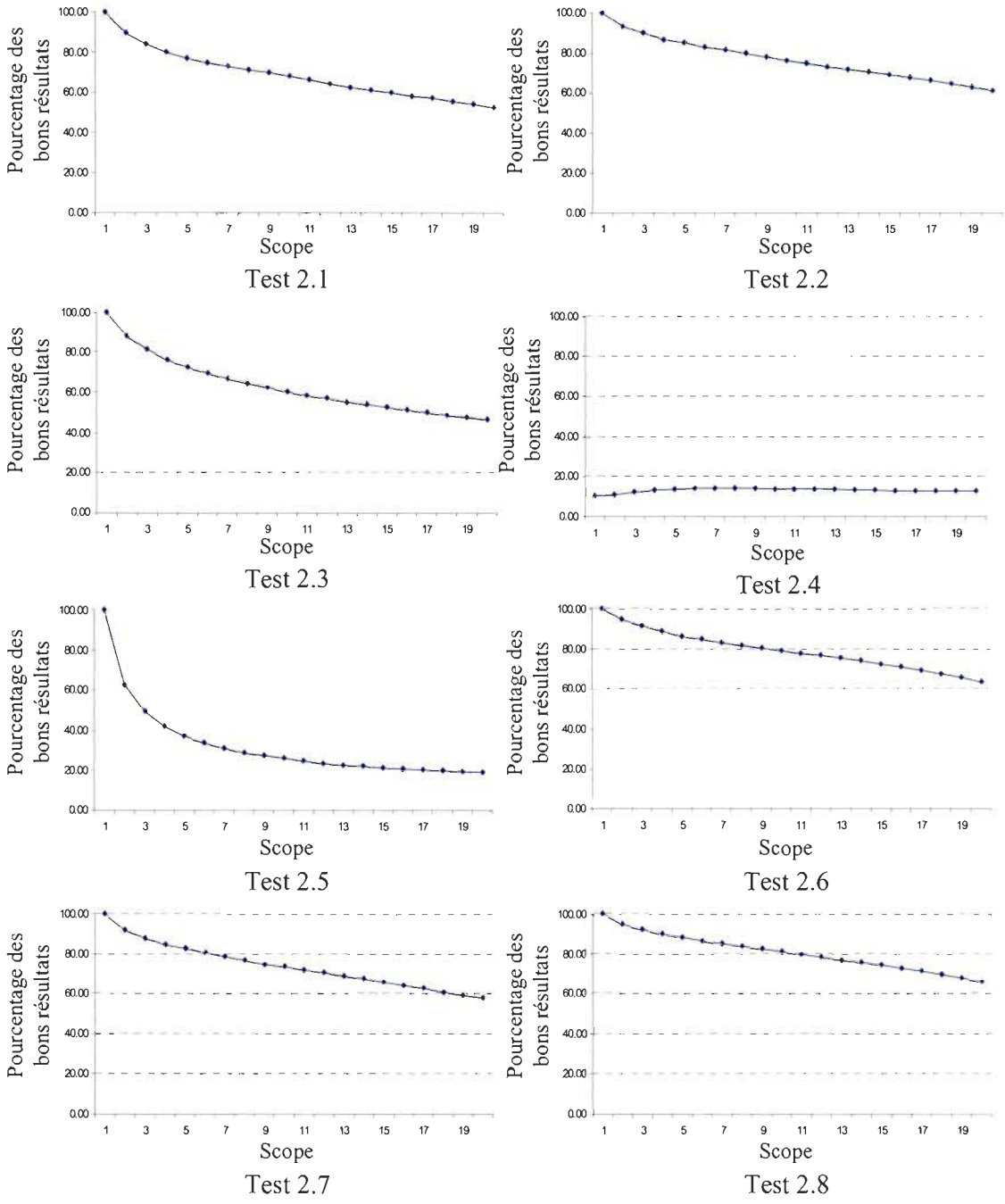


Figure 4.6 : Résultats de la recherche versus le scope du test 2.

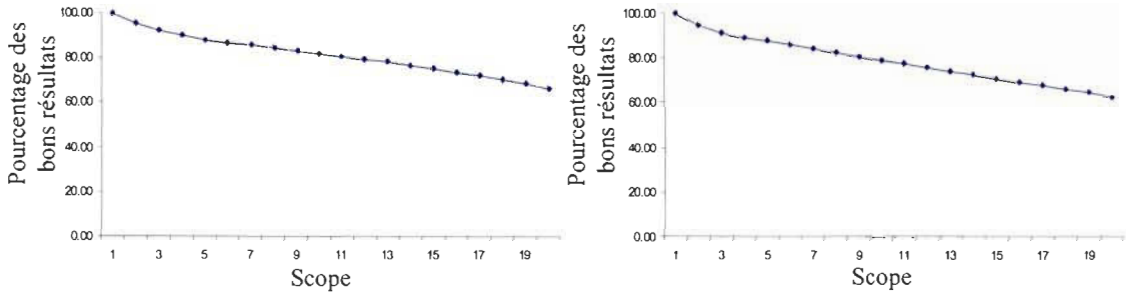


Figure 4.7 : Résultats de la recherche versus le scope du test 2 (suite).

c. Test 3 : faire une recherche en utilisant seulement les caractéristiques ci-dessous pour le calcul de la distance. Nous avons testé les combinaisons des caractéristiques de la texture suivantes :
    1. Test 3.1 : l'indice de moyenne de la texture;
    2. Test 3.2 : l'indice de variance de la texture;
    3. Test 3.3 : l'indice de l'énergie de la texture;
    4. Test 3.4 : l'indice de corrélation de la texture;
    5. Test 3.5 : l'indice de l'entropie de la texture;
    6. Test 3.6 : l'indice de contraste de la texture;
    7. Test 3.7 : l'indice de l'homogénéité de la texture;
    8. Test 3.8 : l'indice *cluster shade* de la texture;
    9. Test 3.9 : l'indice *cluster prominence* de la texture;
    10. Test 3.10 : l'indice de moyenne et de variance de la texture;
    11. Test 3.11 : l'indice de moyenne, de variance et d'énergie de la texture;
    12. Test 3.12 : l'indice de moyenne, de variance et d'homogénéité de la texture;
    13. Test 3.13 : l'indice de moyenne, de variance et d'entropie de la texture;
    14. Test 3.14 : l'indice de moyenne, de variance, d'homogénéité et d'entropie de la texture;

15. Test 3.15 : l'indice de moyenne, de variance, d'énergie et d'entropie de la texture;
16. Test 3.16 : l'indice de moyenne, de variance, d'entropie et de corrélation de la texture.

Les figures 4.8, 4.9 et 4.10 montrent les résultats de ces tests.

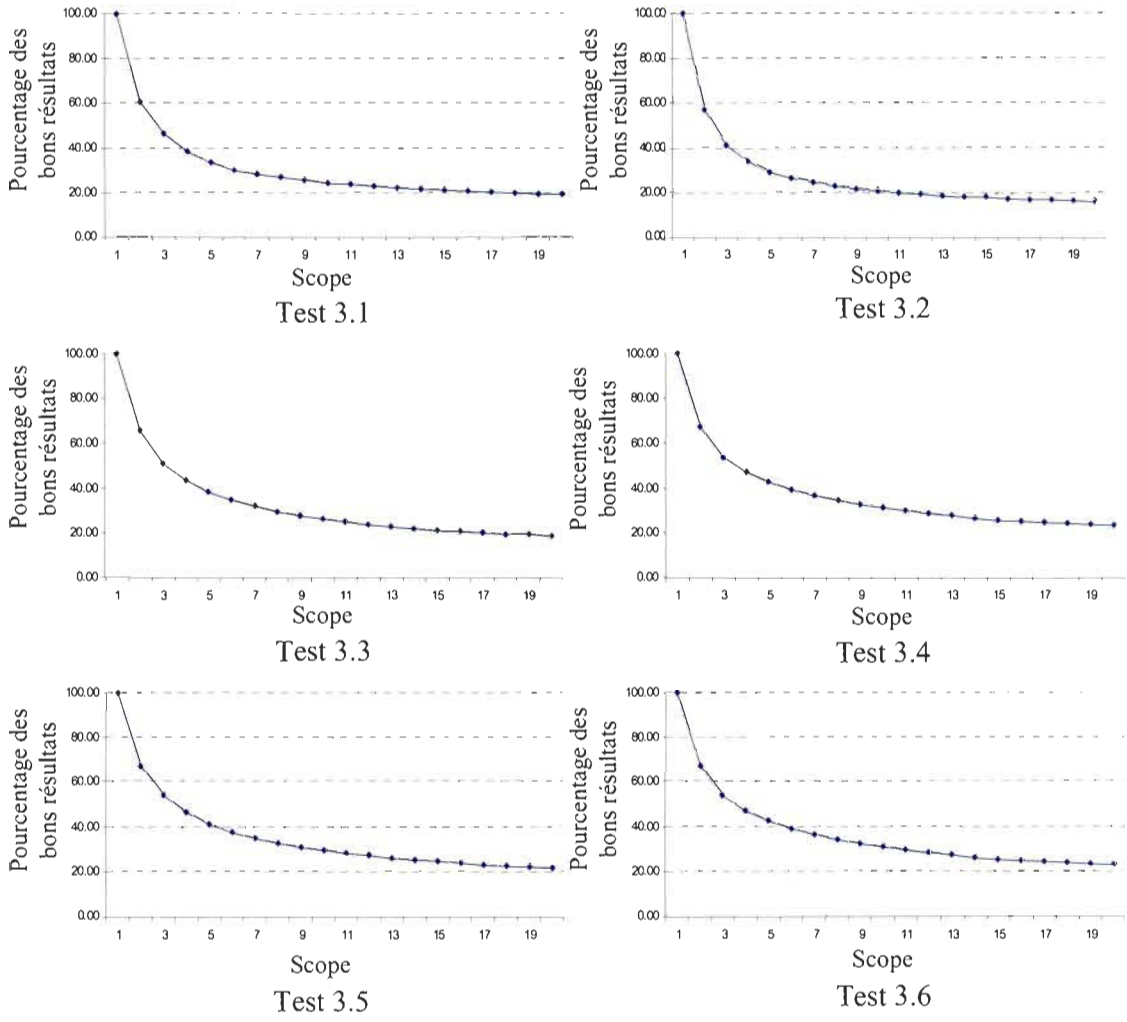


Figure 4.8 : Résultats de la recherche versus le scope du test 3.

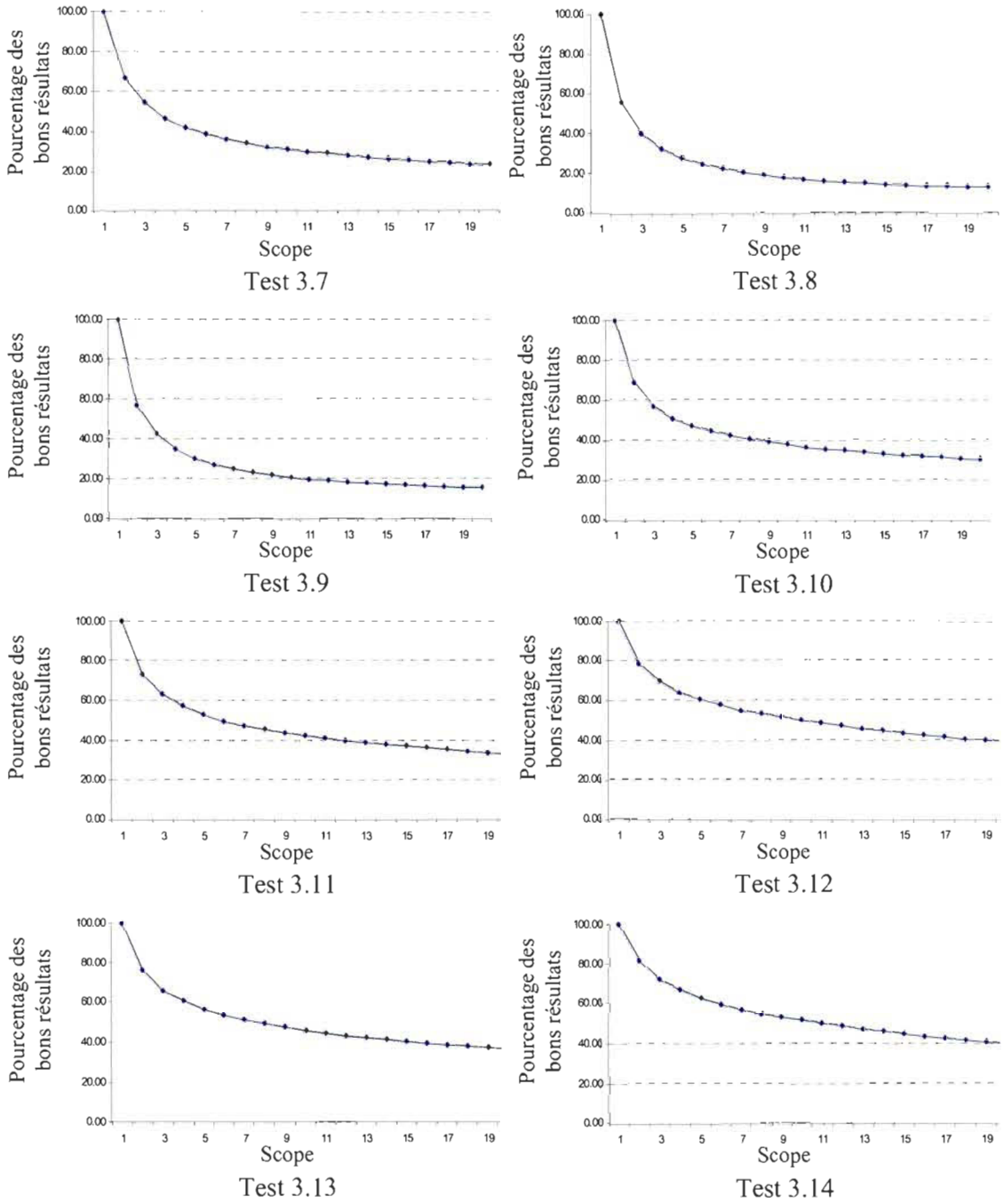


Figure 4.9 : Résultats de la recherche versus le scope du test 3 (suite).

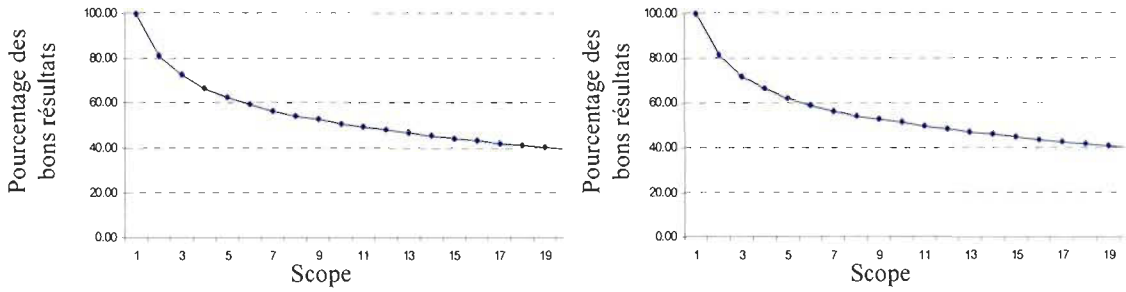


Figure 4.10 : Résultats de la recherche versus le scope du test 3 (suite).

d. Test 4 : faire une recherche en utilisant les combinaisons de caractéristiques avec des poids de pondération pour le calcul de la distance. Voir les détails dans le dans le tableau de la page suivante. Les figures 4.11, 4.12, 4.13 et 4.14 montrent les résultats de ces tests.

Tableau I

La combinaison des caractéristiques utilisées dans la deuxième expérience, test4.

Numéro de test	Les poids de pondérations																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Moment d'ordre 1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1/16	1	1	1	1
Moment d'ordre 2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1/2	1/16	1/16	1/16	1/16	1/16	1/16
Histogramme RGB	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1/16	1	1	1
Histogramme HSV	1	2	2	2	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1/16	1	3
Histogramme de la couleur au alentours des points de contours	4	4	3	3	4	4	4	4	4	4	4	5	5	4	4	4	4	4	4	4	4	4	4	4	4
Pourcentage des points de contours	3	3	2	2	3	3	3	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3
Mouvement	1	1	1	2	2	3	2	1	1	1	1	1	1	1	1	1	1/2	1/16	1/16	1/16	1/16	1/16	1/16	1/32	1/16
Moyenne de la texture	1/2	1/2	1/2	1/2	1/2	1/2	1	1	1	1	1	1	1	1/2	1/8	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/32
Variance de la texture	1/2	1/2	1/2	1/2	1/2	1/2	1	1	1	1	1	1	1	1/2	1/8	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/32
Homogénéité de la texture	1/2	1/2	1/2	1/2	1/2	1/2	1	1	1	1	1	1	1	1/2	1/8	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/32
Entropie de la texture	1/2	1/2	1/2	1/2	1/2	1/2	1	1	1	1	1	1	1	1/2	1/8	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/32

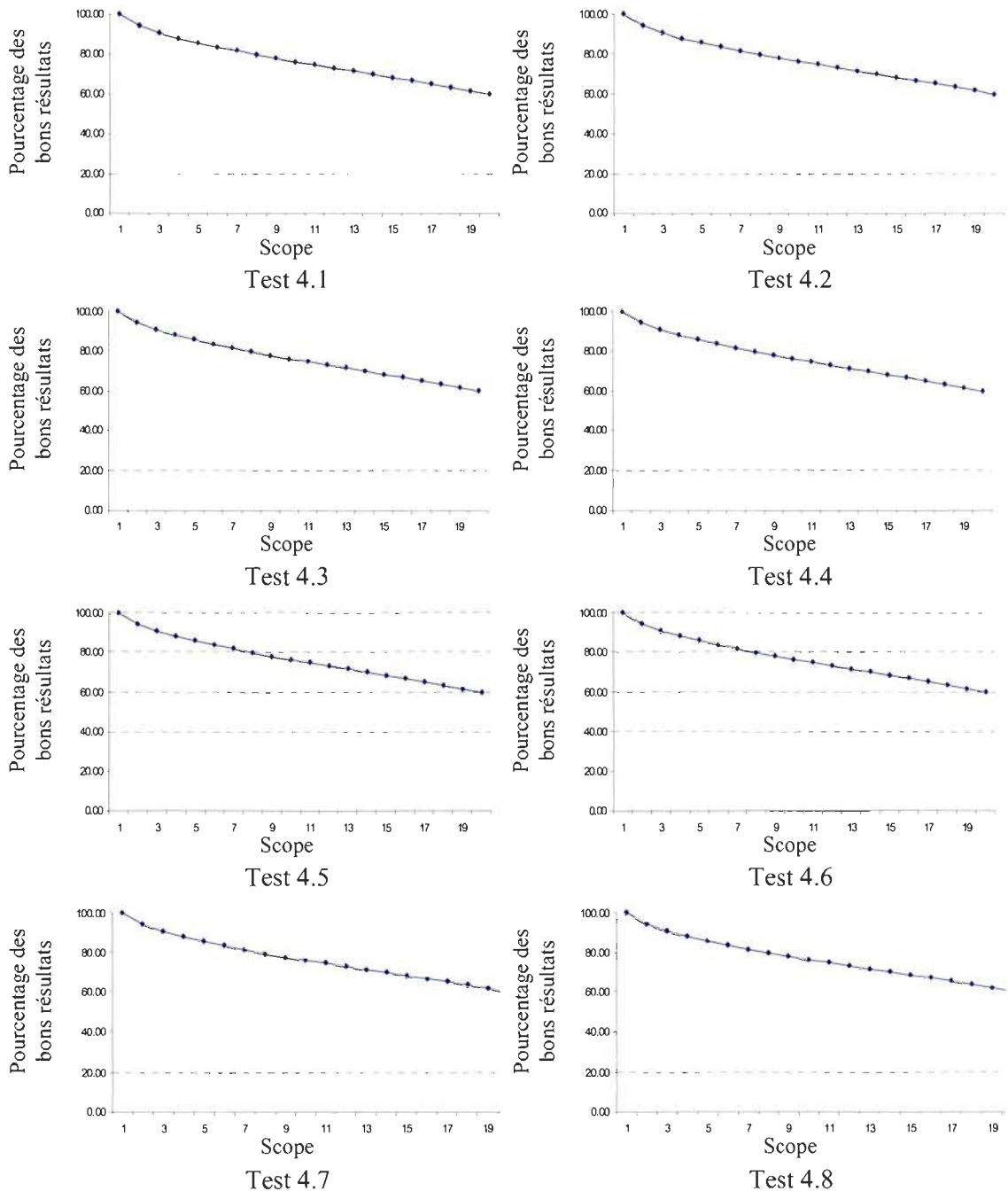


Figure 4.11 : Résultats de la recherche versus le scope du test 4.

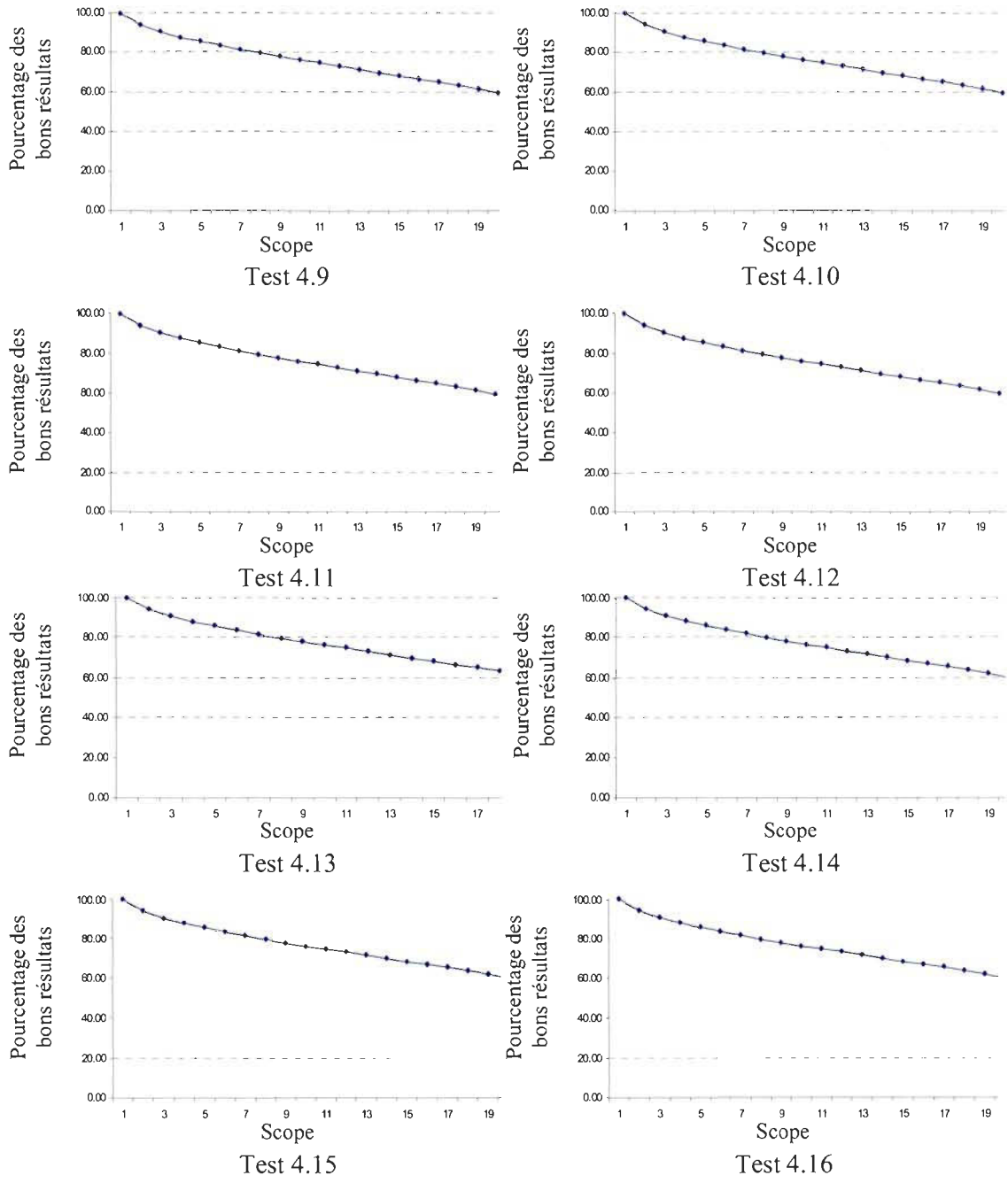


Figure 4.12 : Résultats de la recherche versus le scope du test 4 (suite).

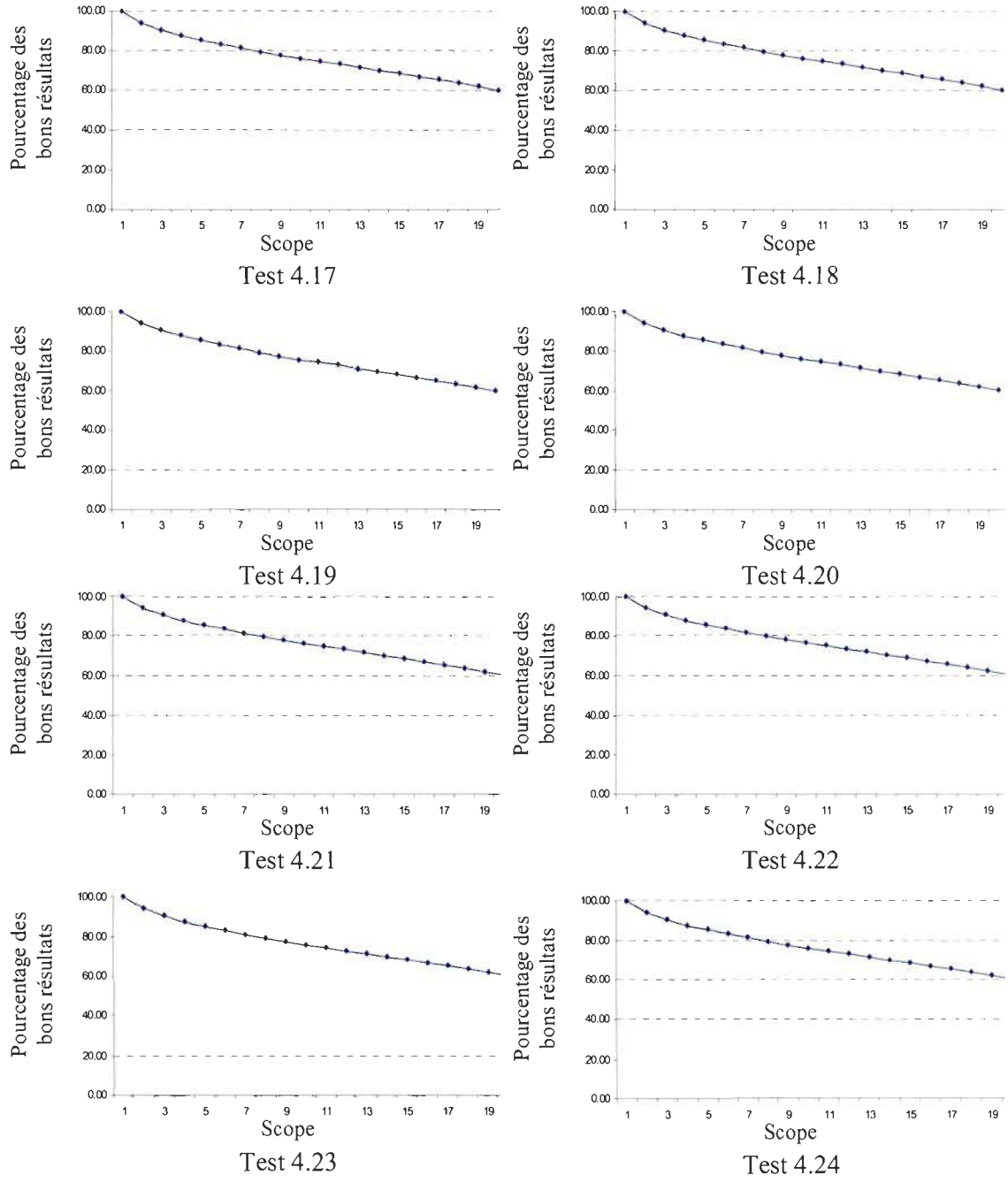


Figure 4.13 : Résultats de la recherche versus le scope du test 4 (suite).

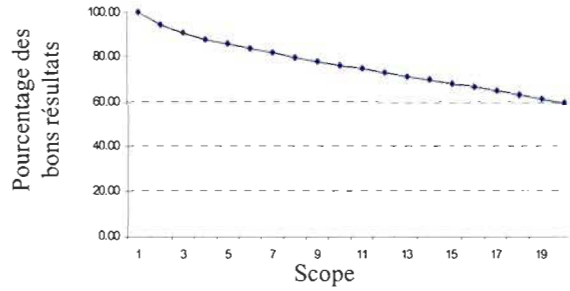


Figure 4.14 : Résultats de la recherche versus le scope du test 4 (suite).

Afin de trouver la bonne combinaison de caractéristiques, nous avons procédé par étape. En premier, nous avons combiné seulement les moments de la couleur de différents ordres. Deuxièmement, nous avons combiné les différents histogrammes, le pourcentage des points de contour et le mouvement, et finalement les caractéristiques de la texture. Dans la quatrième étape, nous avons testé différentes combinaisons de caractéristiques et différents poids de pondération.

D'après les résultats du premier test, nous avons trouvé que l'utilisation des moments d'ordre un et deux seulement (test 1.5) donne de bons résultats. La courbe de la moyenne des résultats reste plutôt haute et ne descend pas rapidement. À titre d'exemple, même rendue à vingt résultats (scope égal à 20), la précision est de 49.2 %. Nous jugeons que c'est un résultat acceptable pour des caractéristiques qui ne représentent pas assez le contenu de l'image. Dans le deuxième test, nous avons trouvé que l'utilisation des histogrammes RGB et HSV combinés avec l'histogramme de la couleur aux alentours des points de contour (test 2.9) donne de bons résultats. La courbe de la moyenne des résultats reste assez haute et ne descend pas rapidement. Par exemple, même rendue à vingt résultats (scope égal à 20), la précision est assez haute : 66.43 %. L'ajout de la caractéristique du mouvement à la combinaison précédente (test 2.10) détériore un peu les résultats par rapport au test 2.9. La courbe se comporte presque de

la même manière que le test 2.9, mais les résultats sont descendus à 62.6 % pour un scope de vingt. La différence est que ce n'est pas toutes les familles de vidéos qui se comportent de la même façon. Les résultats sont meilleurs pour les familles où il y a beaucoup de changements dans le contenu de la vidéo et beaucoup de mouvements comme dans la famille « Anniversaire ». D'après les résultats du quatrième test qui combine seulement les caractéristiques de la texture, nous avons trouvé que l'utilisation de la moyenne, de la variance, de l'homogénéité et de l'entropie est la meilleure combinaison. Rendue à 20 résultats (scope égal à 20), la précision est de 40 %. Cependant, l'utilisation des caractéristiques de la texture toutes seules ne donne pas de résultats satisfaisants. Les résultats des tests des différents poids de pondération nous montrent que la combinaison de caractéristiques et les poids de pondération que nous avons sélectionnés (test 4.1) donnent de bons résultats. La courbe reste assez haute et ne descend pas rapidement. À titre d'exemple, même rendue à vingt résultats (scope égal à 20), la précision est bonne : 60.2 %. L'analyse détaillée des courbes de chaque famille de vidéo nous montre que cette combinaison de caractéristiques apporte un bon compromis entre les résultats des différentes familles de vidéo. Par rapport au test 2.9, les résultats sont meilleurs pour les familles où il y a beaucoup de changements dans le contenu de la vidéo et beaucoup de mouvements comme dans la famille « Anniversaire ». Les résultats sont aussi meilleurs pour les familles où il y a beaucoup de textures comme dans la famille « Espace ».

4.5.3 Troisième expérience

L'objectif de cette expérience est d'évaluer les résultats de la recherche des vidéos versus la classification que nous avons adoptée en utilisant la combinaison que nous avons sélectionnée (chapitre 3, section 3.2.4). Pour ce faire, nous avons utilisé le programme de comparaison automatique pour avoir les résultats de recherche de chaque vidéo dans la base de données. Par la suite, nous avons calculé la moyenne du pourcentage des bons résultats versus le scope de chaque famille de vidéo. Nous

présentons dans les figures 4.15, 4.16, 4,17 et 4.18 les courbes précision-scope *Pr=f(Sc)* de chaque famille de vidéo dans la base de données. Nous avons aussi calculé la moyenne générale du pourcentage des bons résultats par rapport au scope de toutes les vidéos. Dans la figure 4.19 nous présentons la courbe précision-scope *Pr=f(Sc)* de cette moyenne.

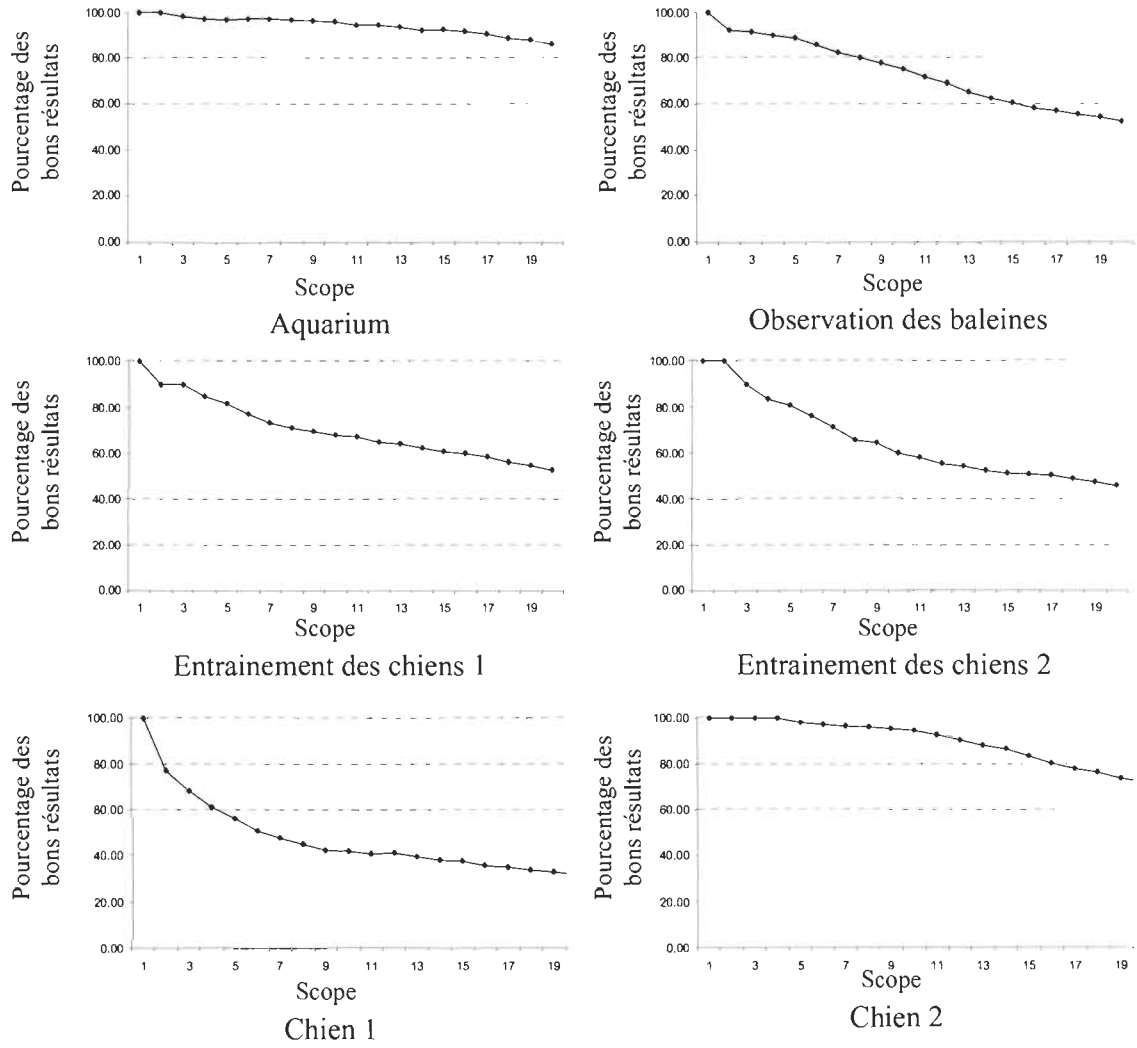


Figure 4.15 : Résultats de la recherche versus le scope des familles de vidéo.

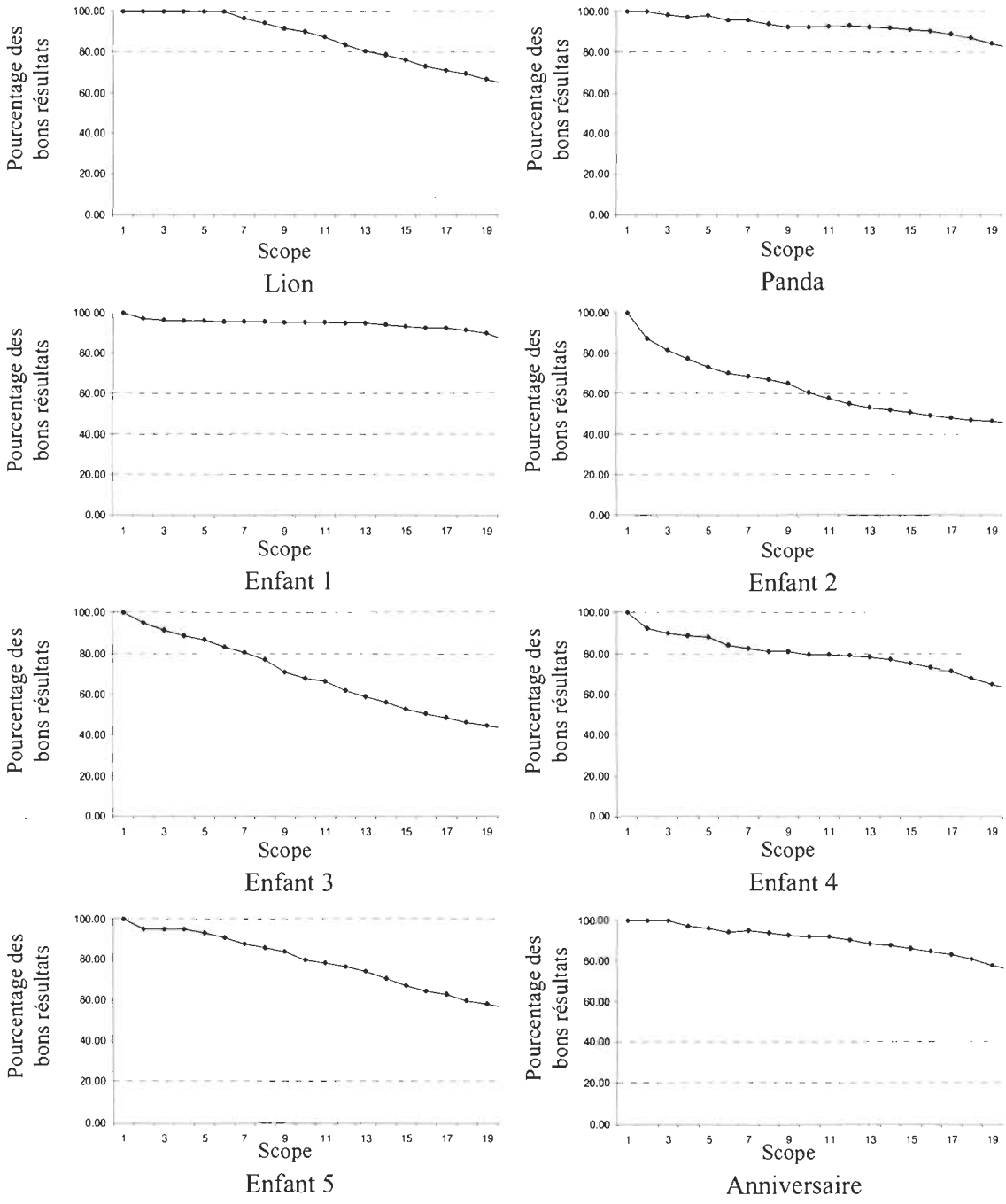


Figure 4.16 : Résultats de la recherche versus le scope des familles de vidéo (suite).

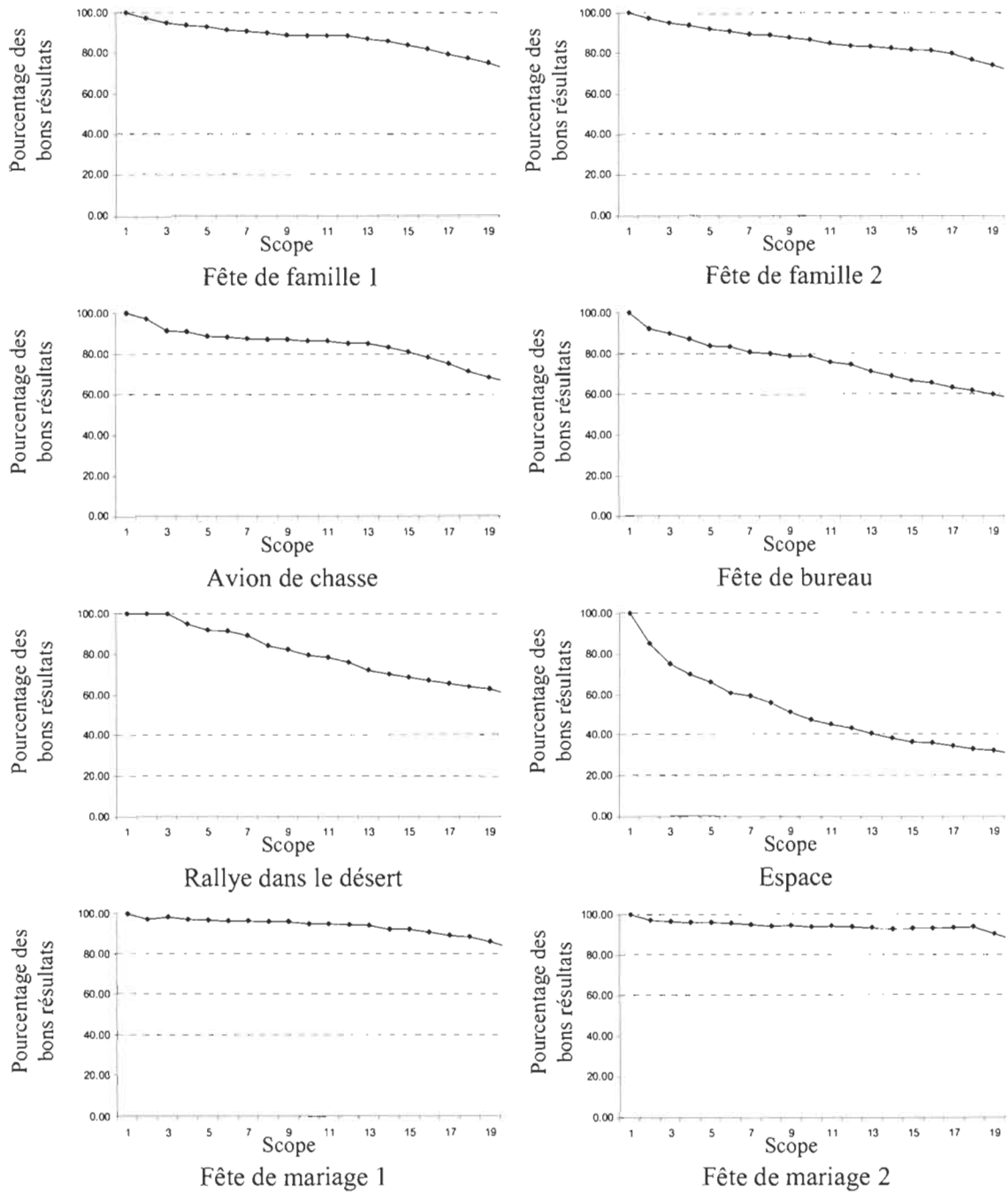


Figure 4.17 : Résultats de la recherche versus le scope des familles de vidéo (suite).

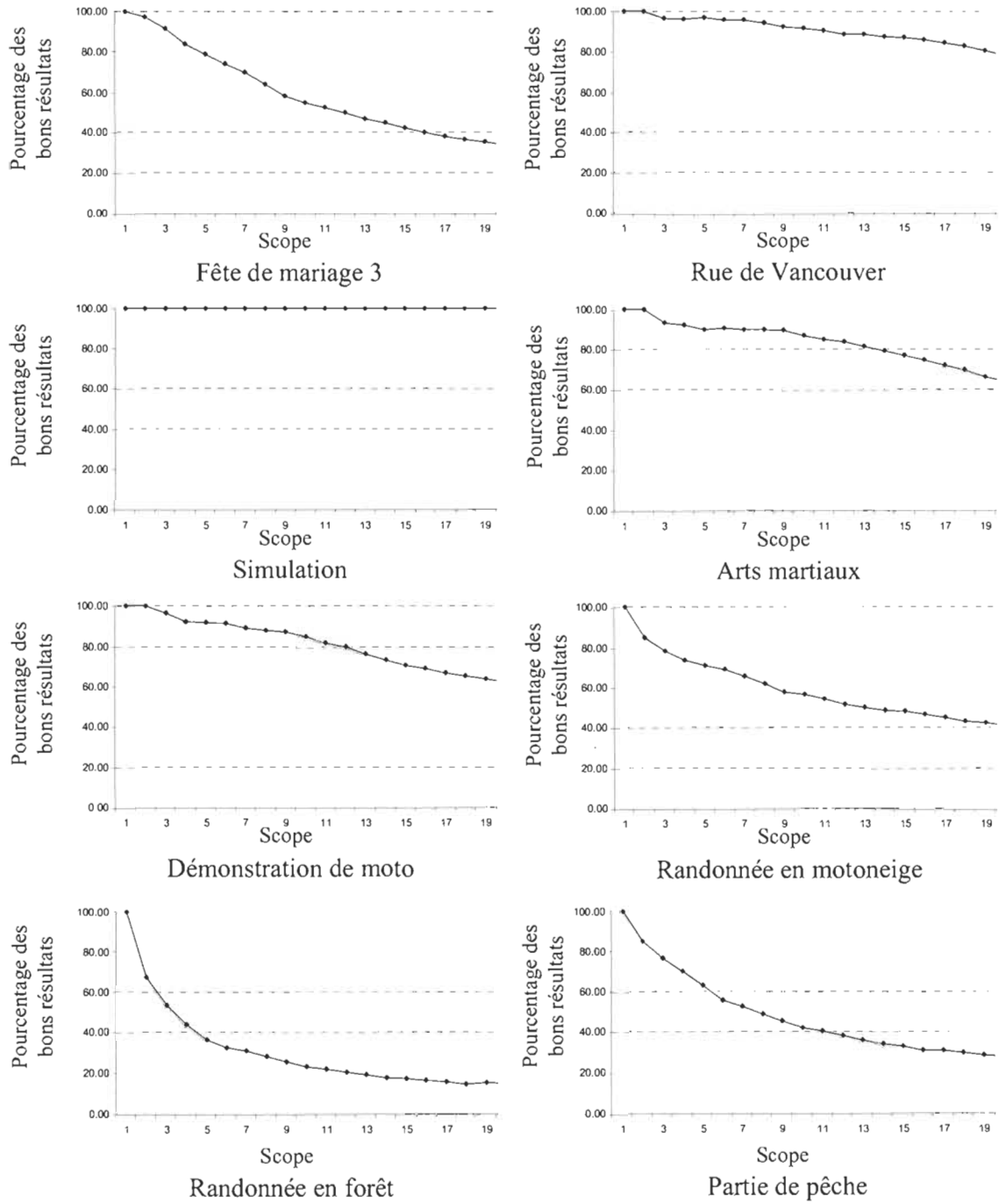


Figure 4.18 : Résultats de la recherche versus le scope des familles de vidéo (suite).

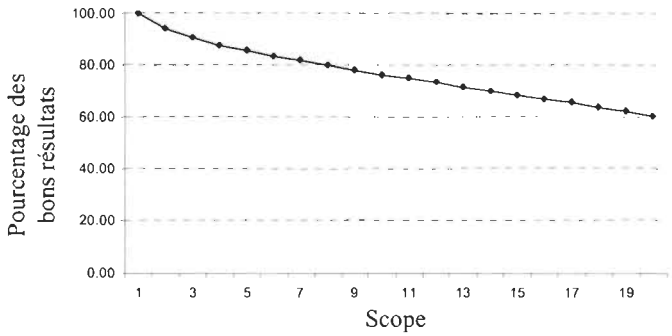


Figure 4.19 : La moyenne des résultats de la recherche versus le scope de toutes les familles de vidéo.

D'après la courbe de la moyenne des résultats de toutes les familles de vidéos, nous remarquons que la précision démarre haute et ne dégringole pas vite. Par exemple, les résultats sont encore au dessus de 60 %, même rendu à vingt (scope égal à 20). Nous jugeons que c'est un bon résultat sur une base de données de 600 vidéos.

Après l'analyse détaillée des résultats de chaque famille de vidéo, nous remarquons que les courbes des résultats des familles « Randonnée en forêt », « Randonnée en motoneige » et « Partie de pêche », descendent très vite. Le mouvement des objets et les mouvements de la caméra (le zoom) font que le contenu de ces vidéos n'est pas cohérent. Par conséquent, les couleurs, les textures et les formes peuvent varier énormément entre le début d'un plan et sa fin, ce qui fait que ces caractéristiques n'arrivent pas à représenter fidèlement le contenu du plan. Par contre, dans les vidéos où leur contenu est cohérent, comme dans la famille « Simulation », les résultats sont parfaits.

4.5.4 Quatrième expérience

L'objectif de cette expérience est de vérifier la pertinence des résultats de notre système d'indexation et de recherche de la vidéo, en demandant à des utilisateurs de l'évaluer. Nous avons demandé à six utilisateurs de tester notre système. À chaque fois,

l'utilisateur doit choisir une vidéo au hasard, puis faire une recherche en utilisant la combinaison des caractéristiques et les poids de pondération que nous avons sélectionnés. À la fin, il doit évaluer la pertinence des résultats de la recherche versus le scope. Pour chaque valeur de scope (de un jusqu'à vingt), l'utilisateur doit juger si la vidéo retournée est bonne ou mauvaise. Nous avons demandé à chaque utilisateur de faire 30 tests. Nous présentons ci-dessous la courbe précision-scope *Pr=f(Sc)* de la moyenne de l'évaluation des utilisateurs.

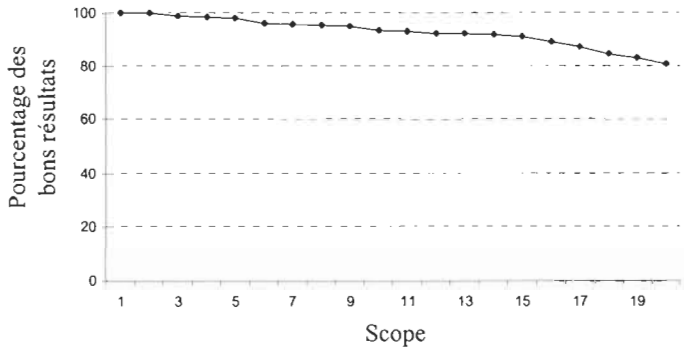


Figure 4.20 : La moyenne de l'évaluation des utilisateurs.

Nous remarquons que les résultats de l'évaluation des utilisateurs sont meilleurs que les résultats de la troisième expérience. La courbe des résultats commence haute et descend lentement. Même rendue à vingt résultats (scope égal à 20), la précision reste haute : 81 %. La cause de cette différence est le fait que les utilisateurs peuvent juger de la pertinence des vidéos même s'ils n'appartiennent pas à la même famille. Alors que dans notre comparaison automatique, nous avons pénalisé les vidéos qui n'appartiennent pas à la même famille. Ainsi, la précision réelle de ces expériences est encore plus élevée comme en attestent les utilisateurs auxquels nous avons fait appel lors de la quatrième expérience.

4.6 Conclusion

Notre système d'indexation et de recherche de la vidéo par le contenu a démontré sa fiabilité et sa précision après l'analyse des résultats des multitudes de tests que nous avons réalisés. Ces tests nous ont permis de sélectionner la bonne combinaison de caractéristiques et les bons poids de pondération, utilisés lors du calcul des distances entre la vidéo requête et les vidéos de la base de données. Même si dans certains cas une seule caractéristique est suffisante pour avoir de bons résultats de recherche, dans la plupart des cas, l'utilisation de plusieurs caractéristiques avec des poids de pondérations est plus adaptée à une base de données de vidéo volumineuse.

Conclusion

Dans ce mémoire, nous avons traité tous les aspects liés au développement d'un outil d'indexation et de recherche des vidéos personnelles par le contenu.

Notre choix a été motivé par la quantité phénoménale de vidéos personnelles disponibles aujourd'hui, qui ne cesse de croître, et le fait que les systèmes existants se sont surtout intéressés à d'autres types de vidéos et d'applications tels que le résumé automatique des bulletins d'information ou encore l'extraction des faits saillants des matchs de soccer, tennis, etc.

Dans le premier chapitre, nous avons décrit tous les aspects liés à l'indexation et la recherche de la vidéo par le contenu, tout en rapportant au fur et à mesure les travaux qui ont été réalisés dans le domaine. En plus, nous avons présenté les applications possibles d'un système de recherche de la vidéo par le contenu. Ensuite, nous avons présenté les systèmes ou les prototypes existants et ou en développement.

Ceci fait, nous avons abordé en détail tous les problèmes qu'il faut résoudre afin de pouvoir développer un système précis et performant, tels que le découpage de la vidéo, l'extraction de l'image clé, l'extraction des caractéristiques de bas niveau (la couleur, la texture, la forme et le mouvement) et des caractéristiques de haut niveau (l'annotation automatique et semi-automatique). Nous avons aussi exposé la problématique de la formulation de la requête dans les systèmes de recherche de la vidéo par le contenu.

D'après notre étude, les caractéristiques sont la base de tout système de recherche de la vidéo. En effet, le problème de recherche de la vidéo est presque toujours ramené à un problème de comparaison entre caractéristiques. Donc, si l'on veut développer un

système qui est précis et efficace, on doit passer par l'adoption et/ou le développement de bonnes caractéristiques.

Forts de cette constatation, nous avons conduit dans le deuxième chapitre une étude détaillée des caractéristiques visuelles des vidéos. Pour ce faire, nous avons présenté les différentes méthodes utilisées pour les extraire et les problèmes qu'il faut résoudre pour que les caractéristiques représentent bien le contenu de la vidéo.

Nous avons utilisé les informations acquises précédemment pour développer notre système de recherche de la vidéo. Dans le troisième chapitre, nous avons présenté l'architecture de notre système, les modules qui le constituent, les caractéristiques des vidéos que nous avons extraites et les méthodes de leur extraction, ainsi que la méthode de comparaison que nous avons utilisée lors de la recherche. À la fin, nous avons expliqué le fonctionnement de l'interface de notre système.

L'évaluation de notre système de recherche que nous avons effectuée dans le quatrième chapitre nous prouve que notre outil est performant, que les caractéristiques que nous avons extraites représentent bien le contenu de la vidéo, et que notre méthode d'extraction de l'image clé est efficace.

Pour conclure, nous pouvons dire que nous avons atteint notre objectif initial, soit le développement d'un outil pour l'indexation des vidéos personnelles par le contenu. Nous sommes partis de la constatation que le nombre de vidéos personnelles disponibles sur des supports de stockage personnels ou partagés sur Internet a explosé, et qu'il n'existe pas de systèmes qui permettent l'accès et la localisation de ce genre de vidéos. Ainsi, nous avons développé un outil qui exploite les caractéristiques de bas niveau (couleur, texture, mouvement) dans la vidéo pour rechercher une vidéo dans une grande collection.

Donc, sans être prétentieux, nous pouvons dire que nous avons apporté une innovation dans le domaine de la recherche de la vidéo par le développent de notre outil. Nous avons aussi développé une nouvelle méthode pour l'extraction de l'image clé. Le choix de cette image est crucial, puisque c'est elle qui remplace le plan lors de l'extraction des caractéristiques. Par conséquent, un bon choix de l'image clé contribue considérablement à l'amélioration des performances du système. L'autre innovation est le développement d'une nouvelle caractéristique qui représente bien le contenu de l'image. C'est une combinaison entre la caractéristique de l'histogramme de la couleur et la caractéristique du contour. Nous l'avons appelé l'histogramme de la couleur aux alentours des points de contour. Une autre originalité de notre approche est l'utilisation d'une méthode simple et efficace pour faire la comparaison entre les caractéristiques. C'est une somme pondérée des distances Euclidiennes entre les caractéristiques des vidéos.

Plusieurs applications possibles pourraient découler de notre approche. La première est son intégration dans un système destiné à faire l'annotation semi-automatique des vidéos. L'utilisateur du système annote manuellement quelques vidéos, puis utilise notre système de recherche de la vidéo pour faire de la propagation des mots-clés au reste des vidéos de la base de données. Cela a été implémenté avec succès dans l'outil développé par Sidibé [140]. D'autres applications possibles de notre approche sont la vidéo sur demande utilisée par les fournisseurs de services de télévision, l'accès aux vidéos sur Internet et leur organisation comme dans YouTube ou la production automatique des vidéos.

Des améliorations potentielles peuvent être faites à notre outil. En premier, il y a l'ajout de nouvelles caractéristiques de bas niveau qui caractérisent le mouvement, le son, le texte, la reconnaissance de la parole et la reconnaissance des visages. En deuxième lieu, il y a l'exploitation du côté sémantique de la recherche de la vidéo par l'intégration de l'annotation automatique. Cela devrait contribuer énormément à l'amélioration de la

précision des recherches. À propos du domaine de la recherche de la vidéo en général, nous voyons comme une piste prometteuse le développement d'ontologie pour créer un standard qui aide à la recherche par la sémantique.

Références

[1] Wactlar, H. D. (1999). New directions in video information extraction and summarization. *Proc. 10th DELOS Workshop,* Greece, pp. 1–10.

[2] Wactlar, H., Kanade T., Smith, M.A., & Stevens, S.M. (1996). Intelligent access to digital video: The informedia project, *IEEE Computer*, 29(5).

[3] Hauptmann, A. & Smith, M. (1995). Text, Speech, and Vision for Video Segmentation: The Informedia Project, *AAAI Fall 1995 Symposium on Computational Models for Integrating Language and Vision.*

[4] Carnegie Mellon University.*The Informedia Project Research Timeline.* Récupéré le 15 juin 2009 de http://www.informedia.cs.cmu.edu/timeline/index.html

[5] Hauptmann, A., Chen, M.-Y., & Christel, M. (2004). *Confounded expectations: Informedia at TRECVID 2004*, TREC Video Retrieval Evaluation Online Proceedings.

[6] Smeaton, A.F. (2002). The Físchlár Digital Library: Networked Access to a Video Archive of TV News. *TERENA Networking Conference 2002*, Limerick, Ireland, 3-6 June.

[7] Cooke, E., Ferguson, P., Gaughan, G., Gurrin, C., Jones, G., Borgue, H.L., Lee, H., Marlow, S., McDonald, K., McHugh, M., Murphy, N., O'Connor, N., O'Hare, N., Rothwell, S., Smeaton, A., & Wilkins, P. (2004). TRECVID 2004 experiments in Dublin city university, *TREC Video Retrieval Evaluation Online Proceedings*.

[8] Imperial College London. *Multimedia and Information Systems: Research.* Récupéré le 29 juin 2009 de http://mmis.doc.ic.ac.uk/research.html

[9] Heesch, D., Howarth, P., Magalhães, J., May, A., Pickering, M., Yavlinski, A., & S. Rüger. (2004). Video Retrieval using Search and Browsing. *TREC2004 – Text REtrieval Conference*, Gaithersburg, Maryland, 15-19 November.

[10] Jesus, R., Magalhães, J., Yavlinski, A., & Rüger, S. (2005). Imperial College at TRECVID, *TRECVid 2005 – Text REtrieval Conference*, TRECVID Workshop, Gaithersburg, Maryland, 14-15 November.

[11] Amir, A., Srinivasan, S., Ponceleon, D., & Petkovic, D.Cue. (1999). Video: Automated video/audio indexing and browsing. *Proceedings of the ACM International Conference on Research and Development in Information Retrieval*

(SIGIR'99) (Berkeley, CA, August 15 – 19, 1999), ACM Press, New York, NY, 326.

[12] Amir, A., Argillander, J., Campbell, M., Haubold, A., Iyengar, G., Ebadollahi, S., Kang, F., Naphade, M.R., Natsev, A.P., Smith, J.R., Tesic, J., & Volkmer, J. (2005). IBM Research TRECVID-2005 Video Retrieval System. *NIST TRECVID workshop* Gaithersburg, MD.

[13] The Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2). Récupéré le 10 août 2009 de http://www.im2.ch/

[14] Viper : Multimedia Information Retrieval and Management. Récupéré le 10 août 2009 de http://viper.unige.ch/doku.php

[15] Mediadico. Consulté le 15 août 2009 - http://www.mediadico.com/

[16] Boreczky, J., & Rowe, L. (1996). Comparison of video shot boundary detection techniques. *Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases*, volume 2670, pages 170–179.

[17] Lupatini, G., Saraceno, C., & Leonardi, R. (1998). Scene break detection: a comparison. *8th International Workshop on Research Issues in Data Engineering*, pages 34–41.

[18] Lienhart, R. (2001). Reliable transition detection in videos: A survey and a practitioner's guide. *International Journal of Image and Graphics*, 1(3):469–486.

[19] Yusoff, Y., Christmas, W., & Kittler, J. (1998). A study on automatic shot change detection. *Proceedings of the 3rd European Conference on Multimedia Applications, Services and Techniques (ECMAST)*, pages 177–189, May.

[20] Kikukawa, T., & Kawafuchi, S. (1992). Development of an automatic summary editing system for the audio-visual resources. *Transactions on Electronics and Information*, J75 A(2):204– 212.

[21] Zhang, H., Kankanhalli, A., & Smoliar, S. W. (1993). Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, June.

[22] Nagasaka, A., & Tanaka, Y. (1992). Automatic video indexing and full-search for video appearances. *Visual database Systems,* volume II, Amsterdam, pages 113–127.

[23] Press, W., Flannery, B., Teukolsky, S., & Vetterling, W. (1988-1992). *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press.

[24] Gargi, U., Kasturi, R., & Strayer, S. (2000). Performance characterization of video-shot change detection methods. *IEEE Transactions on Circuits and Systems for Video Technology,* 10(1):1–13, February.

[25] Lienhart, R. (1999). Comparison of automatic shot boundary detection algorithms. *SPIE Conf. on Storage and Retrieval for Image & Video Databases* VII, volume 3656, pages 290–301.

[26] Ueda, H., Miyatake, T., & Yoshizawa, S. (1991). Impact: An interactive natural-motion picture dedicated multimedia authoring system. *CHI '91*, pages 343–350.

[27] Otsuji, K., & Tonomura, Y. (1993). Projection detecting filter for video cut detection. *ACM Multimedia '93 Proceedings*, pages 251–257.

[28] Hanjalic, A. (2002). Shot-boundary detection: Unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology,* 12(2):90–105, February.

[29] Akutsu, A., Tonomura, Y., Hashimoto, H., & Ohba, Y. (1992). Video indexing using motion vectors. *SPIE Visual Communication and Image Processing*, volume 1818, pages 1522–1530.

[30] Shahraray, B. (1995). Scene change detection and content-based sampling of video sequences. *SPIE Conference on Digital Video Compression: Algorithms and Technologies,* volume 2419, pages 2–13, February.

[31] Vlachos, T. (2000). Cut detection in video sequences using phase correlation. *IEEE Signal Processing Letters*, 7(7):173–175, July.

[32] Fernando, W.A.C., Canagarajah, C.N., & Bull, D.R. (1999). Video segmentation and classification for content based storage and retrieval using motion vectors. *Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 687–698.

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

[34] Dailianas, A., Allen, R.B., & England, P. (1995). Comparison of automatic video segmentation algorithms. *Proceedings of the SPIE Conference on Integration Issues in Large Commercial Media Delivery Systems*, volume 2615, pages 2–16.

[35] Lienhart, R. (2001). Reliable dissolve detection. *Proceedings of the SPIE Conference on Storage and Retrieval for Media Databases,* volume 4315, pages 219–230, January.

[36] Lu, H.B., Zhang, Y.J., & Yao, Y.R. (1999). Robust gradual scene change detection. *IEEE International Conference on Image Processing,* volume 3, pages 304–308.

[37] Ngo, C.W., Pong, T.C., & Chin, R.T. (1999). Detection of gradual transitions through temporal slice analysis. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* volume 1, pages 36–41.

[38] Ngo, C.W., Pong, T.C., & Chin, T.R. (2001). Video partitioning by temporal slice coherency. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(8):941–953, August.

[39] Alattar, A.M. (1993). Detecting and compressing dissolve regions in video sequences with a dvi multimedia image compression algorithm. *IEEE International Symposium on Circuits and Systems,* volume 1, pages 13–16, May.

[40] Fernando, W.A.C., Canagarajah, C.N., and Bull, D.R. (2000). A unified approach to scene change detection in uncompressed and compressed video. *IEEE Transactions on Consumer Electronics*, 46(3):769–779, August.

[41] Truong, B.T., Dorai, C., & Venkatesh, S. (2000). New enhancements to cut, fade, and dissolve detection processes in video segmentation. *Proceedings of the 8th ACM International Conference on Multimedia,* pages 219–227.

[42] Nam, J., & Tewfik, A.H. (2000). Dissolve transition detection using b-splines interpolation. *IEEE International Conference on Multimedia and Expo*, volume 3, pages 1349–1352.

[43] Patel, N.V., & Sethi, I.K. (1996). Compressed Video Processing for Cut Detection. *IEEE Proc. Visual Image Signal Process,* vol. 143, no. 5, pp. 315-23, Oct.

[44] Deardor, E., Little, T.D.C., Marshall, J.D., Venkatesh, D., & Walzer, R. (1994). Video Scene Decomposition with the Motion Picture Parser. *SPIE Conf. Digital Video Compression on Personal Computers Algorithms and Technologies,* Vol.2187, pp.44-55.

[45] Deng, Y., & Manjunath, B.S. (1997). Content-based Search of Video using Color, Texture, and Motion. *Proceedings of IEEE Intl. Conf. on Image Processing*, vol.2, pp 534-537.

[46] Shen, B., Li, D., Sethi, I.K. (1997). HDH Based Compressed Video Cut Detection. *Second Intl. Conf. on Visual Information Systems*, pp. 149-156, Dec.

[47] Chang, S.H., Sull, S., & Lee, S.U. (1999). Efficient video indexing scheme for content based retrieval. *IEEE Transactions on Circuits and Systems for Video Technology,* 9(8):1269–1279, December.

[48] Dufaux, F. (2000). Key frame selection to represent a video. *International Conference on Image Processing,* pages 275–278.

[49] Brunelli, R., Mich, O., & Modena, C.N. (1999). A survey on the automatic indexing of video data. *Journal of Visual Communication and Image Representation,* 10(2):78–112, June.

[50] Idris, F., & Panchanathan, S. (1997). Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation*, 8(2):146–166.

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

[52] Gunsel, B., & Tekalp, A.M. (1998). Content-based video abstraction. *IEEE International Conference on Image Processing,* volume 3, pages 128–132.

[53] Chen, H.-Y., & Wu, J.-L. (1995). A multi-layer video browsing system. *IEEE Transactions on Consumer Electronics,* 41(3):842–850, August.

[54] Ardizzone, E., & Cascia, M.L. (1996). Video indexing using optical flow field. *IEEE International Conference on Image Processing,* volume 3, pages 831–834.

[55] Zhang, H.J., Wu, J., Zhong, D., & Smoliar, S.W. (1997). An integrated system for content based video retrieval and browsing. *Pattern Recognition,* 30(4):643–658.

[56] Kim, S.H., & Park, R.H. (2002). A novel approach to video indexing using luminance projection. *Proceedings of the IASTED International Conference on Signal and Image Processing,* pages 359–362.

[57] Xiong, W., Lee, C.M., & Ma, R.H. (1997). Automatic video data structuring through shot partitioning and key-frame computing, *Machine Vision and Applications,* vol.10, no.2, pp. 55-65.

[58] Gresle, P.O., & Huang, T.S. (1997). Gisting of Video Documents: A Key Frames Selection Algorithm Using Relative Activity Measure, *the 2nd Int. Conf. on Visual Information System*, pp. 297-86, 1997.

[59] Vermaak, J., Peraz, P., Gangnet, M., & Blake, A. (2002). Rapid summarisation and browsing of video sequences. *British Machine Vision Conference*, volume 1, pages 424–433.

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

[61] Djeraba, C. (2002). Content-based multimedia indexing and retrieval, *IEEE Multimedia,* Volume 9, page 18 – 22, April-June.

[62] Stricker, M.A., & Dimai, A. (1996). Color indexing with weak spatial constraints. *Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases*, volume 2670, pages 29–40.

[63] Huang, J., Kumar, S.R., Mitra, M., Zhu, W.-J., & Zabih, R. (1997). Image indexing using color correlograms. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 762–768.

[64] Tuceryan, M., & Jain, A. K. (1998). *The Handbook of Pattern Recognition and Computer Vision.* 2nd Edition. World Scientific Publishing Co.

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

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

[67] Manjunath, B.S., & Ma, W.Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Analysis and. Machine Intelligence*, 18(8) :837–842.

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

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

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

[71] Hu, M.-K. (1962). Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 8(2):179,187.

[72] Reiss, T. (1991). The Revised Fundamental Theorem of Moment Invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 13(8):830,834.

[73] Gunsel, B., & Tekalp, M. (1998). Shape similarity matching for query-by-example.

Pattern Recognition, 31(7):931-944.

[74] Loncaric, S. (1998). A Survey of Shape Analysis Techniques. *Pattern Recognition*,

31(8):983-1001.

[75] Shah, J. (2005). Gray skeletons and segmentation of shapes. *Computer Vision and Image Understanding,* 99(1):96-109.

[76] Veltkamp, R. (2001). *Shape Matching: Similarity Measures and Algorithms*, Rapport Technique UU-CS-2001-03, Université de Utrecht, Pays-Bas.

[77] Barron, J.L., Fleet, D.J., & Beauchemin, S.S. (1994). Performance of optical flow techniques. Int. J. Comput. Vision, 12(1) :43–77.

[78] Quénot, G. (1996). Computation of optical flow using dynamic programming. *Proceedings of IAPR Workshop on Machine Vision Applications*.

[79] Paulin, D., Kumar, D., Bhaskar, R., & Quénot, G. (2001). Recovering camera motion and mobile objects in video documents. *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, pages 39–46.

[80] Durik, M., & Benois-Pineau, J. (2001). Robust motion characterisation for video indexing based on MPEG2 optical flow. *Proceedings of the International Workshop on Content-Based Multimedia Indexing,* pages 57–64.

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

[82] Horn, B., & Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, 17 :185–203.

[83] Lucas, B.D., & Kanade, T. (1981) An iterative image registration technique with an application to stereo vision. *Proceedings of International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

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

[85] Boreczhy, J. S., & Wilcox, L. D. (1998). A Hidden Markov Model Framework for Video Segmentation Using Audio and Image Features. *Proceedings of IEEE International Conference*, Seattle, WA, USA ,vol. 6, pp. 3741-3744, Mai.

[86] Kijak, E., Gravier, G., Oisel, L., & Gros, P. (2003). Structuration Multimodale d'une Vidéo de Tennis par Modèles de Markov Cachés. *Colloque sur le Traitement du Signal et des Images,* France, 2003.

[87] Kittler, J., Messer, K., Christmas, W. J., Levenaise-Obadia, B., & Koubaroulis, D. (2001). Generation of Semantic Cues for Sports Video Annotation. *International Conference on Image Processing, Centre for Vision, Speech and Signal Processing*, School of Electronics, Computing and Mathematics, University of Surrey, Guildford, UK, vol.3, pp. 26-29.

[88] Song, Y., Hua, X.-S., Dai, L., & Wang, M. (2005). Semi-automatic video annotation based on active learning with multiple complementary predictors. *ACM International Workshop on Multimedia Information Retrieval.*

[89] Fan, J., Luo, H., & Elmagarmid, A.K. (2004). Concept-Oriented Indexing of Video Databases: Toward Semantic Sensitive Retrieval and Browsing. *IEEE Transactions of Image Processing,* Vol. 13, No. 7, July.

[90] Smith, J. R., & Chang, S.-F. (1995). *Automated image retrieval using color and texture.* Technical Report CU/CTR 408-95-14, Columbia University, July.

[91] Gonzalez, R.C., & Woods, R. E. (2002). *Digital Image Processing*. Prentice Hall.

[92] Stricker, M., & Orengo, M. (1995). Similarity of color images. *Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Databases III*, volume 2420, pages 381-392, February.

[93] Swain, M., & Ballard, D. (1991). Color indexing. *International Journal of Computer Vision,* Vol.7, No. 1, pp. 11-32.

[94] Hadjidemetriou, E., Grossberg, M. D., & and Nayar, S. K. (2004). Multiresolution Histograms and Their Use for Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 26(7):831–847.

[95] Nayar, S., & Bolle, R. (1996). Reflectance based object recognition. *Int. J. Comput. Vision,* 17(3):219–240.

[96] Finlayson, G.D., Chatterjee, S. S., & Funt, B. V. (1996). Color angular indexing. *Proceedings of the Second European Conference on Computer Vision*, pages 16–27.

[97] Slater, D., & Healey, G. (1996). The illumination-invariant recognition of 3d objects using local color invariants. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(2):206–210.

[98] Wong, K.-M., Chey, C.-H., Liu, T.-S., & Po, L.M. (2003). Dominant color image retrieval using merged histogram. *Proceedings of the International Symposium on Circuits and Systems,* Volume 2, pp: II-908 - II-911 vol.2, May.

[99] Deng, Y., Manjunath, B.S., Kenney, C., Moore, M.S., & Shin, H. (2001). An Efficient Color Representation for Image Retrieval. *IEEE Trans. Image Processing,* 10(1):140–147.

[100] Kherfi, M.L., Ziou, D., & Bernardi, A. (2003). Combining positive and negative examples in relevance feedback for content-based image retrieval. *Journal of Visual Communication and Image Representation,* Vol. 14, No. 4. pp. 428-457.

[101] El-Feghi, I., Aboasha, H., Sid-Ahmed, M.A., & Ahmadi, M. (2007). Content-Based Image Retrieval based on efficient fuzzy color signature. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp: 1118 – 1124, October.

[102] Haralick, R.M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE,* Volume: 67, Issue: 5, page(s): 786- 804.

[103] Weszka, J.S., Dyer, C.R., & Rosenfeld, A. (1976). A comparative study of texture measures for terrain classification, *IEEE Trans. Systems, Man Cybernet.* 6 (1976) 269–285.

[104] Levine, M. (1985). *Vision in Man and Machine,* McGraw-Hill.

[105] Cross, G.R., & Jain, A.K. (1983). Markov random field texture models. *IEEE Trans. Pattern Analysis and Machine Intelligence,* PAMI-5(1): 25–39.

[106] Pentland, A. (1984). Fractal-Based Description of Natural Scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 6, 6, 661-674.

[107] Chellappa, R. (1985). Two-dimensional discrete Gaussian Markov random field models for image processing. *Machine Intelligence and Pattern Recognition: Progress in Pattern Recognition* 2, G.T. Toussaint (Ed.), Elsevier Science Publishers, B.V. (North-Holland). pp. 79–112.

[108] Derin, H., & Elliot, H. (1987). Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9, 1, 39-55.

[109] Manjunath, B., & Chellappa, R. (1991). Unsupervised Texture Segmentation Using Markov Random Fields. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13, 5, 478-482, 1991.

[110] Strzelecki, M., & Materka, A. (1997). Markov Random Fields as Models of Textured Biomedical Images. *Proceedings of 20th National Conf. Circuit Theory and Electronic Networks KTOiUE '97*, Kołobrzeg, Poland, 493-498.

[111] Daugman, J. (1985). Uncertainty Relation for Resolution in Space, Spatial Frequency and Orientation Optimised by Two-Dimensional Visual Cortical Filters. *Journal of the Optical Society of America*, 2, 1160-1169.

[112] Bovik, A., Clark, M., & Giesler, W. (1990). Multichannel Texture Analysis Using Localised Spatial Filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12, 55-73.

[113] Rosenfeld, A., & Weszka, J. (1980). Picture Recognition. In *Digital Pattern Recognition*, K. Fu (Ed.), Springer-Verlag, 135-166, 1980.

[114] Mallat, S. (1989). Multifrequency Channel Decomposition of Images and Wavelet Models. *IEEE Trans. Acoustic, Speech and Signal Processing*, 37, 12, p. 2091-2110.

[115] Laine, A., & Fan, J. (1993). Texture Classification by Wavelet Packet Signatures. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15, 11, p. 1186-1191.

[116] Lu, C., Chung, P., & Chen, C. (1997). Unsupervised Texture Segmentation via Wavelet Transform. *Pattern Recognition*, 30, 5, p. 729-742.

[117] Bharati, M. H., Jay Liu, J., & MacGregor, J.F. (2004). Image texture analysis: methods and comparisons. *Chemometrics and intelligent laboratory systems* , vol. 72, n°1, pp. 57-71.

[118] Sharma, M., Singh, S. (2001). Evaluation of texture methods for image analysis. *Proceedings of the 7th Australian and New Zealand Intelligent Information Systems Conference.*

[119] Zhang, J., & Tan, T. (2001). Brief review of invariant texture analysis methods. *Pattern Recognition.* Vol. 35, iss. 3, March, pp. 735-747.

[120] Haralick, R.M., Shanmugan, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Trans. Syst. Man. Cybern.* SMC-3(6): 610–621.

[121] Unser, M. (1986). Sum and difference histograms for texture classification . *IEEE transactions on pattern analysis and machine intelligence,* vol. PAMI-8, No. 1 Janv.

[122] Torre V., & Poggio T. (1986). On edge detection. *IEEE Pattern Analysis and Machine Intelligence,* vol. 8, p. 147-153.

[123] Marr D., & Hildreth E. (1980). Theory of edge detection. *Proc. Royal Soc. London*, vol. 207 de B, p. 187-217.

[124] Jeannin, S., & Divakaran, A. (2001). MPEG-7 Visual Motion Descriptors. *IEEE Transactions on circuits and systems for video technology,* Vol 11, No. 6, June.

[125] Divakaran, A., Ito, H., Sun, H., & Poon, T. (1998). Scene change detection and feature extraction for indexing MPEG-2 and MPEG-4 sequences. *IEEE Trans. Circuits Systems Video Technology* , Oct.

[126] Fablet, R., Bouthemy, P., & Pérez, P. (2000). Statistical motion-based video indexing and retrieval. *Proceedings of 6th Int. Conference on Content-Based Multimedia Inf. Access*, RIAO'2000, , Paris, April, pp. 602-619.

[127] Jain, A.K., Vailaya, A, & Xiong, W. (1999). Query By Video Clip. *Multimedia Systems: Special Issue on Video Libraries,* vol. 7, no. 5, Mai, pp. 369-384.

[1287] Smolic, A, Sikora, T., & Ohm, J.-R. (1999). Long-term global motion estimation and its application for sprite coding, content description, and segmentation. *IEEE Trans. On Circuits and Systems for Video Technology,* Vol. 9, No. 8, December, pp. 1227-12242.

[129] Gelgon, M., & Bouthemy, P. (1998). Determining a structured spatio-temporal representation of video content for efficient visualization and indexing. *Proceedings of the 5th European Conference on Computer Vision,* ECCV'98, Springer, Freibourg, Juin, Vol 1406, pp. 595-609.

[130] Jeannin, S., Mory, B. (2000). Video Motion Representation for Improved Content Access. *IEEE Trans. on Consumer Electronics,* Vol. 46, No. 3, August, pp. 645-655.

[131] Deng, Y., & Manjunath, B.S. (1998). NeTra-V: Toward an Object-based Video Representation. *IEEE Transaction on Circuits and Systems for Video Technology,* Vol. 8, No.5, Septembre, pp. 616-627.

[132] Zaharia, T., Prêteux, F. (1999). *Motion descriptor: perspective transformation parameters and object trajectory.* Proposal P351, MPEG-7 Proposal Evaluation Meeting, Lancaster, UK, Feb.

[133] Prêteux, F., Zaharia, T., & Preda, M. (1999). Parametric Object Motion Descriptor. *ISO/IEC JTC1/SC29/WG11, MPEG99/M4870,* Vancouver, BC, Canada, July.

[134] Zaharia, T., & Prêteux, F. (2001). Parametric motion models for video content description within the MPEG-7 framework. *Proceedings SPIE Conf. on Nonlinear Image Proc. and Pattern Analysis,* San Jose, USA, 22-23 January.

[135] Chang, S.-F., Chen, W., Meng, H.J., Sundaram, H., & Zhong, D. (1998). A Fully Automated Content-Based Video Search Engine Supporting Spatiotemporal Queries. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 5, September, pp.602-615.

[136] The Open Video Project. Consulté le 15 août 2009- www.open-video.org

[137] TREC Video Retrieval Evaluation. Consulté le 15 août 2009- http://www-nlpir.nist.gov/projects/trecvid/

[138] Ren, W., Weal, P., Singh, M., & Singh, S. (2006). Visual Information Retrieval : MINERVA Video Benchmark. *Proceeding of the 24th IASTED International Multi-Conference Signal Processing, Pattern Recognition, and Applications*, Feb 15-17, Innsbrusk, Austria.

[139] Smith, J.R., & Chang, S.F. (1996). Tools and techniques for color image retrieval. In: *Storage & Retrieval for Image and Video Databases* IV, vol. 2670 of IS&T/SPIE Proceedings, San Jose, USA, pp. 426–437.

[140] Sidibé, Y. (2009). *Un système pour l'annotation semi-automatique des vidéos et applications à l'indexation.* Mémoire de maîtrise, Université du Québec à Trois-Rivières.