



HAL
open science

Améliorer la recherche par similarité dans une grande base d'images fixes par des techniques de fouilles de données

Anicet Kouomou-Choupo

► **To cite this version:**

Anicet Kouomou-Choupo. Améliorer la recherche par similarité dans une grande base d'images fixes par des techniques de fouilles de données. Interface homme-machine [cs.HC]. Université Rennes 1, 2006. Français. NNT: . tel-00524418

HAL Id: tel-00524418

<https://theses.hal.science/tel-00524418>

Submitted on 7 Oct 2010

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

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

N° d'ordre: 3288

THÈSE

Présentée

devant l'Université de Rennes 1

pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention INFORMATIQUE

par

Anicet KOUOMOU CHOUPO

Équipe d'accueil : Texmex - IRISA

École Doctorale : Matisse

Composante universitaire : IFSIC

Titre de la thèse :

*Améliorer la recherche par similarité dans une grande base
d'images fixes par des techniques de fouille de données*

soutenue le 23 février 2006 devant la commission d'examen

M.	José	MARTINEZ	Président du jury
MM.	Chabane	DJERABA	Rapporteurs
	Mohand-Saïd	HACID	
M.	Vincent	ORIA	Examineur
Mmes.	Annie	MORIN	Directrices
	Laure	BERTI-ÉQUILLE	

Remerciements

Ce travail a été réalisé à l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires), au sein de l'équipe TEXMEX. Je tiens à exprimer toute ma gratitude à la Région de Bretagne et à l'Égide qui ont financé ce travail. Grand merci à Véronique Verdon et à Marie-Claude Sance-Plouchart pour votre accueil toujours plein de chaleur. Avec vous, je me suis senti chez moi à l'IRISA.

Je remercie José Martinez, professeur à l'École polytechnique de l'Université de Nantes, pour avoir présidé mon jury de thèse. J'adresse également mes sincères remerciements à Mohand-Saïd Hacid, professeur à l'Université Claude Bernard Lyon 1, ainsi qu'à Chabane Djeraba, professeur à l'Université des Sciences et Technologies de Lille, pour leur évaluation minutieuse de ce travail en qualité de rapporteurs. Je remercie aussi Vincent Oria, maître de conférences à l'Institut des Technologies du New Jersey aux USA pour avoir accepté d'examiner ce travail de thèse.

Je tiens à exprimer toute ma gratitude à l'équipe TEXMEX dirigée par Patrick Gros et qui m'a permis d'accomplir mon travail de thèse dans de bonnes conditions. Je remercie très vivement ma directrice de thèse Annie Morin. Tu as toujours su être présente quand il le fallait.

Je voudrais exprimer ma profonde reconnaissance envers ma co-directrice Laure Berti-Équille pour toute l'attention et le soutien qu'elle m'a portés pendant ces années de thèse et pour sa très grande disponibilité durant toute cette période.

Je remercie mes enseignants camerounais de l'université qui m'ont initié à cette aventure intellectuelle. J'adresse mes vifs remerciements à Maurice Tchuenté, professeur au département d'informatique de l'Université de Yaoundé 1, pour ses encouragements et son soutien constants.

Je ne saurais oublier ma famille qui m'a régulièrement soutenu durant les moments importants de la thèse. Je pense en particulier à Hélène Wokam (Tata Mammie), à Lynda, ainsi qu'à M. et à Mme. Waku Kouomou pour leur aide précieuse en fin de thèse. J'ai compris qu'il faut toujours compter sur quelqu'un d'autre pour parvenir à ses fins.

Ces années de thèse ont été faciles à vivre grâce aux amis qui ont régulièrement été présents. Un grand merci à Madeleine (pour les bonnes ambiances), à Edwige (la reine des bons plans), à Bernard (le saint plaisantin), à Thierry (le très gentil), à Liliane, à Judith, à Martine et à Rodrigue pour leurs précieux coups de main dans la préparation du pot de thèse. Je suis tout particulièrement reconnaissant envers Francine Ngani avec qui j'ai eu la chance de connaître toutes les joies d'une vraie amitié. Merci à tous ceux que je n'ai pas cités, ils ne sont pas oubliés.

Je remercie mes collègues de bureau Pierre-Hugues et Nicolas, ainsi que toutes les personnes que j'ai côtoyées durant ces trois années et qui m'ont permis de passer des moments agréables.

À mon feu père sans qui rien de ce qui a été fait n'aurait pu l'être

À ma mère

À mes sœurs et à mon frère

Table des matières

Table des matières	1
Introduction	5
1 Optimisation des performances de la recherche par le contenu dans une base d'images fixes	13
1.1 Introduction	13
1.2 Évaluation des performances des systèmes de recherche d'images	15
1.3 Similarité visuelle	16
1.3.1 Le modèle métrique	17
1.3.2 Recherche par similarité	18
1.4 Index multidimensionnels	19
1.4.1 Partitionnement de données	19
1.4.2 Partitionnement de l'espace	20
1.4.3 Malédiction de la dimension	21
1.5 Amélioration du temps de recherche	21
1.5.1 Les techniques de filtrage	22
1.5.2 Recherche approximative de plus proches voisins	23
1.6 Amélioration de la qualité des résultats de la recherche	25
1.6.1 Description d'images fixes	26
1.6.2 Recherche sur la couleur	28
1.6.3 Recherche sur la texture	31
1.6.4 Recherche sur la forme	33
1.6.5 Combinaison des types de descripteurs globaux	35
1.7 Synthèse	35
2 Fouille de données pour l'organisation des images fixes	37
2.1 Introduction	37
2.2 Tâches de la fouille de données	38
2.2.1 La classification supervisée	38
2.2.2 La classification non supervisée (<i>clustering</i>)	39
2.2.3 La découverte des règles d'association	41
2.2.4 La découverte de séquences fréquentes	42
2.3 Regroupement des descripteurs visuels	43

2.3.1	Méthodes de partitionnement	43
2.3.2	Arbres hiérarchiques	45
2.4	Règles d'association et indexation d'images	46
2.4.1	Calcul des <i>clusters</i> de descripteurs	48
2.4.1.1	Méthode <i>k-means</i>	48
2.4.1.2	Le <i>M-Tree</i>	49
2.4.2	Découverte des règles à partir des <i>clusters</i> de descripteurs	50
2.4.3	Processus d'organisation des images	51
2.5	Synthèse	51
3	Stratégies de recherche par le contenu basées sur la découverte de règles d'association, l'ordonnancement et le filtrage des <i>clusters</i> de descripteurs d'images	55
3.1	Introduction	55
3.2	Chaîne de traitement d'une requête par l'exemple	56
3.3	Définitions de base	57
3.4	Principe de sélection des règles d'association	59
3.5	Recherche progressive par le contenu	61
3.5.1	Stratégies d'exécution d'une requête progressive par l'exemple	63
3.5.2	Éligibilité des <i>clusters</i> de descripteurs	64
3.5.3	Utilisation des règles d'association dans un contexte progressif	65
3.5.4	Règles de filtrage	66
3.6	Fusion des résultats intermédiaires	67
3.7	Synthèse	68
4	Évaluations expérimentales	71
4.1	Introduction	71
4.2	Environnement expérimental	72
4.3	Description des données	72
4.4	Intérêt des règles d'association	73
4.4.1	Génération des règles d'association : étude des paramètres	73
4.4.2	Confirmation des règles par l'analyse des correspondances multiples	76
4.4.3	Utilisation effective des règles pour la recherche par le contenu	78
4.5	Performances de la recherche progressive	81
4.5.1	Critères de sélection des <i>clusters</i>	81
4.5.2	Gestion des résultats intermédiaires	82
4.5.3	Évaluation des règles de filtrage selon deux méthodes de construction des <i>clusters</i>	85
4.5.4	Passage à l'échelle de la méthode de recherche	90
4.6	Intérêt de la combinaison des descripteurs globaux	92
4.7	Synthèse	94

A Descripteurs <i>MPEG-7</i>	99
A.1 Contexte de standardisation	99
A.2 Description du contenu	100
A.3 Quelques descripteurs <i>MPEG-7</i> d'images fixes	101
A.3.1 Les descripteurs de couleur	101
A.3.1.1 Le descripteur <i>Color Layout</i>	102
A.3.1.2 Le descripteur <i>Scalable Color</i>	103
A.3.2 Les descripteurs de texture	104
A.3.2.1 Le descripteur <i>Homogeneous Texture</i>	104
A.3.2.2 Le descripteur <i>Edge Histogram</i>	105
A.3.3 Les descripteurs de forme	106
Bibliographie	118
Liste des notations	119
Liste des définitions	121
Liste des figures	123
Liste des tableaux	125
Références de l'auteur	127

Introduction

La recherche d'images par le contenu est très sollicitée dans les domaines tels que la synthèse d'images, l'imagerie par satellite, la météorologie, la télévision, l'imagerie médicale, *etc.* Ainsi, parmi les volumes considérables d'informations aujourd'hui disponibles, de grandes quantités d'images sont générées et manipulées. La complexité de leur structure requiert de nouvelles techniques de représentation et d'extraction à partir d'une très grande base de données. La recherche traditionnelle à partir d'un texte ou par mots-clés reste limitée par le faible pouvoir expressif des mots et par le caractère subjectif des annotations. Elle est surtout contraignante pour des collections de taille importante d'images si le texte ou les mots-clés sont générés manuellement. Ce type de recherche ne peut donc pas être utilisée dans toutes les applications. La description textuelle d'une image n'étant pas suffisamment riche pour détecter le contenu visuel de l'image, des algorithmes spécialisés en traitement d'images sont alors appliqués sur le contenu de l'image considérée comme un ensemble de pixels pour en extraire des valeurs numériques. Celles-ci sont ensuite utilisées afin de mesurer la similarité entre images pour répondre aux requêtes des utilisateurs.

La recherche d'images par le contenu consiste à développer des outils permettant de sélectionner les images les plus pertinentes par leur contenu selon différents types de descripteurs (par exemple la couleur), comme c'est le cas pour les systèmes de recherche d'informations. Ces outils doivent en outre s'occuper de la mise en place des stratégies d'organisation destinées à faciliter l'exploitation du contenu au cours d'une recherche. La description automatique des images par le contenu peut être abordée de plusieurs manières en fonction, d'une part, de ce que l'on souhaite décrire dans l'image et, d'autre part, suivant l'utilisation qu'on en fait. Les descripteurs utilisés pour détecter un visage ne sont par exemple pas les mêmes que ceux utilisés pour décrire une image de la mer. Les types de descripteurs sont alors nombreux et un des challenges de la recherche d'images est de pouvoir définir ou choisir le bon type de descripteur pour l'utilisateur non spécialiste. Orienter ainsi l'utilisateur dans le processus de recherche nécessite un minimum de connaissances sur les images traitées. La connaissance des images peut être donnée par un spécialiste qui contrôle le processus de génération de ces images dans un domaine spécifique comme l'analyse et le traitement des images médicales. Dans des domaines plus vagues tels que la télévision où il n'y a pratiquement pas d'expert, l'option de l'analyse automatique des images peut être envisagée à travers les techniques de fouille de données.

Fouille de données et multimédia

La fouille de données (*data mining*) se définit comme un procédé d'analyse par des moyens automatiques ou semi-automatiques de grandes quantités de données dans le but d'en découvrir des connaissances non triviales [Fayyad *et al.*, 1996b]. La quantité de données joue un rôle capital en fouille de données et influence la qualité de la connaissance ainsi extraite. Le procédé d'analyse pour la découverte de connaissances s'inscrit dans un cadre plus général d'un processus itératif d'extraction de connaissances en quatre étapes suivantes (figure 1).

1. **La sélection et l'intégration de bases de données.** Cette phase constitue l'une des étapes préliminaires du processus d'extraction de connaissances à partir de bases de données (ECBD). L'analyste définit ses objectifs, puis identifie les sources de données dont il aura besoin. Les problèmes de conversion de format, de nettoyage (élimination des bruits, inconsistances, *etc.*) et de compression de données doivent également être résolus ici. S'agissant des images, les techniques de description doivent être identifiées. Elles devront gérer les multiples formats de stockage/compression des images tels que JPEG, GIF, PNG, *etc.* ;
2. **Le prétraitement de données.** Les données recueillies ne sont pas toujours directement exploitables par les techniques de fouille de données. Le but du prétraitement de données est de les transformer dans un format manipulable par les algorithmes d'extraction de connaissance. En ce qui concerne les images, celles-ci doivent être décrites par des techniques appropriées et indexées dans des structures adaptées pour optimiser les accès par les algorithmes de fouille ;
3. **La fouille proprement dite.** Elle permet d'extraire de l'information des données. Cette étape est souvent difficile à mettre en œuvre, car coûteuse et les résultats doivent être correctement interprétés et visualisés de façon synthétique. En fouille d'images, les algorithmes tiennent explicitement compte de la complexité de description du contenu et des difficultés liées au passage à l'échelle dans le traitement d'un très grand nombre d'images (plusieurs milliers d'images par exemple). Pour l'aide à la décision, une grande majorité de résultats s'obtient par requête, par analyse multidimensionnelle ou grâce aux outils de visualisation ;
4. **La présentation et la validation des résultats.** L'analyste doit être informé de toutes les sorties issues des manipulations faites sur les données. C'est le but de la présentation. L'utilisateur validera ensuite les résultats de l'analyse.

Les techniques de la fouille de données sont utilisées en imagerie médicale pour l'aide au diagnostic. Il peut s'agir par exemple d'élaborer un modèle de données qui étend le dossier médical d'un patient aux images qu'il peut contenir (radiographie, électrocardiogramme, mammographie, *etc.*), puis d'analyser l'évolution des caractéristiques des images dans le temps pour prédire d'éventuelles anomalies. L'analyse implique cependant que le modèle tienne compte de la sémantique des images qui traduisent des pathologies. Pour ce faire, le modèle exploite des descriptions intermédiaires plus ou moins directement liées au contenu. L'annotation des images par des mots-clés est une stratégie indirectement dépendante du contenu. Elle exige une connaissance et parfois la

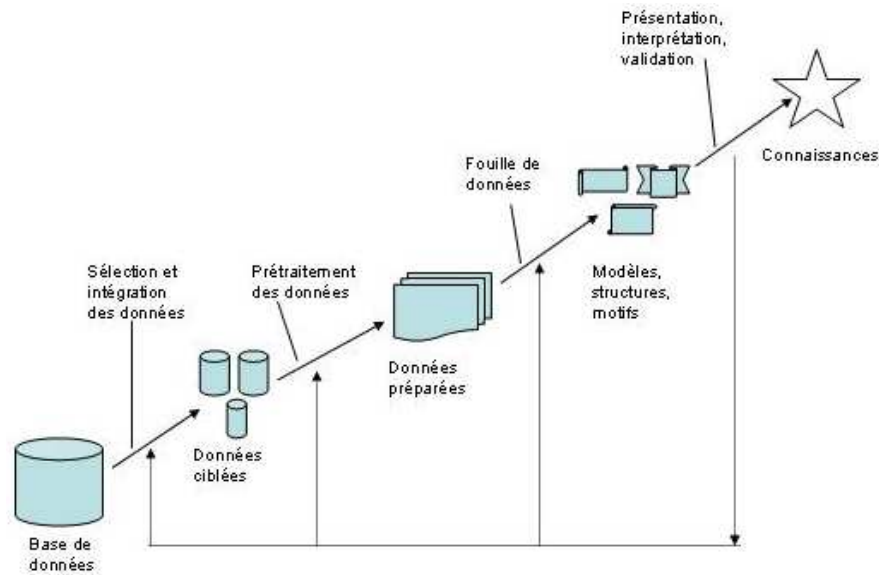


FIG. 1 – Étapes du processus d'extraction de connaissances à partir de données.

définition d'une ontologie du domaine d'exploitation des images. L'annotation manuelle peut s'avérer inopérante en raison d'un très grand nombre d'images à gérer. L'inférence de mots-clés [Aufaure et Hajji, 2002] constitue une solution à ce problème, mais peut altérer la précision de l'annotation. La description visuelle d'images fixes par contre est directement liée aux caractéristiques globales (couleur, texture, forme) ou locales (points d'intérêt) des images. Elle a pour avantage d'être automatique, ce qui convient bien aux grandes bases d'images.

Recherche d'information par le contenu

L'émergence des bibliothèques numériques et les échanges de l'information électronique suscitent un besoin de plus en plus croissant d'organisation et d'exploitation des grandes quantités de données. Le scénario classique d'exploitation d'un système de recherche d'information est le suivant : un utilisateur soumet une requête pour un besoin d'information et le système identifie les documents pertinents pour la requête soumise puis les retourne à l'utilisateur. Le but d'un système de recherche d'information étant de retrouver les documents pertinents par rapport à une requête donnée, plusieurs mesures d'évaluation existent dont les plus courantes sont le rappel et la précision. Le rappel définit la proportion de documents pertinents et effectivement retournés en réponse à une requête soumise au système. La précision est la proportion de documents retournés par le système et pertinents pour la requête considérée. Une étude des mesures d'évaluation des performances des systèmes de recherche d'information est faite dans [Kantor, 1994].

L'évaluation de la pertinence d'un document par rapport à une requête donnée n'est

pas toujours aisée, la notion de pertinence étant très dépendante des préférences de l'utilisateur qui soumet une requête. Cependant, les stratégies en recherche d'information calculent en général une mesure de similarité entre une requête et un document. Ces stratégies sont fondées, pour des documents textuels, sur l'hypothèse qu'un nombre élevé de termes communs entre une requête et un document est caractéristique de la pertinence du document pour la requête. Une stratégie de recherche se résume donc en un algorithme qui, étant donnés une requête Q et un ensemble de documents d_1, d_2, \dots, d_n , détermine le coefficient de similarité $s(Q, d_i)$, $1 \leq i \leq n$. La démarche est semblable pour la recherche d'images par le contenu. Cependant, il est moins évident de dire quelles sont les images pertinentes par rapport à une image-requête donnée. Dans des domaines spécifiques tels que l'imagerie médicale ou la détection de visages, il est relativement aisé de se constituer un ensemble d'images que l'on juge pertinentes pour une requête donnée. C'est ce qu'on appelle vérité terrain. Cependant, pour des images plus générales telles les images photographiques, la constitution de la vérité terrain devient moins évidente et on a recours à des artifices. Un exemple d'artifice consiste à faire subir à une image quelconque I_q un ensemble de transformations (translations, rotations, occultations partielles, variation de l'intensité lumineuse, *etc.*) pour constituer la vérité terrain associée à I_q qui sera par la suite considérée comme une image-requête. Cette artifice permet par exemple de tester la robustesse d'un descripteur. En général, dans les systèmes de recherche d'images par le contenu, on se réfère à la similarité visuelle pour déterminer les images pertinentes.

Problématique

La fouille d'images fixes implique un ensemble d'opérations relativement complexes telles que : i) l'extraction des descripteurs caractéristiques ; ii) la recherche par similarité ; iii) l'indexation en grande dimension ; iv) le passage à l'échelle de grands volumes de données ; v) la personnalisation de la recherche. Il existe de nombreux systèmes de recherche d'images fixes par le contenu [Niblack *et al.*, 1993, Kelly *et al.*, 1995, Smith et Chang, 1997, Shyu *et al.*, 1999, Dy *et al.*, 2003] et chaque système a une approche spécifique d'extraction de descripteurs. De façon générale toutefois, la recherche d'informations par le contenu dans de grandes bases de documents multimédias (vidéos, textes, images, *etc.*) exploite l'indexation de descriptions intermédiaires des documents dont les coûts de calculs peuvent être très importants. Pour un même type de données, l'ensemble des descripteurs potentiellement intéressants est souvent très grand. Considérons, pour illustrer ces propos, une base d'images fixes pouvant être décrite de différentes façons, notamment par des descripteurs visuels globaux de couleur, de texture, ou de forme (au niveau pixel). De nombreux descripteurs sont proposés dans la littérature [Manjunath *et al.*, 2002, Obeid *et al.*, 2001, Tao et Grosky, 1999] dont certains sont standardisés comme *MPEG-7*. Chacun d'eux est défini selon l'information que l'on souhaite extraire de l'image. On privilégiera par exemple un descripteur de couleur si l'on recherche des images de coucher du soleil, alors que pour retrouver toutes les images contenant un clavier, le descripteur de couleur ne sera d'aucune utilité et on privilégiera

le descripteur de forme plus approprié pour ce type de recherche et d'image. Et que dire d'une image contenant à la fois un coucher de soleil et un clavier d'ordinateur ? Il est possible, dans ce cas, d'utiliser à la fois un descripteur de forme et un de couleur. Mais, selon les objectifs de recherche de l'utilisateur, seul le descripteur de forme pourrait suffire à discriminer et à filtrer les images les plus similaires. Dans ce cas, il est inutile de vouloir faire la recherche sur tous les descripteurs pour la totalité de la base d'images.

La problématique de la recherche d'informations multimédias par le contenu est rendue plus complexe lorsque la base devient conséquente, par exemple plus de 10000 images avec des descripteurs de dimension supérieure à 7. La recherche est généralement menée de façon exhaustive sur la totalité de la base, sans doute du fait des schémas d'indexation relativement peu efficaces sur de très grands volumes de données. Ceci se traduit par des temps d'attente inacceptables pour l'utilisateur, jusqu'à ce que toutes les images de la base soient comparées avec l'image-requête selon une mesure de similarité. De plus, tout arrêt brutal de la requête en cours d'exécution a pour conséquence la perte de toutes les informations, obligeant l'utilisateur à relancer sa recherche. Par opposition aux bases de données traditionnelles [Manolescu, 2002, Gounaris *et al.*, 2002], très peu de travaux ont été menés sur l'optimisation et le traitement adaptatif de la recherche par le contenu dans un contexte multimédia. À notre connaissance, seuls quelques travaux sur les requêtes progressives ont été récemment proposés [Kiranyaz et Gabbouj, 2004], mais ils se limitent à lancer périodiquement une même sous-requête sur des portions d'une base de documents multimédias (images ou vidéos) sans ciblage *a priori* des descripteurs hétérogènes. Il convient donc de déployer des techniques d'indexation et de recherche adaptées aux grandes bases de données multimédias.

Objectif de la thèse

La fouille multimédia vise à extraire des grandes bases de données (bases multimédias, web, bases d'images, *etc.*), des connaissances pertinentes pour l'utilisateur dans ses activités de recherche d'informations ou de prises de décision. L'objectif de cette thèse est d'améliorer le temps et la qualité de la recherche par le contenu dans une grande base d'images fixes. Nous nous aidons en cela par les techniques de la fouille de données, principalement celles liées à la classification automatique (*clustering*) et à la découverte des règles d'association. Il s'agit d'exploiter, dans une chaîne de traitement, la connaissance sous forme de règles d'association issues des descripteurs d'une base d'images fixes pour améliorer leur interrogation.

L'extraction de connaissances à partir de données multimédias apporte une complexité nouvelle liée à la représentation des données analysées. La littérature actuelle fait état de diverses approches de description visuelle d'images fixes par le contenu. Parmi ces approches, certaines sont spécifiques à un domaine ou à une utilisation particulière [Shyu *et al.*, 1999, Dy *et al.*, 2003]. D'autres, par contre, sont généralistes [Niblack *et al.*, 1993, Smith et Chang, 1997, Manjunath *et al.*, 2002] et il n'est pas toujours facile pour un utilisateur de choisir celle la plus appropriée pour un contexte particulier de recherche par l'exemple. Il apparaît donc intéressant d'élaborer des stratégies

plus ou moins automatiques pouvant aider les utilisateurs à faire une bonne recherche par le contenu sans se préoccuper de la nature ou de la qualité des descripteurs. Nous proposons dans ce cadre une méthode de recherche d'images fixes par le contenu fondée sur une utilisation automatique, sélective, et progressive des descripteurs visuels globaux disponibles pour une base d'images donnée.

Synthèse des contributions

Notre travail a d'abord porté sur la proposition d'une stratégie de sélection automatique des critères de recherche par le contenu, fondée sur l'utilisation des règles d'association afin de réduire le nombre de descripteurs pour une recherche donnée. La stratégie proposée combine deux techniques de fouille de données : le *clustering* et la découverte des règles d'association. Ces techniques permettent d'améliorer notablement le temps de la recherche dans de grandes base d'images sans véritablement dégrader les résultats que l'on obtiendrait par une recherche séquentielle menée en aveugle.

Nous avons ensuite montré comment la recherche par le contenu peut être adaptée pour proposer des résultats intermédiaires qui sont progressivement fusionnés avec l'avantage, pour l'utilisateur, d'une part, de ne pas attendre que toute la base ait été parcourue et, d'autre part, de stopper la progression de la requête en cours d'exécution. Pour ce faire, nous avons introduit le concept d'éligibilité des *clusters* et nous avons mis en œuvre plusieurs stratégies de parcours des *clusters* d'une base d'images fixes. L'amélioration du temps de recherche est encore plus important avec l'introduction des règles de filtrage.

Enfin, pour valider notre approche de recherche d'images par le contenu, nous avons étudié l'intérêt de la combinaison de descripteurs globaux sur un problème de reconnaissance, et mesuré les taux de rappel/précision/*F-mesure* qui ont confirmé que la combinaison des descripteurs améliore le taux de reconnaissance.

Description des chapitres

Ce document décrit l'ensemble des travaux menés dans le cadre de cette thèse sur l'amélioration de la recherche d'images fixes par le contenu avec des techniques de la fouille de données. Il comporte quatre chapitres décrits comme suit :

- Le premier chapitre s'intéresse à l'optimisation des performances de la recherche par le contenu. Nous y précisons les principaux critères d'évaluation des systèmes de recherche, puis nous faisons un point sur les travaux liés aux index multidimensionnels, essentiels dans l'organisation des images fixes. Ces index posent de nombreux problèmes dont le plus connu pour les descripteurs d'images de grande dimension est le phénomène de la malédiction de la dimension qui altère les performances de la recherche par le contenu. Nous présentons les travaux qui tentent de contourner ce phénomène et permettent d'améliorer le temps de recherche. Nous nous intéressons enfin à la description des images et aux techniques d'amélioration de la qualité de la recherche par le contenu ;

- Le deuxième chapitre rappelle les principales tâches de la fouille de données et montre comment elles s’appliquent à l’organisation et à la recherche d’images fixes. Nous proposons dans ce chapitre une formalisation des règles d’association entre les *clusters* de descripteurs, puis nous montrons comment la classification non supervisée et la découverte de règles d’association peuvent être combinées pour organiser les descripteurs d’images ;
- Le troisième chapitre présente notre méthode de recherche par le contenu. La méthode exploite les règles d’association pour élaborer des stratégies d’exécution destinées à réduire le temps de la recherche par le contenu dans de grandes bases d’images fixes. Nous abordons également dans ce chapitre les aspects liés à la fusion des résultats issus de la recherche sur plusieurs types de descripteurs globaux *MPEG-7* ;
- Dans le quatrième chapitre, destiné à la présentation et à la discussion des résultats expérimentaux, nous présentons le contexte des évaluations, puis nous effectuons une série d’expérimentations qui permettent de valider notre méthode de recherche sur des bases d’images réelles.

La conclusion générale présente une synthèse des travaux effectués dans cette thèse. Elle décrit aussi les perspectives d’extension de la méthode de recherche d’images fixes proposée.

Chapitre 1

Optimisation des performances de la recherche par le contenu dans une base d'images fixes

Les possibilités technologiques actuelles en terme de capacité de stockage et de production des médias favorisent l'accumulation des images d'origines diverses (appareils photo numériques, webcam, etc.). Ainsi, de nombreux systèmes de recherche d'images sont proposés pour organiser et pour retrouver de plus en plus facilement les images stockées. Après avoir précisé les principaux critères d'évaluation de ces systèmes, nous faisons une synthèse de l'état de l'art sur les index multidimensionnels couramment utilisés dans l'organisation des images fixes. Nous décrivons également le phénomène de la malédiction de la dimension, puis nous présentons les travaux qui tentent de le contourner et permettent d'améliorer le temps de recherche. Les techniques d'amélioration de la qualité de la recherche par le contenu sont enfin présentées et s'appuient essentiellement sur la qualité de la description du contenu.

1.1 Introduction

La recherche d'images par le contenu émerge aujourd'hui comme un domaine de recherche important qui exploite de très grandes bases d'images. D'une façon générale les systèmes de recherche d'images par le contenu identifient à partir d'une requête (constituée d'une ou de plusieurs images) les images les plus similaires dans la base d'images. Leur conception peut être soit générique pour être applicable dans de nombreux domaines (images tout venant), soit spécialisée pour répondre à des besoins précis d'un domaine d'application (imagerie médicale, art, etc). Dans tous les cas, la recherche d'images par le contenu se fait en plusieurs étapes :

1. l'extraction automatique ou semi-automatique des informations qui caractérisent une image. Les informations extraites sont aussi appelées descripteurs de l'image. Elles se présentent généralement sous la forme d'un ou de plusieurs ensembles de valeurs numériques définissant des points dans un espace multidimensionnel appelé

espace de description. La définition et le calcul des descripteurs demandent des compétences en traitement du signal ou en vision par ordinateur ;

2. l'indexation (organisation) des informations extraites. Les compétences en bases de données sont nécessaires à cette étape pour concevoir des algorithmes efficaces de structuration des données. Les principes d'indexation traditionnelle telles que les arbres équilibrés, les techniques de hachage, *etc.* se révèlent inadaptés au caractère multidimensionnel des données [Nakamura *et al.*, 1993, Seeger et Kriegel, 1990];
3. l'utilisation des métriques pour mesurer la similarité de deux images. Cette similarité est supposée proportionnelle à celle des descripteurs des images et généralement évaluée par la distance entre les points définissant les descripteurs dans l'espace de description ;
4. la recherche proprement dite se traduisant le plus souvent par une recherche dans l'espace de description des images. Lorsqu'un utilisateur fait une recherche, celle-ci s'effectue par calcul de similarité entre la requête et les images dont les descripteurs sont indexés dans la base de données. Cette recherche par similarité peut s'effectuer dans toute l'image ou dans certaines parties en privilégiant éventuellement certaines caractéristiques des images comme la couleur, la texture, ou la forme. Elle peut aussi se faire localement par des points d'intérêt.

La recherche d'images fixes par le contenu se situe donc à la croisée de plusieurs domaines tels que la vision par ordinateur, le traitement d'images et les bases de données.

Les trois premières étapes de la recherche d'images ne nécessitent pas qu'une requête soit soumise au système. Toutefois les performances de la recherche à l'étape 4 dépendent à la fois de la qualité de la description des images, de l'organisation des descripteurs et de la mesure de similarité choisie pour comparer les images. C'est pourquoi nous commençons par préciser dans ce chapitre les principaux critères d'évaluation des systèmes de recherche d'images par le contenu ainsi que les mécanismes de la similarité visuelle.

Les index multidimensionnels organisent les données pour que les procédures de recherche par le contenu s'exécutent le plus rapidement possible. Pour ce faire, ces index regroupent les vecteurs de l'espace multidimensionnel de description en paquets moins nombreux que les vecteurs et dont les propriétés les rendent beaucoup plus facilement accessibles par les algorithmes de recherche. Ces index ne sont cependant pas toujours bien adaptés à la recherche lorsque la dimension des vecteurs croît (à partir de 16 selon [Weber *et al.*, 1998]). Nous passons donc en revue les techniques déployées pour contourner ce problème afin d'accélérer les temps de recherche.

Nous expliquons dans ce chapitre pourquoi nous nous sommes limités tout au long de ce travail à l'exploitation des descripteurs visuels globaux de couleur, de texture, et de forme pour la caractérisation des images par le contenu. Nous rappelons également le principe général de ce type de description et nous présentons les techniques d'amélioration de la qualité de la recherche par le contenu qui dépendent essentiellement de la qualité de la description du contenu.

1.2 Évaluation des performances des systèmes de recherche d'images

L'évaluation des performances des systèmes de recherche d'images permet de comparer différents algorithmes, en général sur la base de leur pouvoir de reconnaissance et sur la base de leur rapidité. La rapidité des algorithmes se mesure par le temps moyen de réponse du système de recherche sur plusieurs requêtes. L'évaluation du pouvoir de reconnaissance est cependant liée à la satisfaction de l'utilisateur et sa mise en place suscite de nombreux scénarii de recherche pour être représentative. Contrairement aux bases de données transactionnelles, il est difficile de fixer la sortie exacte d'un système de recherche d'images. Cette situation complique l'élaboration des ensembles de test (*benchmark sets*). D'autre part, la description d'une image peut ne pas toujours coïncider avec l'interprétation qu'un utilisateur se fait dans une situation donnée et on parle dans ce cas d'un « fossé sémantique » entre l'image considérée et sa description. La comparaison des systèmes de recherche d'images par le contenu est finalement difficile à faire parce que les expérimentations ne visent pas toujours les mêmes objectifs, même si le même ensemble de test est utilisé.

De façon générale, les systèmes de recherche d'images fixes sont comparés sur la base de leur rappel et de leur précision dont nous rappelons la définition ci-après.

Définition 1.1 (Rappel et précision) Soit I_q une image-requête, $R(I_q)$ l'ensemble des images pertinentes par rapport à I_q et $A(I_q)$ l'ensemble des images retournées en résultat par un système de recherche. Le rappel du résultat $A(I_q)$ par rapport à l'ensemble des images pertinentes $R(I_q)$ est donnée par :

$$rap = \frac{|A(I_q) \cap R(I_q)|}{|R(I_q)|} \quad (1.1)$$

et sa précision par :

$$prec = \frac{|A(I_q) \cap R(I_q)|}{|A(I_q)|} \quad (1.2)$$

$|E|$ désigne la cardinalité de l'ensemble E .

Le rappel définit donc la proportion d'images pertinentes et effectivement retournées en réponse à une requête soumise au système. La précision est la proportion d'images retournées par le système et pertinentes pour la requête considérée. La notion d'images pertinentes par rapport à une image-requête est subjective. Dans la pratique, et en fonction du domaine dans lequel la recherche d'images est faite, on désigne les images susceptibles d'être pertinentes par rapport à une image-requête pour constituer un ensemble appelé vérité terrain. Le rappel et la précision peuvent être combinés en une seule mesure appelée la F-mesure [Rijsbergen, 1979] et définie par :

$$F\text{-mesure} = \frac{2 \times rap \times prec}{rap + prec} \quad (1.3)$$

La recherche par le contenu étant un processus interactif, certains aspects de l'évaluation des performances tels que le nombre moyen d'itérations pour obtenir un résultat

satisfaisant sont importants. Dans des domaines spécifiques, on est souvent amené à définir des critères *ad-hoc* de contrôle de performance des systèmes.

Le standard *MPEG-7* évalue les descripteurs qu'il propose à travers des requêtes par l'exemple. Pour garantir l'objectivité des évaluations, il élabore un certain nombre de mesures sur la base de la spécification de l'ensemble des données, un ensemble de requêtes ainsi que les résultats correspondant à la vérité terrain [Manjunath *et al.*, 2002]. La principale mesure proposée par le standard tient à la fois compte du nombre d'images pertinentes retournées et de l'ordre de ces images dans la liste des résultats. Elle est définie par l'équation suivante :

$$NMRR = \frac{\left(\sum_{k=1}^{NG(I_q)} \frac{Rank(k)}{NG(I_q)} \right) - 0,5 - \frac{NG(I_q)}{2}}{K(I_q) + 0,5 - 0,5 \times NG(I_q)} \quad (1.4)$$

où $NG(I_q)$ est la taille de la vérité terrain associée à l'image-requête I_q , $Rank(k)$ est la valeur du classement de la $k^{\text{ième}}$ image de la vérité terrain par l'algorithme de recherche et $K(I_q)$ la valeur du classement valide associé à l'image-requête I_q . Cette valeur dépend de la taille de la vérité terrain. $NMRR$ varie de 0 à 1. Plus sa valeur est petite, meilleure est l'efficacité de la recherche. Il est évident que l'équation (1.4) n'aura d'intérêt pratique qu'au cas où les images de la vérité terrain sont ordonnées, ce qui n'est en général pas le cas.

Nous nous intéressons dans ce chapitre aux travaux menés dans le sens de l'amélioration du temps de la recherche, puis à ceux qui s'intéressent plutôt à la qualité des résultats retournés. Toutefois, l'utilisateur final reste le seul capable de juger de la pertinence des résultats en fonction de ses objectifs et de ce qu'il perçoit.

1.3 Similarité visuelle

L'œil humain possède incontestablement les facultés de traiter efficacement l'information visuelle et d'identifier les images similaires à partir de leur contenu. On distingue habituellement deux modes de fonctionnement de la perception humaine : la perception pré-attentive et la perception attentive [Bimbo, 1999]. La perception attentive a un lien direct avec l'interprétation. Elle suppose une connaissance *a priori* et un raisonnement. Son utilisation est requise dans des domaines spécifiques tels que la détection de visages, l'imagerie médicale ou mécanique. Ces domaines comparent des objets et définissent des critères de similarités qui leur sont propres. La perception pré-attentive, quant à elle, est tout simplement basée sur la similarité entre les signaux (stimuli) perçus par l'œil, sans aucune forme d'interprétation. Elle est utile dans des applications plus générales pour lesquelles la couleur, la texture, la forme ou la perception des relations spatiales est plus importante. Plusieurs modèles tentent de reproduire automatiquement la perception humaine de la similarité des signaux sensoriels. Parmi les plus importants, on cite le modèle de Tversky [Tversky, 1977] et le modèle métrique [Torgeson, 1965].

D'après le modèle de Tversky [Tversky, 1977], un stimulus est un ensemble de descripteurs et la définition de la similarité est inspirée de la théorie des ensembles. La similarité entre deux stimuli est obtenue par combinaison linéaire d'une fonction de

deux types de descripteurs : ceux qui décrivent les deux stimuli et ceux qui décrivent un stimulus mais ne décrivent pas l'autre. Le modèle tient compte des descripteurs binaires qui se comportent comme des prédicats pour certains stimuli. Santini et Jain montrent que le modèle de Tversky peut être utilisé pour modéliser les distances entre les textures [Santini et Jain, 1995], mais cette approche n'est pas intéressante pour l'indexation. Nous décrivons dans cette section le modèle métrique puis nous présentons les principaux concepts de la recherche par similarité.

1.3.1 Le modèle métrique

La perception humaine de la similarité peut être modélisée par une mesure de distance appropriée dans un espace métrique multidimensionnel. Dans un tel espace, les composantes d'un point décrivent les propriétés du stimulus perçu par l'œil humain. Si d est une fonction de distance, s_1 , s_2 , et s_3 sont des stimuli dans un espace métrique E , alors les axiomes suivants doivent être vérifiés :

- positivité stricte : $d(s_1, s_2) > 0$ si $s_1 \neq s_2$ et $d(s_1, s_2) = 0$ si $s_1 = s_2$
- symétrie : $d(s_1, s_2) = d(s_2, s_1)$
- inégalité triangulaire : $d(s_1, s_3) \leq d(s_1, s_2) + d(s_2, s_3)$

Soit n la dimension de l'espace métrique E . La fonction de distance la plus souvent utilisée est celle de Minkowsky [Minkowsky, 1896] dont la forme générale est :

$$d(s_1, s_2) = \left(\sum_{j=1}^n \mu_j |x_1[j] - x_2[j]|^p \right)^{1/p}$$

où x_1 et x_2 sont les coordonnées des stimuli s_1 et s_2 dans l'espace E , μ_j , $j = 1..n$ et p des nombres positifs quelconques de \mathbb{R}^+ .

Si $p = 2$ et $\mu_j = 1$, $j = 1..n$, $d(s_1, s_2) = \sqrt{\sum_{j=1}^n (x_1[j] - x_2[j])^2}$: c'est la distance euclidienne classique.

Si $p = 1$ et $\mu_j = 1$, $j = 1..n$, $d(s_1, s_2) = \sum_{j=1}^n |x_1[j] - x_2[j]|$. Cette distance est connue sous le nom de distance de *Manhattan*. La distance de *Manhattan* conduit dans la plupart des cas à des résultats semblables à ceux de la distance euclidienne classique et l'effet des larges différences est atténué, puisqu'elles ne sont pas élevées au carré.

Le modèle métrique a beaucoup d'avantages qui impliquent sont utilisation courante dans la recherche d'information visuelle. Parmi ces nombreux avantages, on cite [Bimbo, 1999] :

- l'adaptation à la nature multidimensionnelle des descripteurs utilisés dans les domaines de reconnaissance de modèles et de vision par ordinateur pour lesquels de nombreux algorithmes de calcul de descripteurs sont disponibles ;
- la bonne approximation de certains jugements de perception de similarité tels que ceux liés à la couleur ;
- la possibilité de concevoir des méthodes d'accès multidimensionnel aux images si celles-ci sont représentées par des vecteurs. En effet, des index peuvent être construits sur des vecteurs et sont très utiles pour accéder de grandes bases d'images.

1.3.2 Recherche par similarité

Étant donnée une image-requête, la recherche d'images semblables à la requête dans une base d'images est le mode de recherche le plus courant en recherche d'information visuelle. On parle de recherche par similarité, fondamentalement différente de la recherche exacte dans des bases de données classiques. La recherche par similarité permet donc de retrouver les images les plus similaires à une image-requête au sens d'une mesure de similarité donnée. Elle peut se faire soit par une distance aux proches voisins, soit à ε -près. La recherche de k -plus proches voisins consiste à retrouver les k images les plus proches de l'image-requête au sens de la mesure de similarité associée aux descripteurs d'images. Dans la recherche à ε -près, il s'agit de retrouver les images situées à une distance d'au plus ε de l'image-requête. Ce type de recherche permet à un utilisateur familier de ses images de choisir une valeur appropriée de ε pour éliminer les images trop éloignées de la requête. Par contre, une valeur trop petite de ε peut conduire à des listes résultats vides alors qu'une valeur trop grande de ε peut donner des listes résultats de très grande taille.

La recherche de k -plus proches voisins garantit automatiquement k images dans la liste des résultats mais certaines images de la liste peuvent être très éloignées de la requête pour être considérées comme similaires. La recherche de k -plus proches voisins est souvent préférée à la recherche à ε -près puis qu'elle ne nécessite pas de connaissances *a priori* sur la distribution des descripteurs d'images dans l'espace multidimensionnel. Dans la suite de ce travail, nous considérons uniquement la recherche de plus proches voisins.

Le moyen le plus simple pour mener une recherche par similarité dans une base d'images est le parcours séquentiel. Cette approche consiste à comparer chaque image de la base à une image-requête pour évaluer la similarité et décider de celles qui seront effectivement retournées comme résultat. Le parcours séquentiel, malgré sa simplicité, est très coûteux pour de grandes bases d'images puisque le temps de parcours est proportionnel à la taille de la base. Pour remédier à ce problème, il existe deux méthodes de réduction de la complexité de recherche. L'une des méthodes consiste à utiliser des techniques de filtrage et l'autre s'appuie sur les techniques d'accès aux données. Les techniques de filtrage définissent une nouvelle distance minorant la distance originale tout en étant plus simple à calculer. Cette nouvelle distance est utilisée pour filtrer plus rapidement les images non pertinentes. Les images non filtrées (images candidates) seront ensuite comparées à la requête avec la distance originale.

Les méthodes d'accès aux données utilisent des structures de données et des combinaisons de techniques pour organiser les descripteurs visuels afin de gérer efficacement le processus de recherche. Ces méthodes divisent l'espace de recherche en plusieurs sous-espaces et ne parcourent qu'un sous-ensemble très réduit de ces sous-espaces. Les méthodes d'accès aux données se scindent en deux principales classes : les méthodes multidimensionnelles et les méthodes métriques.

Les méthodes métriques utilisent tous simplement les distances relatives entre objets dans un espace métrique. Elles ne tiennent donc pas compte de la position absolue des objets dans l'espace qu'elles partitionnent en régions par regroupement d'objets

autour d'un objet représentatif appelé pivot [Chavez *et al.*, 2001]. Ces méthodes posent le problème du choix des pivots ainsi que de leur nombre. De plus, les algorithmes de recherche sont très consommateurs de mémoire et s'adaptent mal sur de grandes bases de données.

1.4 Index multidimensionnels

L'accumulation des images numériques implique la nécessité de développer des moyens de les organiser pour faciliter leur exploitation. Le parcours séquentiel de gigantesques bases d'images lors d'une recherche par le contenu pourra ainsi être évité. Les index multidimensionnels organisent les images numériques sous forme de descripteurs (un descripteur étant un vecteur de réels pouvant atteindre plusieurs centaines de dimensions). Ces index offrent la possibilité de filtrer toutes les images non pertinentes par comparaison de leurs descripteurs à ceux d'une requête donnée. Les techniques d'indexation adaptées à l'organisation d'une base d'images fixes reposent principalement sur le partitionnement de données et sur le partitionnement de l'espace dont nous rappelons dans cette section les principes généraux ainsi que les travaux qui s'y réfèrent. Un panorama détaillé de toutes ces techniques d'indexation est présenté dans [Berrani *et al.*, 2002].

1.4.1 Partitionnement de données

Les techniques basées sur le partitionnement de données sont toutes dérivées du R-Tree [Guttman, 1984] et procèdent par un regroupement des vecteurs selon leur proximité relative dans l'espace. Les vecteurs sont englobés dans une forme géométrique simple (hyper-rectangle, par exemple) et le tout est organisé sous forme d'un arbre dans lequel les vecteurs sont stockés dans les feuilles alors que les formes englobantes sont stockées dans les nœuds. Le principal problème du R-Tree [Guttman, 1984] est la tendance des feuilles à se chevaucher, obligeant le parcours de plusieurs nœuds pour retrouver une image. La recherche est ainsi rendue inefficace. Plusieurs améliorations du R-Tree tentent de trouver des solutions à ce problème [Sellis *et al.*, 1987, Bechmann *et al.*, 1990]. Le R*-Tree [Bechmann *et al.*, 1990] partage la même structure que le R-Tree, mais améliore les performances des requêtes par l'introduction des algorithmes d'insertion et de division qui minimisent le recouvrement des nœuds. Le SS-Tree [White et Jain, 1996] reprend le principe du R-Tree, mais les rectangles sont remplacés par des sphères nécessitant moins d'information pour leur description. Par contre, les recouvrements aux feuilles du SS-Tree sont plus importants puisque les sphères englobantes sont beaucoup plus volumineuses que les rectangles englobants équivalents du R-Tree ou du R*-Tree.

Afin d'exploiter les avantages du R-Tree et du SS-Tree, Katayama et Satoh proposent de combiner les sphères et les rectangles englobants au sein d'une même structure : le SR-Tree [Katayama et Satoh, 1997]. Chaque feuille du SR-Tree est l'intersection d'une hypersphère et d'un hyper-rectangle. Les régions construites sont donc plus compactes et améliorent les algorithmes de recherche. D'après l'étude comparative de Wu et Bretschneider sur les techniques d'indexation dérivées du R-Tree [Wu et Bretschneider, 2004], le SR-Tree permet de réaliser les meilleures performances dans la recherche ; le SS-Tree

a de très bonnes propriétés dynamiques avec notamment des coûts très bas à l'insertion ; le R-Tree ne demande que très peu d'espace disque pour la sauvegarde des fichiers d'index et le R*-Tree a de bonnes performances si l'on évalue à la fois l'espace disque utilisé et le coût de la recherche. En définitive, le choix de la technique d'indexation la plus adaptée dépend du facteur à optimiser, mais surtout des objectifs à atteindre par le système d'exploitation des images. Cependant, la tendance est beaucoup plus à l'optimisation des performances des requêtes.

1.4.2 Partitionnement de l'espace

L'indexation multidimensionnelle basée sur le partitionnement de l'espace ne tient pas compte de la distribution des points. Elle partitionne l'espace en cellules disjointes plus ou moins régulières. Les cellules générées peuvent être organisées soit par une table de hachage comme c'est le cas du GridFile [Nievergelt *et al.*, 1984], soit de façon arborescente comme dans le cas du k-d-Tree [Bentley, 1979]. Les cellules définies par des tables de hachage minimisent les accès disque et sont très adaptées aux requêtes par intervalle. Le principal risque toutefois est que plusieurs pointeurs de la table de hachage peuvent référencer un même bloc de données sur disque, créant ainsi des accès redondants. De plus, les risques de multiplication de cellules vides sont importants.

Le k-d-Tree [Bentley, 1979] est un arbre binaire de recherche qui organise les points d'un espace de dimension k . Il divise à chaque nœud intermédiaire l'espace en deux parties par un hyperplan de dimension $k - 1$ contenant au moins un point utilisé par la suite pour représenter l'hyperplan. La dimension suivant laquelle la division est faite change d'un niveau de l'arbre à l'autre dans la limite des k dimensions possibles. La recherche et l'insertion de nouveaux nœuds dans le k-d-Tree sont aisées mais la suppression peut être plus compliquée en raison du problème de réorganisation de l'arbre qu'elle pose. D'autre part, la structure de l'arbre dépend fortement de l'ordre d'insertion des points dont les positions définissent les hyperplans séparateurs de région. Il en résulte des arbres non équilibrés. Friedman *et al.* proposent de construire un k-d-Tree adaptatif pour avoir des arbres équilibrés [Friedman *et al.*, 1977]. Dans le k-d-Tree adaptatif, les hyperplans sont toujours parallèles aux axes mais peuvent ne pas contenir de points et ne changent pas obligatoirement d'un nœud au nœud de niveau inférieur. Le nombre de points aux feuilles est fixé et tout dépassement de ce nombre entraîne automatiquement un éclatement du nœud. Le k-d-Tree est une structure statique et la maintenance de l'équilibre de l'arbre peut s'avérer très lourde si les insertions et les suppressions sont régulières. Le k-d-Tree et le k-d-Tree adaptatif ne gèrent pas de grandes quantités de données. Les arbres qu'ils construisent ne s'appuient que sur des structures en mémoire centrale. De nombreuses extensions telles que le K-D-B-Tree [Robinson, 1981], le LSD-Tree [Henrich *et al.*, 1989] et le LSD^h-Tree [Henrich, 1998] sont alors introduites pour gérer les entrées/sorties sur disque. Elles reprennent toutes le principe du k-d-Tree mais intègrent en plus la gestion des pages de données.

1.4.3 Malédiction de la dimension

La malédiction de la dimension fait référence aux difficultés de gestion et de traitement des données qui apparaissent dans des espaces de grande dimension. Il est en effet difficile d'étendre dans ces espaces les techniques que l'on a dans les espaces à deux ou à trois dimensions. D'après Beyer *et al.*, sous certaines conditions sur la distribution des données, les vecteurs ont tendance à devenir équidistants lorsque la dimension augmente [Beyer *et al.*, 1999]. Ce curieux phénomène entraîne l'inutilité des index multidimensionnels pour la recherche par similarité et le manque de pertinence des résultats retournés. Dans la pratique cependant, les données ne satisfont généralement pas les conditions évoquées dans [Beyer *et al.*, 1999], nécessaires à l'apparition du phénomène. Les techniques d'indexation fondées sur le partitionnement de l'espace sont particulièrement inefficaces en raison de la croissance exponentielle du nombre de cellules en fonction de la dimension, et en raison d'un très grand nombre de cellules voisines d'une cellule donnée (phénomène observé au delà de 60 dimensions). Les algorithmes de recherche sont ainsi forcés à parcourir toutes les cellules voisines à la recherche des plus proches voisins d'une requête donnée. Les techniques d'indexation qui partitionnent les données calculent des paquets de forme géométrique simple comme les hyper-sphères, les hyper-cubes ou les hyper-rectangles. Toutefois, les propriétés géométriques de ces formes ne sont plus tout-à-fait maîtrisables en grande dimension, rendant ainsi difficile le choix des formes. De plus, le phénomène de l'espace vide [Scott et Thompson, 1983] complique le regroupement des vecteurs en paquets car le nombre de vecteurs n'est pas suffisamment élevé par rapport au nombre de dimension. La conséquence est que les algorithmes finissent par construire des formes très volumineuses qui se chevauchent et avec très peu de vecteurs. Les performances des index sont ainsi réduites.

Plusieurs auteurs se sont intéressés au problème de la malédiction de la dimension. Weber *et al.* [Weber *et al.*, 1998] montrent par exemple que dans un espace dont la dimension est supérieure à 16, la recherche séquentielle et exhaustive est meilleure que toutes les techniques d'indexation multidimensionnelle évoquées. La thèse de Berrani propose une étude plus fine du problème de la malédiction de la dimension [Berrani, 2004]. Elle détaille les méthodes de réduction de la dimension couramment utilisées pour faire face à ce problème et montre que leur utilisation n'est pas triviale et reste discutable. Alors que les techniques d'indexation restent adaptées à des vecteurs de petite dimension (inférieure à 16), les performances d'une recherche par le contenu se faisant par calcul d'une distance aux plus proches voisins sont très rapidement dégradées lorsque la dimension des vecteurs des descripteurs devient grande. On est donc amené à définir de nouvelles structures d'indexation et de nouvelles techniques de recherche adaptées aux vecteurs de grande dimension.

1.5 Amélioration du temps de recherche

Plusieurs techniques ont été proposées pour résoudre les problèmes liés aux index multidimensionnels et ainsi améliorer le temps de recherche. Ces techniques permettent de contourner le phénomène de la malédiction de la dimension qui rend finalement une

recherche séquentielle bien meilleure qu’une recherche sur les index multidimensionnels. Elles comportent deux grandes variantes. La première propose un codage particulier de la description des données qui permet d’améliorer la recherche séquentielle par des techniques de filtrage. La seconde consiste à accélérer la recherche de plus proches voisins en y introduisant une approximation. Il s’agit, dans ce cas, de trouver un compromis entre une réduction du temps de réponse et l’imprécision des résultats retournés.

1.5.1 Les techniques de filtrage

Une technique de filtrage effectue d’abord une recherche séquentielle sur une représentation compressée des vecteurs de descripteurs. Cette étape n’est pas coûteuse puisqu’elle est exécutée sur un ensemble de vecteurs de taille beaucoup plus réduite que l’ensemble original et permet d’éliminer (filtrer) les vecteurs ne pouvant pas figurer parmi les k -plus proches voisins recherchés. La technique accède ensuite aux descriptions originales des vecteurs à partir des identifiants des vecteurs retenus à la première étape pour construire l’ensemble des résultats. Le VA-File [Weber *et al.*, 1998] est une des techniques qui améliore la recherche séquentielle par des techniques de filtrage. Son principe de base repose sur la gestion de deux ensembles de données : un fichier qui contient tous les vecteurs de la base et un autre qui contient des approximations géométriques de ces vecteurs. Les approximations sont obtenues en partitionnant chaque dimension d_i en 2^{b_i} intervalles codés chacun sur b_i bits. Les intervalles sont calculés de façon à contenir le même nombre de vecteurs. À chaque cellule obtenue, ainsi qu’aux vecteurs qu’elle contient, est associé un code de longueur $\sum_{i=1}^d b_i$ où d est la dimension des vecteurs.

Les performances du VA-File dépendent essentiellement de la taille du fichier des approximations et du taux de filtrage. Elles se dégradent et deviennent à leur tour plus mauvaise que la recherche séquentielle si le fichier des approximations ne peut plus tenir en mémoire [Amsaleg et Gros, 2001]. Le taux de filtrage dépend du nombre de bits utilisés pour coder un vecteur. Les performances du VA-File ne peuvent être améliorées que par une augmentation du nombre de bits utilisés lors du codage [Cha *et al.*, 2002]. Mais une telle augmentation peut entraîner un surcoût lors du stockage et du traitement du fichier des approximations. Face à ce constat, Cha *et al.* [Cha *et al.*, 2002] proposent le LPC-File où les informations codées sont enrichies par l’introduction des coordonnées polaires des vecteurs lors du calcul des approximations. Cette nouvelle technique donne un taux de filtrage meilleur que celui du VA-File puisque les approximations sont plus précises. Par contre, les calculs sont plus coûteux. Le choix du nombre de bits de codage reste un problème et les codes sont plus long que ceux du VA-File.

Balko *et al.* proposent une méthode appelée **AV** (*Active Vertice Method*) pour l’indexation multidimensionnelle [Balko *et al.*, 2004]. Cette méthode reprend les techniques basées sur les arbres quaternaires [Samet, 1984] pour le partitionnement de l’espace. Tout comme le VA-File, la méthode AV construit les approximations des vecteurs dans l’espace et associe à ces approximations des codes binaires. Cependant, les cellules régulières du VA-File sont remplacées dans la méthode AV par un arbre non équilibré pour refléter la distribution des vecteurs dans l’espace. Chaque nœud n_i de l’arbre détermine

une région de l'espace décrite par un point représentatif c_i et délimitée par l'intersection d'une ellipsoïde et d'un hyper-rectangle. Le rayon maximum de l'ellipsoïde est un paramètre fixé au début de l'indexation. La taille de l'hyper-rectangle dépend du niveau du nœud délimité par l'hyper-rectangle. Elle est réduite de moitié lorsqu'on passe d'un nœud donné aux nœuds de niveau directement inférieur en cas de distribution uniforme des données. Le code binaire de c_i est entièrement défini par le chemin parcouru pour aller de la racine de l'arbre au nœud contenant c_i . Sa longueur est donc variable. La délimitation des régions par l'intersection d'une ellipsoïde et d'un hyper-rectangle dans la méthode AV permet d'obtenir des approximations plus rigoureuses de la distance entre un vecteur requête quelconque et un vecteur donné de l'espace. Balko *et al.* montrent d'ailleurs que les performances de la méthode AV sont meilleures que celles du VA-File et du LPC-File [Balko *et al.*, 2004]. La principale difficulté de la méthode réside dans le choix du rayon maximum et dans la complexité de calcul des rayons de l'ellipsoïde englobante d'un nœud.

1.5.2 Recherche approximative de plus proches voisins

La dégradation des performances de la recherche exacte de plus proches voisins est liée à la grande dimension des vecteurs (au delà de 7 composantes). Un moyen d'améliorer les temps de recherche consiste à recourir aux méthodes approximatives pour accélérer la recherche au détriment de la qualité des résultats. Les approximations sont introduites soit au niveau de la représentation des données, soit au niveau de la technique de recherche elle-même.

Approximation de la représentation des données. L'approximation de la représentation des données consiste à transformer les vecteurs dans un espace de dimension plus petite et à effectuer la recherche dans l'espace transformé. Les techniques de transformation les plus souvent utilisées sont celle de l'analyse en composantes principales [Chakrabarti et Mehrotra, 2000]. En fonction du domaine étudié, on parle de décomposition en valeurs singulières (SVD) ou de transformée de Karhunen-Loeve (KLT). Ces transformations sont calculées en fonction de la distribution des vecteurs dans l'espace. Toute évolution du contenu de la base nécessite donc de calculer de nouveau les coordonnées des vecteurs dans l'espace transformé et de reconstruire l'index. La mise à jour peut donc être très coûteuse si la base d'images évolue régulièrement. De plus, la transformation ne convient qu'aux bases d'images qui utilisent la distance euclidienne comme mesure de similarité.

Le hachage multidimensionnel [Indyk et Motwani, 1998, Gionis *et al.*, 1999] offre un moyen de faire une recherche sur des points obtenus par projection des vecteurs de l'espace original. L'idée consiste à transformer les vecteurs en utilisant plusieurs fonctions de hachage dans le but de maximiser la probabilité que des vecteurs proches de l'espace original se retrouvent dans le même paquet. Plusieurs transformations sont appliquées aux vecteurs et chacune d'elles associe aux vecteurs proches la même valeur. La mise en œuvre du hachage multidimensionnel proposée dans [Gionis *et al.*, 1999] procède d'abord par une transformation des vecteurs de la base vers un cube de Hamming. La

distance de Hamming est ensuite utilisée comme mesure de similarité et correspond à la distance L_1 dans l'espace original. De ce fait, l'approche ne peut s'appliquer qu'aux images dont la similarité est estimée par la distance L_1 .

Approximation de la technique de recherche. Nous distinguons globalement deux moyens d'approximer la recherche pour gagner en temps de réponse. Le premier moyen consiste à ramener le problème de la recherche des plus proches voisins en celui d'une recherche à ε -près dans des structures d'indexation adaptées, puis à estimer le rayon ε de recherche. C'est l'approche que Lang et Singh proposent dans [Lang et Singh, 2002]. Dans leur approximation de la recherche, Lang et Singh estiment dans un premier temps le rayon ε par une recherche de k -plus proches voisins sur un échantillon de vecteurs qui tient en mémoire centrale. Le rayon estimé est ensuite corrigé par un facteur pour minimiser sa dépendance à l'échantillon choisi, puis une recherche à ε -près est effectuée sur l'ensemble de tous les vecteurs de la base de données avec le rayon corrigé.

Soit R le résultat de cette recherche. La dernière étape de la méthode de Lang et Singh [Lang et Singh, 2002] consiste donc à faire une recherche de k -plus proches voisins dans R , puis à retourner les résultats obtenus comme une approximation de la recherche de k -plus proches voisins sur l'ensemble de tous les vecteurs de la base de données. La technique proposée réduit les coûts des entrées/sorties pendant la recherche mais la dépendance des résultats approximatifs à l'échantillon choisi au départ constitue une limite importante de la méthode. En effet, d'après les expérimentations présentées dans [Lang et Singh, 2002], plus la taille de l'échantillon est petite, plus l'erreur commise dans l'estimation de ε est grande. Cette erreur croît également avec nombre de plus proches voisins recherchés. La taille de R n'est pas contrôlée. Il peut donc arriver qu'elle soit inférieure au nombre de plus proches voisins à calculer.

Le second moyen d'approximer la recherche de plus proches voisins, beaucoup plus direct que le premier, consiste à exploiter les propriétés des structures d'index dans le processus de recherche. Lorsque la structure d'index est constituée de plusieurs cellules, les algorithmes de recherche de plus proches voisins peuvent identifier dans un premier temps les cellules pouvant contenir les vecteurs résultats et les ordonner suivant des critères plus ou moins liés au vecteur requête. Ces algorithmes accéderont ensuite aux cellules les unes après les autres et arrêteront la recherche après le parcours d'un nombre de cellules arbitrairement fixé. C'est le principe des arrêts prématurés repris dans [Weber et Böhm, 2000] à travers la version approximative du VA-File et dans [Li *et al.*, 2002] à travers CLINDEX. La taille géométrique des cellules d'index peut aussi rentrer en jeu dans le processus d'approximation de la recherche. Elle permet d'élaborer des propriétés sous forme de règles pour identifier les cellules à éliminer de la recherche ou pour arrêter la recherche.

Soit I_q une image-requête, C_i et C_j deux cellules quelconques de l'index, $dmin(I_q, C_i)$ la distance minimale entre I_q et C_i , $dmax(I_q, C_j)$ la distance maximale entre I_q et C_j . On suppose que C_j contient au moins k vecteurs où k est le nombre de plus proches voisins recherchés. Une des règles consiste à éliminer C_i de la recherche si $dmin(I_q, C_i) \geq dmax(I_q, C_j)$. Cette règle est appliquée au début de la recherche pour éliminer (filtrer)

les cellules ne contenant pas les plus proches voisins recherchés. La suite de la recherche ordonne, puis traite les cellules restantes selon leur distance croissante à I_q . Au cours de cette seconde phase, les plus proches voisins, et en particulier le $k^{\text{ième}}$ que nous notons ppv_k , sont modifiés au fur et à mesure du traitement des cellules.

La seconde règle permet d'arrêter la recherche dès que $dmin(I_q, C_i) \geq d(I_q, ppv_k)$ avec $d(I_q, ppv_k)$ la distance entre I_q et ppv_k . L'efficacité des deux règles énoncées est liée à la dimension des vecteurs. Elle est fortement réduite en grande dimension où les cellules ont tendance à se chevaucher. Pour renforcer l'efficacité des règles, certaines méthodes considèrent les cellules plus petites qu'elles ne le sont en réalité pour réduire le risque de chevauchement [Weber et Böhm, 2000]. Elles introduisent pour cela un taux d'imprécision ε qu'elles contrôlent par la suite. Il s'agit d'une approximation géométrique. Les règles sont alors transformées comme suit :

$$\begin{aligned} &\text{si } dmin(I_q, C_i) + \varepsilon \geq dmax(I_q, C_j) - \varepsilon \text{ alors éliminer } C_i \\ &\text{si } dmin(I_q, C_i) + \varepsilon \geq d(I_q, ppv_k) \text{ alors arrêter la recherche} \end{aligned}$$

Les méthodes d'approximation géométrique reposent donc sur un principe simple mais le choix de ε est difficile pour faire un bon compromis entre le temps de réponse et la précision de la recherche. Un petit ε préserve la précision mais ne garantit pas toujours une forte réduction du temps de réponse. À l'inverse, une très grande valeur de ε permet de réduire significativement le temps de réponse mais n'assure pas une bonne précision.

D'autres critères existent pour arrêter la recherche. DBIN [Bennett *et al.*, 1999] et le *P-Sphere Tree* [Goldstein et Ramakrishnan, 2000] utilisent des approches probabilistes pour approximer la recherche et contrôler la précision des résultats. Ils ne recherchent cependant pas les k plus proches voisins mais se limitent à la recherche du plus proche voisin. Berrani *et al.* proposent dans [Berrani *et al.*, 2003] une approche hybride de recherche des k plus proches voisins. Leur approche combine à la fois les approximations géométriques des formes englobantes et le contrôle probabiliste de la précision des résultats. Le contrôle de la précision se fait en fonction de la probabilité maximale de ne pas retrouver dans le résultat approximatif un vecteur qui serait présent dans le résultat exact. La méthode de Berrani *et al.* calcule des *clusters* de vecteurs et associe à chacun de ces *clusters* un ensemble de formes englobantes approximatives pour réduire le taux de chevauchement et augmenter le nombre de cellules filtrées durant la recherche. La contrainte de la méthode réside dans le fait que les formes associées aux *clusters* dépendent du niveau d'imprécision fixé *a priori*. Ce niveau d'imprécision est choisi par l'utilisateur et oblige le système à recalculer toutes les formes englobantes chaque fois qu'il est modifié.

1.6 Amélioration de la qualité des résultats de la recherche

Les systèmes de recherche d'images par le contenu n'exploitent pas l'aspect sémantique de l'information dans le processus de recherche puisque les descripteurs de bas niveau ne contiennent pas ces types d'informations. Ces descripteurs peuvent d'ailleurs changer de signification en fonction du contexte d'application. Plusieurs travaux proposent, pour résoudre ce problème, des mécanismes de retour de pertinence (*relevance*

feedback) [Su *et al.*, 2001, Wu et Manjunath, 2001]. Dans un système avec retour de pertinence, l'utilisateur devient acteur du processus de recherche et peut indiquer au système les images qu'il juge intéressantes parmi celles qui lui ont été retournées à une étape antérieure de la recherche. Le système s'inspire ensuite des exemples pertinents ou non indiqués par l'utilisateur pour effectuer une nouvelle recherche et le processus interactif continue jusqu'à ce que l'utilisateur soit satisfait des résultats retournés. Le véritable défi du mécanisme de retour de pertinence est de pouvoir utiliser efficacement les indications de l'utilisateur pour préciser la requête. Les retours utilisateur sont en général utilisés pour modifier l'espace de description des données et/ou la fonction de distance entre les images de façon à mieux capturer les exigences de l'utilisateur. Nous nous intéressons donc particulièrement aux techniques de description des images, et plus particulièrement à la couleur, à la texture, à la forme, et à la combinaison de ces types de descripteurs pour améliorer la qualité de la recherche par le contenu.

1.6.1 Description d'images fixes

La description d'une image a pour but de rassembler tous les éléments nécessaires à la caractérisation de l'image dans un contexte d'utilisation donné. La nature et le type de descripteurs proposés par la communauté du traitement d'images dépendent totalement et précisément du but de la recherche (par exemple un descripteur pour trouver le visage de G. W. Bush). Quatre niveaux de description sont cependant envisageables [Zhang *et al.*, 2001] : le pixel, l'objet, le niveau sémantique, et la connaissance. Au niveau pixel, une image est considérée comme un ensemble de points ou de pixels dont on calcule les caractéristiques de couleur, de texture, et/ou de forme. Il peut s'agir par exemple de la détection d'un cercle dans une image. Le but au niveau de l'objet est de pouvoir identifier dans une image des objets comme un ballon. Au niveau sémantique, l'on s'appuie sur la connaissance du domaine (médical, spatial, sportif, ...) pour extraire des concepts de haut niveau à partir des objets identifiés dans une image. Il est ainsi possible de distinguer les images d'un match de football. Le niveau de la connaissance s'appuie sur le niveau sémantique auquel il est rajouté des informations complémentaires décrivant le contenu. À une image de football par exemple, on peut rajouter des mots décrivant les équipes en compétition, les actions de tir au but, et le score du match, *etc.*

Les traitements du niveau de l'objet cherchent à caractériser et détecter dans une image les objets d'un domaine spécifique. Il importe à ce niveau de préciser la définition de l'objet et les moyens de son extraction. Cette tâche, bien que relativement facile pour un être humain, l'est beaucoup moins automatiquement. Le problème de la reconnaissance d'objets dans une image est identifié à celui de l'étiquetage supervisé basé sur un modèle d'objets appris. Les niveaux supérieurs (sémantique et connaissance) dépendent directement ou non de l'objet et sont d'autant plus difficiles à mettre en œuvre qu'il n'est pas aisé d'établir des liens sémantiques entre les objets. Dans bien des cas, l'annotation manuelle est nécessaire mais malheureusement subjective et non envisageable pour de très grandes bases d'images. Le niveau pixel offre l'avantage d'être entièrement automatique et tend à traduire le contenu.

Notre étude est volontairement limitée à la description visuelle des images qui sont

traitées comme des ensembles de pixels. Dans ce cadre, étant donnée une image i , la description de i consiste à trouver un vecteur $d = f(i)$ qui résume les caractéristiques visuelles de l'image i . Le descripteur d de l'image i est un vecteur de réels ou d'entiers de dimension n et f la fonction de calcul du descripteur. Cette fonction peut être robuste aux occultations partielles, aux variations des conditions de production de l'image et à l'orientation de celle-ci [Smeulders *et al.*, 2000]. Les images au niveau du pixel peuvent être décrites soit par des descripteurs globaux, soit par des descripteurs *ad-hoc* ou par des descripteurs locaux. Ces derniers sont une façon plus fine et plus précise de décrire les images, accroissant ainsi le pouvoir de reconnaissance lors de la recherche par le contenu. En revanche, ils sont coûteux en temps de calcul et demandent beaucoup d'espace de stockage [Amsaleg et Gros, 2001]. Nous limitons notre étude à l'utilisation des descripteurs globaux qui décrivent de façon plus compacte et plus synthétique les images. Contrairement aux descripteurs locaux, les temps de recherche avec les descripteurs globaux sont beaucoup plus réduits. Nous utilisons principalement les descripteurs *MPEG-7* de couleur, de texture, et de forme.

Choix des descripteurs MPEG-7. Le standard *MPEG-7* propose plusieurs types de descripteurs visuels [Manjunath *et al.*, 2002]. Parmi ces types de descripteurs, certains sont de très bas niveau et permettent d'extraire des propriétés de couleur (*Color Layout, Color Structure, Dominant Color, et Scalable Color*), de texture (*Edge Histogram, Homogeneous Texture, et Texture Browsing*), et de forme (*Region-based Shape et Contour-based Shape*). D'autres s'appuient sur les descripteurs de bas niveau ou sur des informations sémantiques additionnelles pour définir des structures de description des propriétés plus complexes. Ainsi, le type de descripteur *Group-of-Frames/Group-of-Pictures* est une agrégation des descripteurs de type *Scalable Color*. Les types de descripteurs composés que propose *MPEG-7* combinent tout simplement des types de base, mais ne changent rien aux caractéristiques des informations extraites. Nous nous intéressons donc uniquement aux descripteurs de bas niveau.

Nous nous sommes appuyés sur les études statistiques menées par Eidenberger [Eidenberger, 2004] pour choisir les types de descripteurs *MPEG-7* utilisés dans cette thèse. Eidenberger étudie principalement les propriétés de redondance et de sensibilité des types de descripteurs par rapport aux images décrites (images en noir et blanc ou en couleur, images synthétiques). *Contour-based Shape* n'a pas été étudié parce que sa structure ne se prête pas aux analyses statistiques. D'après Eidenberger [Eidenberger, 2004], les descripteurs de type *Texture Browsing* sont très sensibles aux informations manquantes de texture. Les descripteurs de type *Color Structure* ont une très forte redondance et sont sensibles aux informations manquantes de couleur. *Dominant Color* est particulièrement sensible aux variations d'intensité lumineuse dans les images. Toutefois, Eidenberger recommande tous les types de descripteurs *MPEG-7* de couleur pour leurs bonnes performances sur les photos (images en couleur). D'après lui, *Edge Histogram* est le meilleur descripteur *MPEG-7* de texture et *Region-based Shape* est bien adapté à tous les types de contenu. Nous avons finalement retenu cinq types de descripteurs *MPEG-7* dont deux de couleur (*Color Layout et Scalable Color*), deux de texture (*Edge Histogram et Homogeneous Texture*), et un de forme (*Region-based Shape*). Le

principe d'extraction de ces types de descripteurs retenus est donné en annexe A.

1.6.2 Recherche sur la couleur

La couleur est la propriété visuelle la plus étudiée des images. Elle correspond à un des éléments fondamentaux de la perception visuelle humaine et ne nécessite aucune transposition intellectuelle pour être interprétée. La propriété de couleur est en général définie par un triplet numérique permettant de coder l'intensité de chacune des composantes du système de couleurs. L'espace RGB (*Red, Green, Blue*) est le système le plus connu et le plus couramment utilisé pour le codage. Il maintient la compatibilité entre les unités qui génèrent ou affichent les images mais présente l'inconvénient de ne pas être uniforme et nécessite la manipulation de plusieurs composantes pour exprimer une seule couleur. Certains systèmes préfèrent donc considérer les couleurs dans un espace basé sur des expériences visuelles d'appréciation de la couleur [Munsell, 1915, Gibson et Nickerson, 1940, Judd et Wyszecki, 1975]. On cite par exemple l'espace HSV (*Hue, Saturation, Value*) où *Hue* représente la teinte ou la couleur dominante perçue (jaune, bleu, *etc.*), *Saturation* indique la pureté de la teinte dominante (pâle, vif, *etc.*), et *Value* donne la luminosité (sombre, clair, *etc.*). Cet espace peut être dérivé de l'espace RGB par les transformations suivantes :

$$\begin{aligned}
 V &= \frac{1}{3}(R + G + B) & S &= 1 - \frac{3}{R+G+B} \min(R, G, B) \\
 H &= 180 \frac{0,5(R-G)+(R-B)}{((R-G)^2+(R-B)(G-B))^{\frac{1}{2}}} & & (1.5) \\
 H_n &\text{'est pas défini si } S = 0 & H &= 360 - H \text{ si } B/V > G/V
 \end{aligned}$$

Après avoir spécifié l'espace de représentation de la couleur, on modélise en général la distribution des couleurs par un histogramme de couleurs. L'histogramme peut représenter trois distributions indépendantes des couleurs dont une pour chaque composante de l'espace ou une seule distribution obtenue par comptage du nombre de pixel par couleur détectée. Cette dernière forme, encore appelée histogramme de couleurs cumulé, est la plus courante. La dimension d'un tel histogramme est proportionnelle au nombre de couleurs détectées dans l'image ou dans la zone d'image décrite. Il n'est pas envisageable de gérer des histogrammes de plusieurs millions de couleurs, tant pour des raisons de calcul des critères de similarité que pour des raisons de stockage. On réduit donc le nombre de couleurs possibles en quantifiant l'espace de couleur.

La recherche d'images sur la couleur peut être classée en trois grands groupes : les approches globales, les approches par partitionnement, et les approches régionales [Stehling *et al.*, 2003]. Les approches globales décrivent simplement la distribution des couleurs d'une image sans tenir compte de la répartition spatiale des couleurs. Elles sont en général les plus efficaces en termes d'extraction et de comparaison des descripteurs, et en terme d'occupation de l'espace. Cependant, l'absence d'information spatiale et topologique est à l'origine de la qualité médiocre des résultats. Les approches régionales par contre utilisent une technique beaucoup plus complexe de description des images

qu'elles décomposent en plusieurs régions de forte similarité. Ces approches donnent des résultats de très bonne qualité mais nécessitent des algorithmes de calcul des descripteurs beaucoup plus coûteux et des fonctions complexes de distance. L'occupation de l'espace se trouve aussi considérablement augmentée. Les approches par partitionnement pourraient se situer entre les approches globales et les approches régionales. Elles décomposent une image suivant un principe qui consiste à superposer une grille rectangulaire sur l'image à décrire.

Les approches globales. La démarche la plus simple et la plus courante de recherche d'images par le contenu consiste à quantifier uniformément l'espace RGB et à représenter la distribution des couleurs par un histogramme global que l'on transcrit dans un espace vectoriel multidimensionnel. Les images sont ensuite accédées *via* les index multidimensionnels (voir section 1.4) et généralement comparées avec une distance L_1 . Le mode de description des images dans cette démarche est très peu coûteux en temps de calcul et les propriétés d'invariance à la rotation et à la translation sont respectées. Par contre, il peut arriver que des images ayant des dispositions spatiales très différentes de couleur aient exactement la même description. D'autres part, l'interférence de couleurs peut entraîner des corrélations très fortes entre les coordonnées des vecteurs dans l'espace multidimensionnel. Ainsi, deux images de couleurs très voisines mais non identiques peuvent se retrouver très éloignées l'une de l'autre en terme de distance.

Un moyen de résoudre le problème d'interférence de couleurs est d'exploiter les corrélations entre les composantes des vecteurs en utilisant une distance euclidienne pondérée [Bimbo, 1999, Lu, 1999]. Malheureusement, lorsque ce type de distance est utilisée pour comparer les histogrammes, l'usage des index multidimensionnels n'est plus possible puisqu'ils ne gèrent pas les corrélations entre composantes de vecteurs. Une alternative est d'utiliser les méthodes d'accès métriques [Chavez *et al.*, 2001] mais ces méthodes posent le problème de passage à l'échelle. Stricker et Orengo proposent deux manières de modifier la représentation des images pour régler le problème d'interférence de couleurs [Stricker et Orengo, 1995]. La première est d'utiliser un histogramme de couleurs cumulé qui exploite le fait que la similarité entre deux barres rapprochées d'un histogramme doit être plus grande que celle des barres plus éloignées. Cette idée est reprise et améliorée dans [Zhang *et al.*, 1998]. La seconde approche de Stricker et Orengo se limite uniquement à la représentation des couleurs dominantes à partir des trois premiers moments de chaque canal de couleur. Cette approche de représentation aboutit à une occupation de l'espace beaucoup plus réduite que celle des histogrammes de couleurs. De plus, l'utilisation des méthodes statistiques dans la description évite de quantifier l'espace de représentation des couleurs. En revanche, des couleurs complètement différentes peuvent avoir les mêmes statistiques.

De nombreuses études ont été faites sur l'amélioration de la représentation des descripteurs de couleurs. Certaines proposent de remplacer l'espace de description RGB par un espace uniforme et plus proche de la perception humaine, ce qui éviterait des problèmes de quantification. D'autres proposent soit un encodage spécial des histogrammes de couleurs pour obtenir une forme beaucoup plus compacte des descripteurs, soit une structure plus élaborée que les histogrammes pour tenir compte des relations spatiales

entre les pixels d'une image.

Les approches par partitionnement. La recherche d'images sur la couleur décrite par partitionnement consiste à superposer une grille rectangulaire fixe (3×3 , 4×4 , 8×8 , *etc.*) sur l'image à décrire de telle manière que toutes les cellules aient la même taille et ne se chevauchent pas. Chaque cellule est ensuite décrite par un histogramme local dans un espace uniformément quantifié. La distance entre deux images est estimée par la moyenne des distances entre les histogrammes locaux des cellules équivalentes. L'approche par partitionnement, tout comme l'approche globale, dépend de la quantification de l'espace sous-jacent et des mesures adoptées pour la similarité. Si elle tient compte de la distribution spatiale des couleurs dans les images, elle introduit toutefois le problème de la sensibilité à la rotation et à la translation. Ainsi, deux images semblables mais dont la position des objets diffère peuvent être très éloignées l'une de l'autre puisque le calcul de la distance ne tient pas compte des cellules de positions différentes. On retrouve un problème analogue à celui de l'interférence de couleurs dans les approches globales. On parle dans ce cas d'interférence de cellules [Stehling *et al.*, 2003]. Le problème est abordé de deux façons : soit on définit une distance beaucoup plus élaborée entre les cellules, soit on définit une nouvelle structure de représentation des cellules.

La définition d'une distance plus complexe et plus élaborée entre deux images partitionnées I_1 et I_2 ne se limite pas à la comparaison des cellules équivalentes mais identifie pour chaque cellule de I_1 les cellules les plus proches dans I_2 . Wang propose de comparer les cellules deux par deux, chacune appartenant à une image différente, et de pondérer les distances par l'écart spatial entre les cellules [Wang, 2001]. Il modélise pour cela chaque cellule par un nœud d'un graphe biparti dans lequel le coût d'un arc est la distance entre les couleurs des cellules reliées par l'arc. L'ensemble des couples de cellules qui minimisent la fonction de distance est la solution du problème d'affectation [Kuhn, 1955] et ce problème est très complexe (complexité en $O(n^3)$ où n est le nombre de cellules d'une image).

Le problème d'interférence de cellules est habituellement contourné par la définition d'une structure de représentation hiérarchique des cellules en arbre quaternaire [Shusterman et Feder, 1994]. Au sommet de la hiérarchie le contenu de l'image est représenté de façon globale sans contraintes spatiales. Au second niveau, la représentation de l'image est décomposée en cellules de taille 2×2 . Au troisième niveau, la décomposition se fait en cellules de taille 4×4 et ainsi de suite. Les cellules de niveaux différents ont des tailles différentes mais se chevauchent, ce qui réduit les problèmes liés aux contraintes spatiales. La comparaison entre deux images commence au sommet de la hiérarchie, puis elle est progressivement affinée aux niveaux suivants. L'inconvénient d'une telle représentation hiérarchique est qu'elle est gourmande en ressources de stockage.

Les approches régionales. Les approches régionales de recherche d'images par le contenu utilisent les techniques de segmentation pour décomposer les images en fonction de leur contenu visuel. La segmentation est plus flexible et plus robuste que la

grille rectangulaire fixe et imposée des approches par partitionnement. Les techniques de segmentation peuvent cependant donner des régions incohérentes qui ne correspondent pas du tout à ce à quoi on s'attendrait dans le monde réel. Plusieurs systèmes mettent en œuvre l'approche régionale de recherche d'images. Le système QBIC d'IBM [Ashley *et al.*, 1995] s'appuie sur le processus de *clustering* dans lequel deux *clusters* sont fusionnés si leur rang mutuel est inférieur à un seuil fixé. Le rang mutuel des *clusters* C_1 et C_2 est $n + m$ si C_2 est le $n^{\text{ième}}$ *cluster* le plus proche de C_1 et si C_1 est le $m^{\text{ième}}$ *cluster* le plus proche de C_2 . Chaque *cluster* de couleur est représenté par un arbre et la distance entre deux *clusters* est la distance euclidienne entre la moyenne de leurs descripteurs de couleur. La distance entre deux régions tient donc compte de la distance entre les couleurs des régions et de celle entre les arbres associés. Enfin, la distance entre deux images I_1 et I_2 est la moyenne des distances entre chaque région de I_1 et ses régions voisines dans I_2 .

Le système *Netra II* utilise un algorithme de détection de frontière appelé *EdgeFlow* pour segmenter les images [Deng et Manjunath, 1999]. Une quantification perceptuelle de la couleur est ensuite effectuée sur chaque région puis un histogramme est calculé pour chaque couleur obtenue après l'étape de la quantification. Les histogrammes de couleur de deux régions sont comparés par une distance équivalente à la distance euclidienne pondérée [Bimbo, 1999]. Chaque couleur quantifiée d'une région est indexée séparément dans un espace à trois dimensions, puis son pourcentage et l'étiquette de la région sont stockés. Le système *Netra II* n'utilise pas les techniques traditionnelles d'indexation multidimensionnelle mais définit une structure qui lui est propre. En définitive, les systèmes qui implémentent les approches régionales de recherche d'images par le contenu diffèrent sur la stratégie mise en place pour la segmentation des images en région ainsi que sur la définition de la fonction de comparaison des images segmentées.

1.6.3 Recherche sur la texture

La texture permet de distinguer des zones de couleurs similaires mais de signification complètement différentes. On peut ainsi, grâce à la texture, faire la différence entre le coucher du soleil et une orange. Les textures traduisent donc l'aspect homogène d'une zone et peuvent être décrites selon leurs propriétés spatiales et fréquentielles. L'approche basée sur la configuration spatiale de l'image consiste à représenter la texture sous la forme d'un histogramme en niveau de gris. Des moments statistiques sont alors calculés sur les matrices de co-occurrence pour donner des indications sur le contraste, la directionnalité et la périodicité du motif [Haralick *et al.*, 1973, Haddon et Boyce, 1993]. Les matrices de co-occurrence définissent la probabilité jointe de l'occurrence de deux pixels quelconques dans une image en niveau de gris. La relation spatiale entre ces deux pixels est exprimée par une distance et un angle. Le calcul des matrices de co-occurrence peut s'avérer très coûteux sur des images de haute résolution.

La méthode de description la plus utilisée de la texture concerne les propriétés fréquentielles et s'appuie sur la transformée de Fourier, la représentation de Gabor, les ondelettes et la transformée en cosinus discrète [Manjunath et Ma, 1996, Nastar *et al.*, 1998]. Elle repose sur l'analyse d'une fonction de densité spectrale dans un domaine fréquentiel

[Bajscy, 1973]. Les coefficients d'une transformation bi-dimensionnelle indiquent la corrélation des motifs dans une image. La granularité de la texture étant proportionnelle à la période spatiale, les textures de grande granularité ont une énergie spectrale concentrée aux fréquences basses alors que les textures plus fines concentrent leur énergie aux fréquences hautes. Dans tous les cas, à l'issue des transformations, on obtient un vecteur de valeurs numériques que l'on va comparer avec une distance pondérée.

Une approche de recherche d'images par l'exemple sur la texture est de considérer les mêmes mesures de texture que celles employées au moment de la classification ou de la reconnaissance de motif et de comparer ces mesures à celles extraites de l'exemple. La qualité de l'approche est alors dépendante de la représentation adoptée. Une autre approche, plus dépendante de l'utilisateur, utilise des mécanismes de retour de pertinence (*relevance feedback*) pour faire apprendre au système la façon dont l'utilisateur perçoit la similarité de texture.

Utilisation des mesures de texture. L'idée consiste à définir et à calculer des descripteurs de texture sur les images. Une mesure de similarité permet ensuite de comparer les images entre elles pendant la recherche. Les systèmes qui exploitent cette idée diffèrent essentiellement par la méthode de description de la texture des images. Le système CANDID (*Comparison Algorithm for Navigating Digital Image Databases*) fait de la recherche sur des images décrites par des signatures de texture estimées avec une somme de fonctions gaussiennes [Kelly *et al.*, 1995]. La similarité entre deux signatures de texture est estimée soit par une distance euclidienne normalisée, soit par le produit intérieur des deux signatures.

Dans le système *NeTra*, Manjunath et Ma [Manjunath et Ma, 1996] décrivent la texture d'une image par la moyenne et l'écart-type des coefficients de Gabor appliqués à toute l'image. Au cours de la recherche, les distances entre les vecteurs décrivant la texture et le vecteur requête sont calculées puis triées par ordre croissant. Seules les images décrites par les vecteurs les plus proches du vecteur requête sont retournées. Manjunath et Ma proposent de choisir les filtres de Gabor en fonction du type de modèle à extraire. Ils calculent pour cette raison la différence entre le spectre de la requête et le spectre moyen de toutes les images de la base et choisissent ensuite le filtre le plus approprié. Cette stratégie permet d'accélérer la recherche.

Utilisation des attributs perceptuels. L'approche consiste à tenir explicitement compte d'une part de la perception humaine en choisissant des descripteurs appropriés et, d'autre part, de l'utilisateur par des mécanismes de retour de pertinence (*relevance feedback*). Le système QBIC décrit la texture des images suivant les trois principales composantes de granularité, de contraste, et de directionnalité [Niblack *et al.*, 1993]. Les descripteurs de ces trois composantes sont extraites des images de couleur préalablement converties en niveaux de gris. La comparaison des images est faite par l'évaluation des la distance euclidienne pondérée dans l'espace tridimensionnel de description. Les poids sont estimés par l'inverse de la variance de chaque composante de texture.

Le système *Photobook* est un ensemble d'outils interactifs pour la navigation et la recherche par le contenu des images ou des séquences d'images [Pentland *et al.*, 1996].

Dans ce système, la base d'images est organisée en plusieurs groupes en fonction des objets qu'elle contient. On distingue ainsi le groupe des visages, celui des formes et celui des textures. Chaque groupe peut éventuellement avoir des sous-groupes et utilise une méthode d'indexation et de recherche qui lui est spécifique. Dans le groupe des textures la recherche se fait en trois étapes dont l'objectif est de détecter dans l'ordre : la présence des structures périodiques, l'orientation de texture, et une composante aléatoire si les motifs ne sont pas très structurés. Dans le cas général, les deux premiers points sont suffisants pour évaluer la similarité des images.

1.6.4 Recherche sur la forme

La forme peut être décrite par des vecteurs, par des relations ou à travers des transformations. Le choix d'une forme particulière de représentation est guidé par les besoins réels d'utilisation tels que les caractéristiques de la forme analysée, la possibilité d'indexation, la robustesse aux déformations, *etc.* La description de la forme à travers des transformations consiste à mesurer par une distance (distance transformationnelle) l'effort qu'il faut pour passer d'une forme à une autre. Cette mesure se fait uniquement au cours de la recherche et une indexation n'est pas envisageable pour une telle approche. Dans les approches relationnelles, une forme est décomposée en plusieurs parties pour lesquelles des descripteurs appropriés sont calculés. La description globale de la forme inclut ensuite les descriptions de chacune des parties ainsi que les relations entre les différentes parties. Cette approche de description n'est pas très courante dans la recherche par le contenu, mais en revanche, elle est très utilisée pour la reconnaissance des formes complexes.

Les techniques de modélisation de la forme par des vecteurs sont les plus couramment utilisées dans la recherche par le contenu et se prêtent bien à l'indexation. Elles représentent une forme par un vecteur numérique et la différence entre deux formes est évaluée par une distance. On distingue deux techniques de modélisation vectorielle de la forme : celles basées sur les contours et celle basées sur les régions.

Les premières décrivent la région considérée uniquement à travers sa frontière, c'est-à-dire au moyen des pixels situés sur le contour de la région. Elles peuvent utiliser les propriétés locales ou globales de la frontière des objets qu'elles décrivent. Les descriptions basées sur les propriétés locales sont les plus robustes aux irrégularités de forme et les plus adaptées à la recherche des formes partiellement occultées ou possédant des contours qui se chevauchent.

Les secondes approches de modélisation de la forme par des vecteurs représentent une région grâce à ses propriétés internes c'est-à-dire à travers tous les pixels de la région considérée. Elles utilisent soit des attributs géométriques simples de la région à décrire, soit des moments, soit les premiers coefficients des transformations telles que la transformation $2D$ de Fourier ou les ondelettes. La description des propriétés géométriques d'une région peut être obtenue en mesurant les propriétés des points de la région telles que : le nombre de pixels de la région, l'ellipse ou le rectangle englobant minimal, ou la compacité de la région. Avec les seuls critères géométriques toutefois, une très simple variation de forme, même perceptuellement insignifiante peut entraîner

ner de très grosses différences de représentation. Les techniques les plus intéressantes sont la théorie de Fourier et celle des moments, chacune étant précise et invariante aux transformations géométriques d'intérêt [Rui *et al.*, 1996, Jähne, 1997]. Ces techniques donnent des résultats similaires pour caractériser la propriété forme dans les images.

La recherche d'images sur la forme dépend de la manière dont celle-ci est représentée selon l'approche globale. Elle peut se faire soit sur des vecteurs de descripteurs, soit en considérant un ensemble de transformations allant d'une forme initiale soumise comme requête à la forme désirée.

Vecteurs de descripteurs. Les systèmes tels que QBIC [Niblack *et al.*, 1993] qui utilisent des descripteurs considèrent les formes comme des vecteurs et la mesure de similarité comme une distance dans l'espace multidimensionnel des descripteurs. Cette méthode de recherche est particulièrement adaptée aux bases d'images de taille importante. Les vecteurs de forme sont alors organisés au sein d'une structure d'index qui permet de filtrer les accès aux images. Il est bien entendu utile de noter que l'on n'échappe pas à tous les problèmes posés au cours d'une recherche lorsque la dimension des vecteurs est très grande. Grosky *et al.* utilisent une courbe polygonale pour approximer les formes [Grosky *et al.*, 1992]. Ils associent à chaque sommet s de la courbe un descripteur local de forme défini par : (A_s, D_s, X_s, Y_s) où A_s est l'angle interne au sommet s , D_s la distance entre le sommet s et le sommet adjacent pris dans le sens des aiguilles d'une montre, et (X_s, Y_s) les coordonnées du sommet s . Une forme est alors représentée comme une chaîne de caractères et la distance entre deux formes f_1 et f_2 correspond au nombre de modifications à effectuer pour passer d'une chaîne de caractères représentant la forme f_1 à la chaîne représentant f_2 . Grosky *et al.* proposent une description fine qui permet de faire des requêtes sur des formes partielles. La qualité des résultats dépend de la longueur des chaînes de description. Le système QBIC représente les formes de façon plus globale par des images binaires en utilisant une série de moments algébriques [Niblack *et al.*, 1993]. Les descripteurs obtenus subissent ensuite une transformée de Karhunen-Loeve (KLT) pour être indexés par un R*-tree dans un espace de dimension réduite. La similarité entre les formes est estimée par une distance euclidienne pondérée. La qualité de la recherche dépend très fortement de la requête soumise. La méthode est particulièrement adaptée aux formes simples et ne permet pas de décrire avec précision les formes complexes.

Séries de transformations. Des études ont montré des incohérences entre la description vectorielle et les distances métriques pour les similarités de forme. La proximité des descripteurs dans l'espace ne correspond pas toujours à la similarité perçue par l'œil humain. Pour contourner ce problème, la vision par ordinateur utilise des distances transformationnelles pour la reconnaissance des formes [Youille *et al.*, 1991]. Cette approche modifie une forme en une autre par un nombre fini de déformations. Elle mesure ensuite la similarité par le taux de déformation nécessaire pour faire coïncider les deux formes. Les déformations sont évaluées soit par des modèles élastiques, soit par des modèles évolutionnaires. Les modèles élastiques utilisent un ensemble discret de paramètres ou un contour continu initial et arbitraire qui devra subir une déformation

continue pour obtenir la forme désirée. Les modèles évolutionnaires par contre supposent que les formes résultent d'un processus au cours duquel des forces sont appliquées à des points spécifiques du contour et à chaque étape de la déformation. L'indexation avec les modèles élastiques est plus difficile qu'avec les modèles évolutionnaires.

1.6.5 Combinaison des types de descripteurs globaux

Les descripteurs de couleur, de texture, ou de forme peuvent être combinés pour caractériser de façon encore plus efficace le contenu. Dans certaines situations, la couleur, la texture, ou la forme seule ne suffit pas pour décrire une région d'intérêt dans une image. Si, par exemple, un utilisateur recherche une image avec un ciel bleu, la couleur seule peut ne pas réduire efficacement l'ensemble des résultats possibles. Il pourrait avoir dans la base d'images à parcourir un très grand nombre d'images avec des régions bleues non intéressantes pour l'utilisateur. Ce dernier peut en revanche préciser sa requête en mentionnant par exemple que la région est homogène et lisse du point de vue textural. La précision supplémentaire aura pour effet de réduire et de cibler davantage les images susceptibles d'être retournées.

Une approche de combinaison des types de descripteurs globaux pour la recherche d'images par le contenu consiste à calculer séparément les mesures de similarités pour chaque type de descripteur puis à dériver une mesure composée de la similarité globale entre les images. Les systèmes CANDID [Kelly *et al.*, 1995], QBIC [Niblack *et al.*, 1993], et NeTra [Manjunath et Ma, 1996] offrent la possibilité de combiner facilement les types de descripteurs de texture et de forme. Les types de descripteurs de couleur peuvent aussi rentrer dans la combinaison.

1.7 Synthèse

Après avoir défini le concept de la recherche d'images par le contenu, nous avons présenté dans ce chapitre les éléments les plus courants qui rentrent en jeu dans l'évaluation des systèmes de recherche d'images fixes. Nous avons ensuite abordé l'optimisation des performances de ces systèmes sous deux angles : celui de l'organisation (ou de l'indexation) des images qui influence le temps de recherche et celui de la description des images qui a un impact sur la qualité de la recherche par le contenu.

Les techniques d'indexation multidimensionnelle présentées dans ce chapitre montrent leurs limites dans des espaces de grande dimension où le phénomène de la malédiction de la dimension apparaît. Il en résulte donc une recherche dégradée et beaucoup plus lente qu'une simple recherche séquentielle exhaustive. Pour contourner ces problèmes, les travaux également présentés dans ce chapitre proposent soit un codage particulier des données pour faciliter la recherche, soit de recourir aux techniques approximatives qui exploitent les propriétés des structures d'index pour aller plus vite.

Les techniques de description du contenu quant à elles influencent la qualité des résultats de la recherche. Les travaux qui tentent d'améliorer cette qualité jouent principalement sur les stratégies de description du contenu. Ainsi, le facteur le plus important pour une recherche est le choix de la mesure de similarité. On observe qu'en

général, une mesure simple à évaluer donne des temps de recherche réduits mais ne tient pas compte de la complexité de la description du contenu et il en résulte une qualité médiocre de recherche. En revanche, une mesure de similarité beaucoup plus complexe aboutit à de très fortes similarités visuelles mais les temps de calculs peuvent être plus longs. Ainsi, les techniques d'amélioration de la qualité de la recherche établissent en réalité des compromis entre les descripteurs qu'ils proposent et le temps mis pour les comparer.

La tendance actuelle est que la fouille de données semble une voie intéressante pour améliorer les systèmes de recherche par le contenu. Nous pensons que les techniques de la fouille de données peuvent être un moyen de contourner la faiblesse des techniques d'indexation physique plus ou moins adaptées au caractère multidimensionnel des descripteurs visuels. Nous montrerons que ces techniques, dont la description est un des objets du chapitre suivant, peuvent faciliter l'accès aux images en réduisant l'espace de recherche.

Chapitre 2

Fouille de données pour l'organisation des images fixes

L'extraction de connaissances à partir d'images apporte une complexité nouvelle liée à la représentation des données analysées. Après avoir décrit les principaux concepts de la fouille de données, nous en étudions les applications à l'indexation et à la recherche par similarité dans de grandes bases d'images fixes, puis nous nous focalisons sur les techniques de clustering et de recherche de règles d'association. En particulier, nous faisons le point sur les principales approches de calcul des clusters puis nous montrons comment la découverte de règles d'association sur des clusters de descripteurs d'images peut être exploitée pour le traitement d'une requête par l'exemple.

2.1 Introduction

La fouille de données consiste à appliquer aux données ciblées et préalablement préparées, des techniques qui fournissent un ensemble de connaissances non triviales sur les données analysées [Fayyad *et al.*, 1996b]. La fouille d'images est la découverte de connaissances sous forme de classes, de groupes (*clusters*), de règles ou de modèles à partir d'un grand ensemble de descriptions d'images. Son processus est analogue à celui de la fouille classique de données et peut se résumer en quatre étapes : la sélection et la constitution de la collection d'images à analyser, la description des images, la fouille proprement dite qui consiste à extraire des modèles, la présentation et la validation des modèles extraits.

La recherche d'images à partir du contenu est un procédé par lequel une liste d'images de descriptions les plus similaires à celles d'une image-requête est retournée à partir d'une grande collection d'images. La procédure de recherche opère donc suivant des contraintes liées à la spécification des requêtes. Elle n'a donc rien à voir avec celle de la fouille d'images. Cependant, l'extraction des descripteurs pour l'organisation des images et pour la recherche par le contenu peut exploiter des techniques comme celles de la fouille d'images.

La suite de ce chapitre décrit les principaux concepts de la fouille de données et

présente leurs applications à l'indexation et à la recherche par similarité dans de grandes bases d'images fixes. Nous faisons ensuite le point sur les principales approches de calcul des *clusters* puis nous montrons comment les règles d'association peuvent être découvertes à partir des *clusters* de descripteurs pour le traitement des requêtes par similarité.

2.2 Tâches de la fouille de données

La nature de la connaissance extraite à partir des données dépend de la finalité du processus d'extraction, c'est-à-dire de la tâche abordée. Nous présentons dans cette section les quatre principales tâches de la fouille de données et nous décrivons les travaux relatifs à leur application à l'organisation des images fixes.

2.2.1 La classification supervisée

La classification supervisée consiste à regrouper divers objets (individus) en sous-ensembles d'objets ou classes selon un modèle de classement. Les classes sont connues *a priori* et ont une sémantique associée bien définie. Le modèle de classement est élaboré à partir d'un échantillon représentatif dans lequel chaque objet (ou encore exemple) va être associé à un nom de classe. Chaque classe est ainsi décrite par un modèle utilisé ensuite pour classifier les objets de la base de données. On distingue plusieurs modèles de classement, entre autre, arbres de décision [Breiman *et al.*, 1984, Quinlan, 1986, Quinlan, 1993], plus proches voisins [Akkus et Güvenir, 1996, Denœux, 1995], réseaux de neurones [Lippmann, 1987, Lu *et al.*, 1995], treillis de concepts [Oosthuizen, 1991, Kourie et Oosthuizen, 1998], et les applications du classement sont nombreuses. En géographie par exemple, les applications peuvent être l'annotation automatique des images satellites de façon à distinguer sur une carte, la mer, le désert, ou une agglomération. En imagerie médicale, il pourra s'agir d'annoter les tissus (normal *vs.* cancéreux) à partir des images mammographiques.

Le principal but visé par le classement d'images est de pouvoir obtenir une organisation « sémantique » d'une base d'images pour faciliter la navigation ainsi que la recherche dans la base. Il est pourtant difficile d'atteindre cet objectif puisque les images réelles partageant la même « sémantique » peuvent n'avoir aucune similarité visuelle. À l'inverse, des images présentant une très forte similarité visuelle peuvent n'avoir aucune « sémantique » commune. Par « sémantique » d'une image, nous voulons parler d'une interprétation que celle-ci peut avoir pour un utilisateur dans une situation donnée. À titre d'exemple, les images de la plage et celles du ski en montagne peuvent être regroupées dans la rubrique « photos de vacances » alors qu'elles sont visuellement très différentes. Par contre, les images d'un désert au coucher du soleil n'ont rien à voir avec celles d'une nappe de table de couleur orangée, pourtant la similarité visuelle est importante.

L'un des premiers articles sur le classement d'images s'intéresse aux images satellites [Fayyad *et al.*, 1996a]. Il décrit le système SKICAT qui combine le traitement d'images, la gestion efficace des bases de données et les techniques d'apprentissage pour traiter

trois téraoctets d'images. SKICAT utilise un algorithme d'induction d'arbres de décision pour apprendre l'ensemble des objets classés par les astronomes. Huang *et al.* [Huang *et al.*, 1998] proposent un schéma de classement hiérarchique des images d'une bibliothèque numérique destiné à faciliter la navigation, la recherche et la présentation conviviale de la bibliothèque. La description des images utilise une technique simple et efficace basée sur les corrélogrammes de couleurs. Le modèle appris est présenté sous forme d'un arbre de classification. La méthode considère que l'ensemble d'apprentissage est connu à l'avance et que toutes les classes sont correctement labellisées. Cette hypothèse constitue une limite de la méthode surtout pour de grands ensembles d'images. Une bibliothèque numérique peut contenir plusieurs millions d'images et il est difficile de constituer des classes et des labels corrects.

Wang et Li [Wang et Li, 2002] proposent une technique d'indexation d'images par des termes linguistiques. Le principe repose sur la construction d'un modèle d'apprentissage statistique (modèle de Markov caché). Les auteurs définissent une liste de concepts à apprendre et à inclure dans un dictionnaire (dictionnaire des concepts). Ils préparent pour chaque concept à apprendre un ensemble d'apprentissage constitué d'images non nécessairement similaires sur le plan visuel mais supposées décrire le concept. Des modèles de Markov cachés sont ensuite construits à partir des ensembles d'apprentissage constitués et tiennent compte de la description des images à différentes résolutions. Une description textuelle est associée à chaque modèle construit puis l'ensemble est stocké dans le dictionnaire avec le concept correspondant. L'indexation des images se fait par comparaison des modèles statistiques construits. Une image est associée au concept dont le modèle est le plus proche de l'image selon une mesure de similarité statistique. L'image et les termes du concept sont stockés dans une base de données pour des traitements ultérieurs de requêtes par mots-clés. L'avantage de l'approche de Wang et Li [Wang et Li, 2002] réside dans la mise à jour des concepts. Dans ce cas, seul le concept mis à jour est réappris et non l'ensemble de tous les concepts de la base. Toutefois, la méthode n'offre qu'un nombre très limité de concepts puisque l'ensemble des concepts est constitué manuellement.

2.2.2 La classification non supervisée (*clustering*)

Le but de la classification non supervisée¹ est de partitionner un ensemble hétérogène d'objets en sous-ensembles beaucoup plus homogènes (c'est-à-dire dont les éléments ont des caractéristiques semblables) appelés classes ou *clusters*. Contrairement au classement, les *clusters* ne sont pas connus d'avance mais ils doivent contenir des objets très similaires (minimisation de la dissimilarité *intra-cluster*) et être suffisamment éloignés les uns des autres (maximisation de la dissimilarité *inter-cluster*). La similarité des objets est en général mesurée en termes de distance entre les objets. La classification non supervisée peut être un préalable à la résolution de certains problèmes complexes. Il peut ainsi s'avérer utile de commencer par segmenter la population en espérant que le problème soit plus facile à résoudre sur les groupes ainsi constitués. Ces groupes seront donc considérés homogènes suivant un critère donné. Un analyste peut, par exemple, classer

¹Le terme *segmentation* est aussi employé.

les caractéristiques d'une image de télédétection en utilisant des éléments de l'interprétation visuelle tels que la couleur pour identifier automatiquement des groupes homogènes de pixels qui représentent des classes. Il associera ensuite à un groupe de pixels d'une couleur donnée une classe particulière en fonction de sa connaissance du domaine : le bleu sera par exemple associé à l'eau et le vert à la végétation. Le *clustering* a été largement étudié en statistiques [Carpineto et Romano, 1993, Cheeseman et Stutz, 1996], en apprentissage numérique [Fisher, 1987] et en base de données [Zhang *et al.*, 1996].

Les méthodes de *clustering* s'appliquent à de nombreuses situations. Dans l'organisation des images par le contenu, les images sont décrites par des vecteurs dans un espace multidimensionnel puis regroupées par des algorithmes de *clustering* en fonction de leur similarité visuelle. Cette approche facilite la recherche par le contenu. Les algorithmes de *clustering* s'appliquent aussi sur des images considérées comme des ensembles de pixels. C'est, par exemple, le cas chez les spécialistes (environnementalistes, écologistes, *etc.*) qui s'intéressent à l'évaluation et à l'interprétation de toutes les données relatives aux sols et à l'environnement. Dans ce cadre, ils reçoivent régulièrement des milliers d'images de très grande résolution (généralement 512×1024 pixels) à partir desquelles ils essaient d'abord de séparer les arbres de l'arrière-plan. Ils constituent ensuite des groupes d'arbres pour des analyses statistiques. L'algorithme BIRCH (*Balanced Iterative Reducing and Clustering*) proposé dans [Zhang *et al.*, 1996] a été utilisé pour réaliser un tel travail. Il est conçu pour partitionner des ensembles de données de taille très importante et qui ne tiennent pas toujours en mémoire centrale. La décision de classer par l'algorithme BIRCH est faite sans avoir à parcourir tous les points ou toutes les classes disponibles. Le principe de classification repose sur l'hypothèse que tous les points ne sont pas uniformément répartis dans l'espace et n'ont pas la même importance. BIRCH repose sur les concepts de CF (*Clustering Feature*) et d'arbre CF (*CF-Tree*). CF est un descripteur de *cluster* qui résume l'information des points du *cluster* sous forme d'un triplet : $CF = (n, LS, SS)$ où n est le nombre de points du *cluster* C , LS est la somme linéaire $\sum_{i=1}^n x_i$ et SS est la somme quadratique $\sum_{i=1}^n x_i^2$; x_i , $i = 1..n$ étant un point quelconque du *cluster* C . Le calcul des *clusters* se fait en plusieurs étapes. La première et la plus importante construit une hiérarchie appelée arbre CF (*Clustering Feature Tree*) dont les nœuds et les feuilles contiennent des CF. La construction de l'arbre CF considère les points de la base de données un à un et utilise les paramètres de contrôle suivants :

- le nombre B maximum de cases d'un nœud intermédiaire qui représente un *cluster* constitué de *sous-clusters* correspondant à ces cases ;
- le nombre L maximum de cases (*sous-clusters*) d'un nœud terminal.

Les autres étapes de BIRCH permettent tout simplement d'affiner la construction de la première étape. Dans certains cas, l'utilisation d'un algorithme hiérarchique traditionnel peut s'avérer nécessaire pour partitionner les centres de gravité des *sous-clusters*. La constitution des *sous-clusters* des nœuds terminaux doit satisfaire un critère de seuil : le diamètre (ou le rayon) du *sous-cluster* ne doit pas dépasser un certain seuil T qui détermine la taille de l'arbre CF à construire. BIRCH construit des *clusters* convexes et sphériques et tient compte des données isolées (*outliers*). Cependant, les *clusters* construits sont très sensibles aux paramètres de contrôle B et T .

2.2.3 La découverte des règles d'association

Découvrir des règles d'association c'est générer des relations informatives à partir des données. Une formalisation de la technique est introduite pour la première fois dans [Agrawal *et al.*, 1993]. Soit $I = \{i_1, i_2, \dots, i_N\}$ un ensemble de N éléments distincts appelés items, objets ou attributs, et D un ensemble de transactions ayant comme attributs les éléments de I . Une règle d'association est une implication de la forme $A \rightarrow B$, où $A, B \subset I$, et $A \cap B = \phi$. A est appelé *antécédent* et B *conséquent* de la règle. Un ensemble d'objets ou d'*items* est appelé *itemset*. A chaque *itemset*, on associe une mesure statistique appelée *support* et notée *supp*. Pour un *itemset* $A \subset I$, $supp(A) = s$, si la fraction de transactions de D contenant A est égale à s . Le support d'une règle d'association $A \rightarrow B$ est défini comme suit :

$$supp(A \rightarrow B) = supp(A \cup B)$$

Une mesure de confiance notée *conf* permet de caractériser la pertinence d'une règle. Pour $A \rightarrow B$, elle est définie par :

$$conf(A \rightarrow B) = \frac{supp(A \cup B)}{supp(A)}$$

La découverte de règles d'association peut conduire, dans de très grandes bases de données, à l'extraction de millions de règles. Toutes ces règles ne sont pas forcément utiles et l'on est amené à se restreindre à celles dont le support et la confiance sont supérieurs ou égaux à deux seuils respectivement *minsupp* et *minconf* fixés par l'utilisateur. La découverte des règles s'articule autour de deux principaux points : **le calcul d'itemsets fréquents** et **la génération des règles**. Le premier point consiste à produire tous les itemsets dont le support est supérieur ou égal au seuil *minsupp*. C'est l'étape la plus importante et la plus coûteuse du processus de détermination des règles. Les premiers algorithmes apparus dans la littérature génèrent un ensemble d'itemsets candidats qu'ils élaguent ensuite. On peut par exemple citer *Apriori* [Agrawal *et al.*, 1993] et ses dérivés : *Apriori Tid*, *Apriori Hybrid* [Agrawal et Srikant, 1994], *Close* [Pasquier *et al.*, 1999], *Pascal* [Bastide *et al.*, 2002]. Ces variantes optimisent l'algorithme de référence *Apriori* en réduisant soit l'ensemble d'*itemsets* candidats soit la base de données à parcourir.

De nombreux algorithmes d'extraction des règles d'association sont ainsi proposés et optimisés pour les bases de données transactionnelles dont la structure est bien définie [Pasquier *et al.*, 1999, Bastide *et al.*, 2002]. Mais, leur application aux images nécessite des adaptations. Ordonez et Omiecinski transforment les images et leur appliquent des techniques classiques d'extraction de règles d'association [Ordonez et Omiecinski, 1999]. Ils calculent pour chaque image des descripteurs et identifient des objets simples tels que des rectangles ou des triangles. Ils créent ensuite une base auxiliaire dans laquelle une image est représentée par une paire composée de la transaction et des différentes régions la constituant identifiées à des objets. L'avantage de cette approche est qu'aucune connaissance *a priori* n'est requise. Par contre, la méthode s'adapte très mal aux images complexes et l'étape la plus délicate dans la démarche est celle de l'identification des objets qui conditionne la validité et l'interprétation des règles extraites.

Une limitation de la définition classique des règles d'association est l'absence de prise en compte de la répétition d'objets pouvant sous certaines conditions être porteuse d'information. Un grand nombre de régions de couleur bleue dans une image donne par exemple une idée de l'intensité de la couleur bleue. Zaïane *et al.* [Zaïane *et al.*, 2000] proposent un nouveau formalisme de règles d'association avec objets récurrents comme une implication de la forme :

$$\alpha_1 P_1 \wedge \alpha_2 P_2 \wedge \dots \wedge \alpha_n P_n \rightarrow \beta_1 Q_1 \wedge \beta_2 Q_2 \wedge \dots \wedge \beta_m Q_m \quad (2.1)$$

où P_i (avec $i \in [1..n]$) et Q_j (avec $j \in [1..m]$) sont des descripteurs d'images, et α_i, β_j sont des entiers indiquant le nombre d'occurrences des descripteurs. $\alpha_i P_i$ (resp. $\beta_j Q_j$) est vrai si P_i (resp. Q_j) a α_i (resp. β_j) occurrences. L'algorithme *MaxOccur* décrit dans [Zaïane *et al.*, 2000] est une adaptation de *Apriori* pour le calcul de motifs fréquents avec objets récurrents. La notion de support d'une règle dans cet algorithme peut être soit relative aux transactions, soit relative aux objets pour limiter le nombre de motifs fréquents. La découverte de règles de la forme (2.1) introduit une complexité nouvelle du fait de la prise en compte de la répétition d'objets. De plus, l'application de *MaxOccur* nécessite des hypothèses très fortes sur la nature et le nombre d'objets par image. Dans leurs expérimentations, Zaïane *et al.* génèrent des images synthétiques dans lesquelles les objets sont des formes géométriques simples et dont les positions relatives sont prédéfinies. De telles hypothèses sont rarement vérifiées sur des images courantes, ce qui rend l'approche inexploitable en situation réelle.

2.2.4 La découverte de séquences fréquentes

La recherche de séquences fréquentes² a été initialement développée pour analyser un ensemble de passages en caisse dans des supermarchés. Contrairement à la découverte des règles d'association, elle exige une certaine contrainte d'ordre sur les données. Une étiquette de temps peut par exemple être associée à chaque objet ou attribut d'une base de données relationnelle. Une séquence se définit alors comme une liste ordonnée d'*itemsets*. Le problème de la découverte des séquences fréquentes a été formalisé par Agrawal et Srikant dans [Agrawal et Srikant, 1995] puis étendu dans [Srikant et Agrawal, 1996]. Il consiste à trouver, à partir d'un ensemble de transactions datées, l'ensemble des séquences présentant un support supérieur à un seuil *minsupp* fixé. Le problème de la détermination des séquences fréquentes est plus général que celui du calcul d'*itemsets* fréquents dans lequel il n'y a pas de contrainte d'ordre sur les *itemsets*.

De nombreux algorithmes ont été développés pour le calcul de séquences fréquentes [Srikant et Agrawal, 1996, Zaki, 1998, Pei *et al.*, 2001, Ayres *et al.*, 2002], mais les plus connus sont GSP³ [Srikant et Agrawal, 1996] et PrefixSpan [Pei *et al.*, 2001] qui représentent respectivement les deux principales approches du problème : les approches fondées sur la génération suivie du test des séquences candidates et celles fondées sur

²on parle aussi de séries chronologiques

³Generalized Sequential Pattern

l'analyse des préfixes communs que présentent les séquences de données de la base à traiter. Antunes et Olivera font une analyse de ces approches et proposent l'algorithme SPARSE⁴ qui exploite et combine les avantages des deux approches citées [Antunes et Oliveira, 2004].

À notre connaissance, très peu de travaux se sont intéressés aux applications de la recherche des séquences fréquentes sur les images. Hsu *et al.* introduisent un concept appelé modèle de point de vue (*Viewpoint Pattern*) qui est en réalité une adaptation du concept de séquence aux images [Hsu *et al.*, 2003]. Les images sont considérées comme des ensembles d'objets et la relation d'ordre entre deux objets est traduite par la distance entre les deux objets et la position relative d'un objet par rapport à l'autre sur l'image. Ainsi, un modèle de point de vue indique des invariants dans les positions relatives d'un objet quelconque par rapport à un autre sur une image. L'observation d'une image donnant le plan d'un bureau de travail peut par exemple révéler le modèle suivant : « les bouquins sont toujours placés à gauche de l'ordinateur ». Dans ce modèle, « bouquins » et « ordinateur » sont des objets de l'image et « placés à gauche de » représente la position relative de « bouquins » par rapport à « ordinateur ».

Hsu *et al.* mettent en œuvre un algorithme inspiré de *Apriori* pour extraire les modèles fréquents de point de vue. Ils appliquent leur méthode aux images photographiques ainsi qu'aux images des plans de cuisine mais les auteurs ne disent pas comment ils procèdent pour segmenter les images afin d'obtenir les objets sur lesquels leur méthode s'applique.

2.3 Regroupement des descripteurs visuels

Le regroupement des descripteurs visuels (ou *clustering*) est utilisé dans le domaine de l'indexation d'images pour former des groupes appelés *clusters*. Les données en entrée sont des vecteurs de plusieurs dimensions et de composantes numériques. Le principal objectif du regroupement est de découvrir parmi un ensemble \mathbb{V} de N vecteurs à n dimensions des groupes de vecteurs choisis par rapport à une distance définie dans un espace métrique E (cf. section 1.3.1, page 17). Une fois la distance définie, le *clustering* cherche à structurer les vecteurs initiaux de manière à regrouper ceux qui sont les plus proches et à séparer les plus éloignés. Plusieurs méthodes de *clustering* existent et diffèrent les unes des autres par la structuration des groupes à construire et par les algorithmes mis en œuvre pour construire les groupes. Nous faisons dans cette section un point sur les principaux algorithmes de *clustering* classés en deux catégories : les algorithmes de partitionnement et les algorithmes de classification hiérarchique. Le lecteur intéressé pourra trouver un état de l'art plus détaillé dans [Nakache et Confais, 2005].

2.3.1 Méthodes de partitionnement

Les méthodes de partitionnement créent une partition, c'est-à-dire un ensemble \mathbb{P} de k groupes de vecteurs $P_1, P_2, \dots, P_k \subset \mathbb{V}$ qui vérifient la condition suivante : pour

⁴Sequential PAttern mining with Restricted SEarch

tout vecteur $x \in \mathbb{V}$, il existe un et un seul groupe $P_i \in \mathbb{P}$ avec $i \in [1, k]$ tel que $x \in P_i$. Ainsi, l'union de tous les groupes P_i , $i \in [1, k]$ donne l'ensemble \mathbb{V} et l'intersection de deux groupes quelconques P_i et P_j est vide si $i \neq j$. Les groupes créés sont appelés classes ou *clusters*. Dans la suite de cette thèse, nous parlerons indifféremment de classes ou de *clusters*. Le nombre de classes de la partition doit être fixé au départ. Il peut être donné comme paramètre par l'utilisateur ou calculé automatiquement. Lorsque le nombre de classes n'est pas connu *a priori*, il est admis que les méthodes automatiques permettent de découvrir une partition « intuitive » des vecteurs, basée sur l'utilisation des paramètres tels que le nombre minimal de vecteurs par classe, le rayon d'une classe, *etc.* Que le nombre de classe soit connu à l'avance ou pas, le principe de partitionnement est le même. Nous allons donc supposer connus l'ensemble des vecteurs à grouper ainsi que le nombre k de classes de la partition à construire.

Une partition optimale peut être obtenue en énumérant de façon exhaustive toutes les partitions possibles. Cette manière de procéder est coûteuse du point de vue du temps de calcul. Des méthodes de partitionnement basées sur l'optimisation d'un critère permettent d'obtenir des classes assez éloignées les unes des autres en un temps raisonnable de calcul. Il s'agit de schémas de réaffectation [Forgy, 1965, Diday, 1971] qui redistribuent itérativement les vecteurs dans les k classes. Ces schémas améliorent progressivement la qualité des classes. Selon la définition des représentants des classes, les algorithmes de partitionnement par optimisation itérative sont divisés en méthodes des k -moyennes (*k-means*) et méthodes des k -représentants (*k-medoids*). Ces algorithmes construisent des classes de formes convexes sans tenir compte des données isolées (*outliers*). Dans la méthode des k -représentants, chaque *cluster* est représenté par le vecteur le plus approprié (*medoid*), ce qui rend la méthode moins sensible aux vecteurs atypiques que le *k-means*. La méthode procède par une optimisation itérative qui combine la réaffectation des vecteurs aux *clusters* avec une interversion des représentants et des autres vecteurs. La stratégie est coûteuse et donc valable uniquement pour des données de petite taille.

Les méthodes de type *k-means* les plus souvent utilisées sont celles des centres mobiles et celles des nuées dynamiques. La méthode des centres mobiles proposée par Forgy [Forgy, 1965] consiste à construire une partition en k *clusters* à partir de k vecteurs tirés au hasard et considérés comme centres initiaux. La méthode calcule ensuite les distances de chaque vecteur à chacun de ces centres et affecte les vecteurs au centre le plus proche. Les centres des *clusters* ainsi obtenus constituent de nouveaux centres (centres mobiles) qui fournissent une nouvelle partition. Le processus de réaffectation des vecteurs continue jusqu'à la stabilisation des centres. La méthode des nuées dynamiques quant à elle a été mise au point par Diday [Diday, 1971]. Elle peut être considérée comme une variante de la méthode des centres mobiles dans laquelle le représentant d'un *cluster* est soit le barycentre de celui-ci, soit un ensemble de vecteurs appelé noyau du *cluster*. Tout comme la méthode des centres mobiles, celle dite des nuées dynamiques vise à déterminer sur une base de données une partition en k *clusters* qui satisfait un critère de qualité.

2.3.2 Arbres hiérarchiques

La classification hiérarchique peut être considérée comme un partitionnement récursif dans lequel les *clusters* découverts sont stockés dans une arborescence binaire indicée appelée dendogramme. Chaque nœud non-terminal de l'arbre possède exactement deux fils qui vérifient les conditions de partitionnement : l'intersection de deux *clusters* fils est vide et leur union donne le *cluster* père (figure 2.1). La technique de classification hiérarchique fournit un système de *clusters* emboîtés dont l'hétérogénéité augmente avec la taille des *clusters* comme le montre la figure 2.1 dans laquelle l'indice de dispersion correspond à une mesure de proximité entre les vecteurs contenus dans un *cluster* à un niveau de l'arbre donné. La classification hiérarchique s'effectue jusqu'à un niveau de

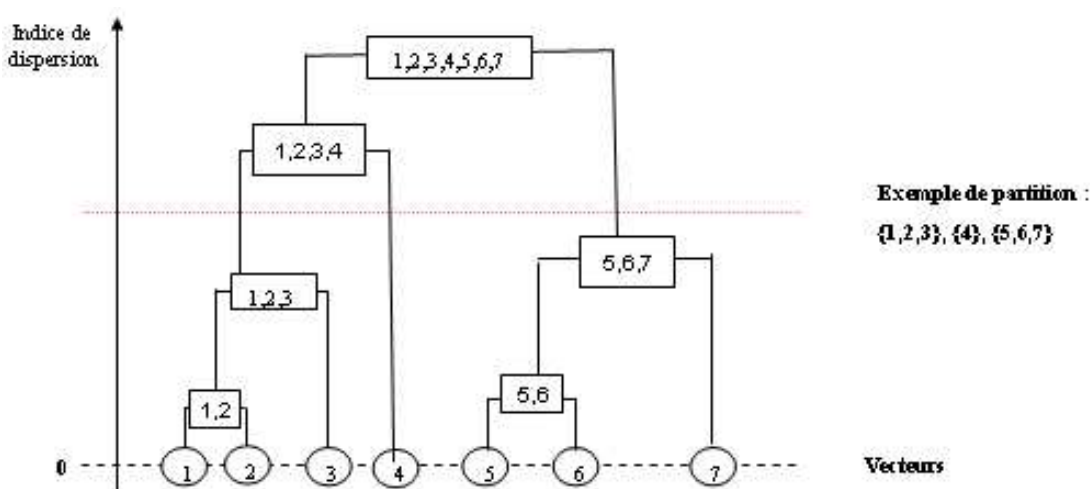


FIG. 2.1 – Exemple de hiérarchie binaire indicée construite à partir d'un ensemble de neuf vecteurs.

détail le plus important où les *clusters* ne contiennent qu'un seul vecteur. Ces *clusters* ont d'ailleurs un indice de dispersion nul. D'autre part, la hiérarchie obtenue permet de déterminer une partition des vecteurs avec un nombre de *clusters* compris entre 2 et la taille de la base de données. Graphiquement, la découverte d'une partition en k *clusters* consiste à trouver une droite horizontale, correspondant à une valeur donnée de l'indice de dispersion, qui coupe exactement k relations père-fils. La figure 2.1 donne un exemple où $k = 3$. L'arbre hiérarchique permet donc de disposer d'un grand nombre de partitions. Le soin de choisir une des partitions est laissé à l'utilisateur spécialiste des données. La méthode hiérarchique est alors plus complète que le partitionnement qu'elle englobe.

La construction d'une hiérarchie s'appuie sur la définition d'une notion de distance entre les vecteurs. Cette notion induit i) une mesure d'hétérogénéité d'un *cluster* basée sur les distances entre les vecteurs du *cluster* et ii) une mesure de dissimilarité entre deux *clusters* basée sur la distance entre un vecteur de l'un et un vecteur de l'autre. Il existe de nombreuses manières de calculer la distance entre deux *clusters*. Les plus simples

considèrent la distance entre centres de gravité, la distance minimale ou la distance maximale entre les vecteurs des deux clusters. Les plus élaborées sont basées sur la variance des vecteurs des *clusters* et peuvent être trouvées dans [Lebart *et al.*, 1984].

Parmi les méthodes de classification hiérarchique, on distingue les ascendantes et les descendantes. Les méthodes ascendantes créent un *cluster* en regroupant deux *clusters* existants sur la base d'un critère défini. Au début, il est naturel de regrouper les deux vecteurs les plus proches au sens de la dissimilarité de départ. Mais immédiatement après cette opération on peut regrouper soit des vecteurs, soit un vecteur et un *cluster*, soit deux *clusters*. Plusieurs stratégies peuvent alors s'insérer dans le schéma général suivant :

1. on définit une matrice de dissimilarités entre N vecteurs où N est la taille de la base de données. Chaque vecteur donne un *cluster* réduit à lui-même auquel la valeur 0 (indice de dispersion) est attribuée. Un *cluster* à deux éléments est ensuite constitué à partir du couple correspondant à la plus petite valeur de la matrice de dissimilarités, puis une valeur positive est attribuée à ce nouveau *cluster*. Le nombre total de *clusters* à cette étape est alors de $N - 1$;
2. à chaque itération, une valeur $h(i)$ est associée à chacun des m *clusters* disponibles. Un critère M permet de regrouper deux de ces *clusters*, puis une valeur h , supérieure ou égale à la valeur des deux *clusters* regroupés, est attribuée au groupe. Le critère M s'appuie en général sur la distance entre deux *clusters* ;
3. le processus recommence jusqu'à ce qu'il ne reste que le *cluster* regroupant tous les vecteurs de la base de données. Une valeur h supérieure à toutes les autres est enfin attribuée à ce *cluster*.

Une classification ascendante hiérarchique (CAH) est entièrement déterminée par la définition du critère M de choix pour le regroupement et la fonction de valuation h . Les approches classiques de CAH telles que EML [Sarle, 1983] ont une complexité globale de calcul pouvant aller jusqu'à $O(N^3)$ mais les approches plus récentes telles que BIRCH [Zhang *et al.*, 1996] sont de l'ordre de $O(N)$.

À l'inverse de la CAH, les méthodes descendantes divisent un *cluster* existant pour en faire deux nouveaux. Elles partent du *cluster* contenant toute la base de données et situé au sommet de l'arbre puis construisent de façon itérative des *clusters* de plus en plus petits jusqu'à l'obtention de la partition constituée des *clusters* contenant chacun un seul vecteur.

2.4 Règles d'association et indexation d'images

La technique des règles d'association consiste à générer des associations informatives à partir de la fréquence de co-occurrence de certaines valeurs dans une grande masse de données. Ces associations permettent d'expliquer les relations entre différents attributs des données relationnelles.

Dans cette thèse, nous proposons une application des règles d'association à l'indexation d'une base d'images fixes. Notre approche s'inscrit dans le cadre de la transformation des données multimédias et de l'application des techniques classiques d'extraction

de règles. Nous utilisons plusieurs descripteurs et les images sont regroupées en *clusters* pour chaque descripteur. Nous identifions une image à une transaction et un numéro de *cluster* d'un type de descripteur donné à un attribut. Le problème de la répétition d'objets ou d'attributs ne se pose pas puisque les *clusters* sont identifiés de manière unique pour chaque type de descripteur.

D'autres travaux exploitent les règles d'association dans un contexte d'indexation du contenu. Parmi ceux-ci, citons les travaux de Djeraba [Djeraba, 2003] et ceux de Bouet et Khenchaf [Bouet et Khenchaf, 2002] qui nous permettent de resituer notre approche. Djeraba utilise une technique d'indexation voisine de la nôtre en ce sens qu'elle combine la construction des *clusters* et la détermination des règles d'association [Djeraba, 2003]. Sa base de travail est formée de plusieurs sous-groupes d'images classées et étiquetées thématiquement (animaux, plantes, ...) et l'idée consiste à caractériser ces groupes (ou *clusters* d'images) par des règles d'association définissant une sémantique entre les valeurs des descripteurs et les labels des classes d'images. L'approche est cependant contrainte par la construction semi-automatique d'un dictionnaire nécessitant une labellisation sémantique des *clusters*. Notre approche ne limite pas le nombre ainsi que le type de descripteurs utilisables et aucun étiquetage n'est nécessaire puisque nous travaillons au niveau pixel, c'est-à-dire « sans sémantique ajoutée ». Notre stratégie combine uniquement le calcul des *clusters* et l'extraction des règles d'association. Elle est donc entièrement non supervisée et convient aux grandes bases d'images.

Le travail de Bouet et Khenchaf [Bouet et Khenchaf, 2002] tente de capturer le niveau sémantique par la découverte des relations existantes entre les images d'une base de données. Ces images sont regroupées en *clusters* pour chacun des descripteurs disponibles. Les règles d'association, utilisées à titre descriptif, sont calculées pour faciliter l'exploitation des *clusters*. Leur antécédent correspond à une valeur de propriété visuelle et leur conséquent à un *cluster*. Les *clusters* réduisent l'espace de recherche alors que les règles d'association permettent de classer automatiquement toute nouvelle image dans le *cluster* auquel elle a des chances d'appartenir. Notre approche, au contraire, utilise les *clusters* et les règles d'association pour choisir les descripteurs pertinents et pour réduire l'espace de recherche.

L'approche d'Aouiche *et al.* [Aouiche *et al.*, 2003], destinée à l'auto-administration d'une base de données relationnelle, fonde le calcul de l'index sur les ensembles d'attributs les plus fréquemment utilisés dans les requêtes. Cette approche détermine des relations entre attributs (sous forme de motifs fréquents) à partir de la fréquence d'utilisation de ceux-ci. Elle fait donc une administration à partir d'un journal d'utilisation de la base (intervention indirecte de l'utilisateur) alors que nos traitements sont entièrement transparents à l'utilisateur. Aouiche *et al.* [Aouiche *et al.*, 2003] appliquent leur méthode dans un contexte plus classique, c'est-à-dire sans lien avec l'indexation du contenu.

Nous détaillons dans cette section comment les *clusters* de descripteurs sont calculés puis nous formalisons la notion de règles d'association entre ces *clusters*. Nous montrons enfin comment la découverte des règles et le calcul des *clusters* s'intègrent dans une chaîne de traitement qui permet d'améliorer la recherche par similarité.

2.4.1 Calcul des *clusters* de descripteurs

Le *clustering* en tant que prétraitement permet d'organiser les descripteurs d'images pour estomper les problèmes liés aux très grandes bases d'images. Ainsi, au cours de la recherche, le but est d'accéder plus facilement les *clusters* pré-calculés au lieu de parcourir systématiquement toute la base. Il existe cependant plusieurs techniques de *clustering* selon les deux grandes approches déjà citées : les approches par partitionnement et les approches hiérarchiques (cf. section 2.3). Nous décrivons dans cette section l'algorithme *k-means* utilisé comme approche par partitionnement dans notre travail. Par contre, nous n'utilisons pas les méthodes classiques de classification hiérarchique puisqu'elles coûtent cher en temps de calcul. En contre partie, nous adaptons le *M-Tree* [Ciaccia *et al.*, 1997] qui est une technique d'indexation permettant de construire un arbre dont nous utilisons les feuilles comme des *clusters*.

2.4.1.1 Méthode *k-means*

La méthode *k-means* utilise initialement k *clusters* contenant les k premiers vecteurs qui servent de centres des *clusters*. Chaque vecteur restant est ensuite ajouté au *cluster* dont le centre est le plus proche. Les k *clusters* sont produits à l'issue du *k-means* de façon à minimiser la fonction objective

$$E = \sum_{j=1}^k \sum_{x_i \in C_j} (x_i - g_j)^2$$

où x_i est un point quelconque du *cluster* C_j et g_j le point moyen de C_j , $j = 1..k$. Le paramètre d'entrée du *k-means* est le nombre k de *clusters* et la mesure de similarité que nous utilisons est soit la distance L_1 , soit la distance L_2 selon le type de descripteur *MPEG-7* choisi (voir annexe A). L'algorithme se résume aux quatre étapes suivantes :

1. choisir k vecteurs aléatoirement pour initialiser les centres des *clusters*;
2. affecter chaque vecteur au centre le plus proche. Il en résulte une partition en k *clusters* C_1, C_2, \dots, C_k ;
3. recalculer les centres de chacun de ces *clusters*. Chaque *cluster* est représenté par son centre qui, en général, n'est pas un vecteur de la base de données. Pour contourner ce problème de manière à rester dans le domaine de description des images, nous remplaçons ce centre par un vecteur de la base qui lui est le plus proche selon la mesure de similarité utilisée (distance L_1 ou L_2);
4. les étapes 2 et 3 sont répétées jusqu'à ce que le choix des centres des *clusters* se stabilise.

Il existe de nombreuses méthodes de type *k-means*. Elles diffèrent les unes des autres suivant le mode de sélection des vecteurs initiaux, le calcul des centres des *clusters* ou le calcul de la distance entre les vecteurs. La variante que nous venons de présenter correspond à celle que nous avons effectivement utilisée dans le cadre de nos travaux donc l'objectif est tout simplement d'obtenir des groupes de descripteurs d'images.

La complexité du temps de calcul de l'algorithme *k-means* est fonction du nombre de distances calculées $O(lkN)$ où l est le nombre d'itérations, k le nombre de *clusters* et N le nombre de vecteurs de la base de données. En principe, l et k sont très petits par rapport à N puisque nous travaillons sur des grandes bases de plusieurs milliers d'images. La complexité du *k-means* peut donc être ramenée à $O(N)$.

Le *k-means* est une méthode simple à mettre en œuvre et peu coûteuse en temps de calcul. Cependant il ne gère pas les points isolés (*outliers*) et la stabilité dans le choix des centres des *clusters* ne correspond pas toujours à la partition optimale (problème de minimum local). Le *k-means* est par ailleurs sensible aux valeurs initiales des paramètres.

2.4.1.2 Le *M-Tree*

Le *M-Tree*, proposé par Ciaccia *et al.* dans [Ciaccia *et al.*, 1997], est une méthode d'indexation multidimensionnelle qui cherche à réduire le coût CPU des calculs des distances ainsi que le nombre d'entrées/sorties au cours de la recherche. Il a également l'avantage d'opérer dans un espace simplement métrique, contrairement aux autres techniques d'indexation telles que le *SR-Tree* [Katayama et Satoh, 1997] qui imposent que l'espace de description soit muni d'une distance particulière comme la distance euclidienne. Pour éviter les calculs inutiles de distance lors de la recherche de plus proches voisins, le *M-Tree* exploite la propriété de l'inégalité triangulaire vérifiée dans tous les espaces métriques ainsi que certaines distances pré-calculées. Nous nous sommes intéressés dans ce travail à la structure d'arbre construit par le *M-Tree* pour en dériver les *clusters* qui correspondent aux feuilles de l'arbre.

Le *M-Tree* partitionne les vecteurs sur la base de leurs distances relatives mesurées par une fonction de distance d et stocke ces objets dans des nœuds de taille fixe. Les feuilles de l'arbre construit par le *M-Tree* stockent toutes les images représentées par des vecteurs alors que les nœuds internes de l'arbre contiennent des objets de redirection (*routing object*). Un objet de redirection O_r comprend le vecteur de descripteur et d'autres informations permettant de faciliter la recherche, notamment par l'application de l'inégalité triangulaire. Il s'agit plus particulièrement :

- d'un pointeur $ptr(T(O_r))$ sur la racine du sous-arbre $T(O_r)$ correspondant au nœud qui contient l'objet de redirection O_r ;
- d'un rayon de couverture $r(O_r)$ (*covering radius*), c'est-à-dire la distance entre l'objet de redirection O_r et l'objet le plus éloigné dans le sous-arbre pointé par $ptr(T(O_r))$;
- et de la distance entre O_r et l'objet de redirection $P(O_r)$ contenant le pointeur référant le nœud de O_r .

$P(O_r)$ est appelé objet parent de O_r (*parent object*). Une entrée de la feuille de l'arbre contient le vecteur de descripteur de l'image, une référence sur l'image présente sur disque et la distance entre l'entrée et son objet parent. Le rayon de couverture n'est pas nécessaire aux feuilles puisque les objets de celles-ci ne pointent sur aucun sous-arbre.

L'algorithme de construction du *M-Tree* indique comment les objets sont insérés et supprimés, et comment la saturation est gérée au niveau des nœuds de l'arbre. L'arbre du *M-Tree* est construit par insertions successives de façon à minimiser les rayons de

couverture des objets de redirection. Le principal critère d'insertion d'un nouvel objet O_n consiste pour tout sous-arbre $T(O_r)$ à ne parcourir que les nœuds de $T(O_r)$ pour lesquels $d(O_r, O_n) \leq r(O_r)$. Si plusieurs sous-arbres vérifient ce critère, on choisit d'insérer dans le sous-arbre pour lequel O_r est le plus proche de O_n . Lorsqu'aucun sous-arbre ne vérifie le critère, on insère O_n dans le sous-arbre qui minimise la différence $d(O_r, O_n) - r(O_r)$.

L'arbre du *M-Tree* évolue dynamiquement de façon ascendante au fur et à mesure des insertions. La saturation d'un nœud N est gérée par l'allocation d'un nouveau nœud N' au même niveau que N . L'algorithme de fractionnement des nœuds saturés élit ensuite deux objets de redirection parmi les objets de N puis répartit les objets restants entre N et N' de façon à minimiser le volume des deux régions obtenues ainsi que leur taux de chevauchement. Plusieurs politiques de répartition existent à cet effet. Elles dépendent à la fois de la manière d'élire les objets de redirection et de la méthode de redistribution des objets dans les nœuds. Ces politiques visent toutefois à optimiser le taux d'occupation de l'espace par la structure d'arbre construite.

L'avantage du *M-Tree* est que les feuilles, considérées comme des *clusters*, sont compactes puisque les taux d'occupation de l'espace sont optimisés. L'arbre étant construit par insertions successives, les mises à jour sont simples. En revanche, le choix des objets de redirection n'est pas toujours aisé en raison de la difficulté de calculer les taux de chevauchement dans des espaces métriques non euclidiens.

2.4.2 Découverte des règles à partir des *clusters* de descripteurs

Nous partons d'une base d'images fixes (de type tout venant) à partir de laquelle nous calculons l'ensemble des valeurs des descripteurs retenus pour sa caractérisation. La base d'images est ensuite organisée en *clusters* selon le *k-means* ou le *M-Tree* tel que décrit dans la section 2.4.1. Après l'étape de *clustering*, nous procédons à la découverte des règles d'association. Le processus de découverte des règles se fait hors-ligne.

Soit $D = \{d_1, d_2, \dots, d_m\}$ l'ensemble de m types de descripteurs pouvant être calculés par le système. Un *cluster* est identifié par un numéro et le type de descripteur pour lequel il est calculé. Nous désignons par $(n_j(I), d_j)$ le *cluster* d'appartenance de l'image I pour le descripteur d_j et nous associons à I une transaction dont les attributs sont les *clusters* auxquels l'image appartient.

Une base d'images notée B peut donc être décrite par :

$$B = \{I | I = \{(n_1(I), d_1), (n_2(I), d_2), \dots, (n_m(I), d_m))\}\}$$

Il est possible d'appliquer un algorithme de calcul de motifs fréquents (tel que *Apriori*) sur la base B ainsi décrite et d'extraire des règles d'association pour obtenir des implications (avec un ou plusieurs éléments en partie gauche) de la forme :

$$r : (n_1, d_1) [\wedge (n_2, d_2) \wedge \dots \wedge (n_p, d_p)] \rightarrow (n_k, d_k) < \text{supp}, \text{conf} > \quad (2.2)$$

avec $d_i \neq d_j$ si $i \neq j$; $i, j \in \{1, 2, \dots, p\} \cup \{k\}$

où *supp* et *conf* sont respectivement le support et la confiance de la règle r , exprimés en pourcentages. La règle (2.2) signifie qu'une image appartenant à tous les *clusters* de

la partie gauche se trouve aussi dans (n_k, d_k) avec les taux *supp* et *conf* respectifs de support et de confiance. Les seuils minimaux de support et de confiance sont fixés et permettent de juger la qualité des règles découvertes.

2.4.3 Processus d'organisation des images

Nous nous intéressons dans cette section à l'aspect pratique de notre approche d'organisation des images fixes. L'intégration de la classification non supervisée et de la découverte des règles d'association dans notre approche s'articule autour de trois principales étapes qui sont : la description des images, la classification automatique des images au sein des *clusters*, et la génération des règles d'association entre *clusters* de descripteurs (figure 2.2).

La **description des images** calcule tous les descripteurs à partir d'un ensemble de fichiers d'images stockés dans un répertoire. Dans le cadre de cette thèse, nous avons travaillé avec des descripteurs globaux *MPEG-7* de couleur, de texture, et de forme. Notons cependant que tout autre type de descripteur peut convenir, tant qu'une image est décrite par un vecteur. Notre système d'organisation génère un fichier décrivant toutes les images pour chacun des types descripteurs utilisés (étape I1 de la figure 2.2).

La **classification automatique des images** organise les fichiers de descripteurs en *clusters* pour chaque type de descripteur, calcule les paramètres d'accès à chaque *cluster* formé et les stocke dans un fichier (étape I2). Chaque *cluster* est identifié par un numéro et le type du descripteur pour lequel il est calculé. Dans cette étape de l'organisation des images, tout algorithme de calcul de *clusters* peut être utilisé. Toutefois, pour évaluer la sensibilité de l'approche en fonction de la technique de *clustering* utilisée, nous avons choisi de travailler avec un algorithme de partitionnement (le *k-means*) et un algorithme qui construit une hiérarchie assez particulière dont nous exploitons uniquement les feuilles (le *M-Tree*).

La **génération des règles d'association** utilise les fichiers de *clusters* produits à l'issue de la classification automatique pour extraire des relations entre *clusters* sous forme de règles d'association (étape I3). Une règle d'association ainsi générée est une implication de la forme (2.2). À l'issue du processus d'indexation, nous avons donc deux fichiers : le fichier de *clusters* et celui des règles d'association.

2.5 Synthèse

Ce chapitre a rappelé les différentes tâches de la fouille de données et illustré les applications de chacune d'elles à l'organisation et à la recherche d'images fixes par le contenu. Les techniques de classification supervisée tentent de répartir les images dans des classes connues à l'avance. Dans la réalité, il est toutefois difficile de décrire précisément les classes dans lesquelles les images sont réparties, surtout lorsque la base d'images est grande. La classification non supervisée répartit les images dans des classes appelées *clusters* et le principal critère de répartition assure tout simplement que les images les plus proches au sens d'une mesure de similarité donnée se retrouvent dans le même *cluster* et que les *clusters* sont le plus distincts et éloignés possible les uns

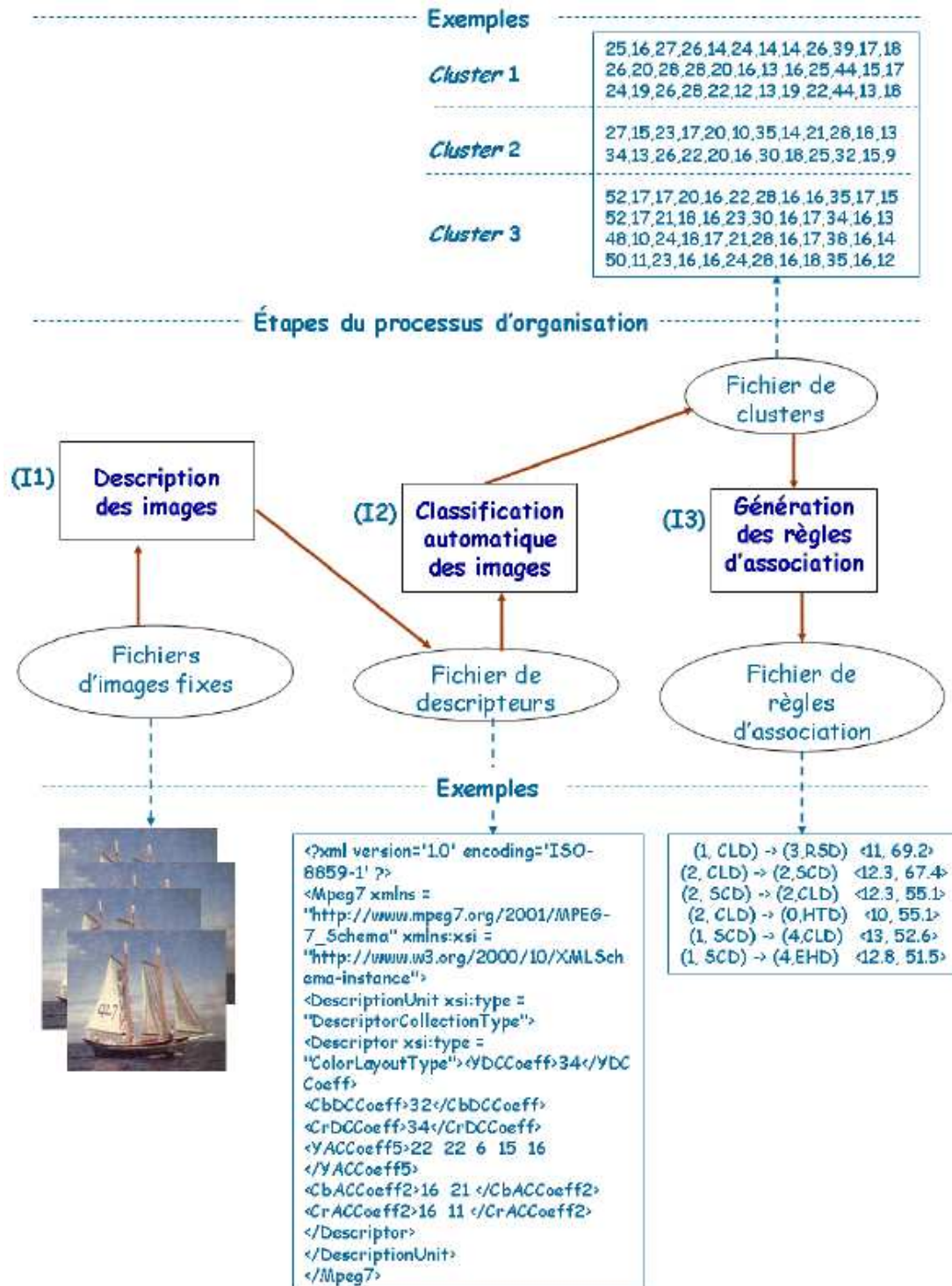


FIG. 2.2 – Processus d'organisation des images fixes.

des autres. Les règles d'association, appliquées aux images, cherchent à identifier des implications qui expliquent les relations entre images d'une collection donnée.

La classification non supervisée présente un très grand intérêt pour l'organisation d'images par le contenu, puisqu'elle s'appuie sur le critère de similarité, essentiel pour la comparaison des images entre elles. Nous avons décrit dans ce chapitre les principes de regroupement en *clusters* des descripteurs visuels. Nous avons ensuite défini la notion de règle d'association dans le contexte des *clusters* de descripteurs puis indiqué le moyen de combiner les deux tâches de la fouille de données que sont la classification non supervisée et la découverte des règles d'association pour l'organisation des images par le contenu. Cette combinaison s'articule dans une chaîne de traitement que nous avons présentée. Parmi les techniques de fouille, notre travail a consisté à étudier celles qui seraient les plus avantageuses pour améliorer l'organisation et la recherche des images par le contenu. Le chapitre qui suit développe les stratégies de recherche basées sur l'utilisation conjointe des techniques de *clustering* et des règles d'association abordées dans ce chapitre.

Chapitre 3

Stratégies de recherche par le contenu basées sur la découverte de règles d'association, l'ordonnancement et le filtrage des *clusters* de descripteurs d'images

Pour éviter de parcourir de façon séquentielle et aveugle de grandes bases d'images, nous proposons deux approches : l'une basée sur les règles d'association et l'autre sur des règles de filtrage. Ces deux approches sont complémentaires et peuvent être combinées. Dans ce chapitre, nous présentons également comment une recherche par le contenu peut être adaptée pour proposer des résultats intermédiaires qui sont fusionnés de façon progressive avec l'avantage pour l'utilisateur, d'une part, de ne pas attendre que toute la base ait été parcourue avant de fournir un résultat et, d'autre part, de lui permettre de stopper la requête en cours d'exécution.

3.1 Introduction

La recherche d'images fixes par le contenu au niveau pixel exploite et combine des descripteurs visuels dont le coût de calcul est d'autant plus important que la taille de la base d'images est grande. Le parcours exhaustif d'un très grand nombre d'images pour la recherche par similarité des plus proches voisins d'une requête par l'exemple peut s'avérer inefficace en raison de l'attente inutile imposée à l'utilisateur. D'autre part, un sous-ensemble de descripteurs pourrait suffire à répondre à une recherche par similarité beaucoup plus rapidement, tout en gardant une qualité acceptable des résultats de recherche. Notre hypothèse est que l'organisation des images à l'indexation ainsi que la planification des descripteurs pendant la recherche offrent plus de flexibilité à l'utilisateur, en ce sens qu'il n'attend plus inutilement et peut même arrêter la recherche s'il

est satisfait du résultat partiel retourné.

Il apparaît donc intéressant d'élaborer des stratégies automatiques pour aider les utilisateurs à faire une recherche par le contenu efficace sans se préoccuper de la nature ou de la qualité des descripteurs. Nous proposons dans ce cadre une méthode de recherche d'images fixes par le contenu fondée sur une utilisation automatique, sélective, et progressive des descripteurs visuels disponibles. Le but de la méthode proposée est de pouvoir optimiser les performances de la recherche par le contenu en évitant le parcours systématique de toutes les images de la base et pour tous les types de descripteurs globaux. Notre travail combine deux techniques de fouille : le *clustering* et la découverte des règles d'association pour améliorer le temps de la recherche dans de grandes base d'images sans véritablement dégrader les résultats que l'on obtiendrait par une recherche séquentielle aveugle. Nous décrivons d'abord notre chaîne de traitement d'une requête par l'exemple. Nous précisons ensuite notre terminologie, puis nous montrons comment l'organisation des images en *clusters* et la découverte des règles d'association s'intègrent dans le processus de recherche par utilisation progressive des descripteurs du contenu et améliorent, comme nous le prévoyions, l'efficacité de la recherche.

3.2 Chaîne de traitement d'une requête par l'exemple

Nous présentons une stratégie d'utilisation sélective, progressive, et automatique des descripteurs visuels pour une recherche par le contenu dans une base d'images fixes. Cette stratégie est fondée à l'indexation sur le *clustering* d'images pour chaque type de descripteur visuel et sur la découverte des règles d'association entre les *clusters* de ces types de descripteurs. À la recherche, elle se décompose en plusieurs étapes (figure 3.1) :

1. l'utilisateur soumet une image-requête au système et tous les descripteurs de la requête sont calculés dans la limite des types de descripteurs gérés par la base d'images ;
2. les mesures des caractéristiques des *clusters* dépendantes de la requête sont calculées. Nous évaluons la distance du centroïde du *cluster* à l'image-requête ainsi que les distances minimum et maximum de la requête au *cluster*. Nous estimons aussi le coût de parcours d'un *cluster*, proportionnel au nombre d'images du *cluster* ;
3. les *clusters* sont triés sur la base de leurs caractéristiques combinées en une mesure d'éligibilité ;
4. les règles d'association et de filtrage sont appliquées pour sélectionner les *clusters* les plus pertinents et pour accélérer le processus de recherche ;
5. les *clusters* sélectionnés sont parcourus dans l'ordre de leur éligibilité pour produire des résultats le plus vite possible ;
6. les résultats intermédiaires sont présentés à l'utilisateur et la requête progressive reprend à l'étape (4) jusqu'à ce qu'il n'y ait plus de *clusters* à parcourir. Une fréquence de présentation des résultats peut être fixée et adaptée en fonction du nombre de résultats intermédiaires requis et du nombre total de *clusters* à parcourir.

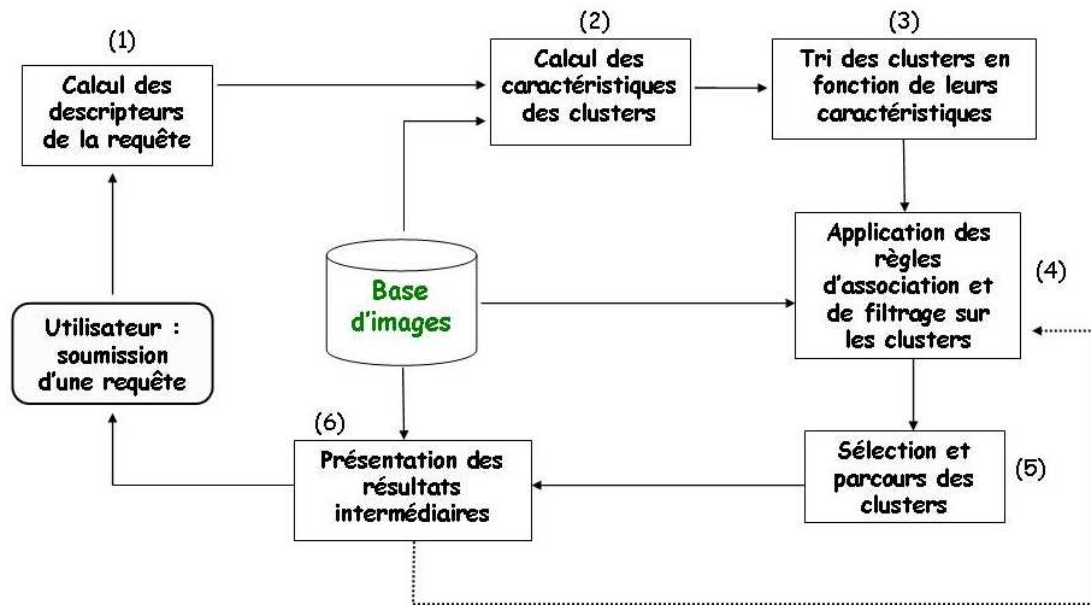


FIG. 3.1 – Scénario de recherche progressive par le contenu.

La suite de ce chapitre formalise les principales notions des étapes (2) à (5). Certaines caractéristiques des *clusters* telles que leur taille peuvent être calculées à la phase d'indexation. D'autres plus dépendantes de l'image-requête sont évaluées pendant la recherche. Toutes les mesures des caractéristiques sont ensuite normalisées pour les rendre comparables. La notion d'éligibilité est définie pour combiner les mesures normalisées afin d'induire un ordre sur les *clusters*. Cet ordre est suivi au cours de la recherche progressive. Nous décrivons aussi comment les règles d'association et les règles de filtrage sont utilisées pour réduire le nombre de *clusters* traités pendant la recherche.

3.3 Définitions de base

Nous présentons la définition de quelques termes utiles à la compréhension du principe de la méthode que nous avons mis au point. Les notations de cette section seront conservées tout le long de ce chapitre.

Définition 3.1 (Descripteur visuel) *Un descripteur visuel est un ensemble d'informations qui résumant les caractéristiques visuelles d'une image au niveau pixel. Ces caractéristiques peuvent être soit globales (couleur, texture, forme), soit locales (points d'intérêt).*

Nous travaillons essentiellement avec les caractéristiques globales représentées par des vecteurs multidimensionnels de réels ou d'entiers. Dans notre étude, tous les vecteurs d'un même type de descripteur ont la même dimension.

Définition 3.2 (Base d'images fixes) Soit $D = \{d_1, d_2, \dots, d_m\}$ l'ensemble des m types de descripteurs d'images fixes. La base d'images B est définie comme l'union des $n_c(d_i)$ clusters C_{ij} de chaque type de descripteur d_i présent de la base ($d_i \in D$). Plus formellement,

$$\forall d_i \in D, i = 1..card(D), B = \bigcup_{j=1..n_c(d_i)} C_{ij} \quad (3.1)$$

Une image appartient à la base B si sa description selon chaque type de descripteur présent dans la base appartient à un cluster.

$$I \in B \text{ si } \forall d_i \in D, \exists C_{ij}, d_i(I) \in C_{ij}$$

Définition 3.3 (Requête par l'exemple) Une requête par l'exemple $I_q = (n_q, O_q, W_q, D_q)$ dans l'espace multidimensionnel S d'un type de descripteur quelconque $d_i \in D$ de dimension d_s est composée des informations suivantes :

- un nombre de points n_q dans l'image-requête I_q
- un ensemble de n_q points $O_q = \{O_q^{(1)}, \dots, O_q^{(n_q)}\}$ dans l'espace multidimensionnel S du type de descripteur d_i
- un ensemble de n_q poids $W_q = \{w_q^{(1)}, \dots, w_q^{(n_q)}\}$, le i^e poids $w_q^{(i)}$ étant associé au i^e objet $O_q^{(i)}$ ($0 \leq w_q^{(i)} \leq 1, \sum_{i=1}^{n_q} w_q^{(i)} = 1$)
- une fonction de distance D_q , qui, pour un point O donné dans l'espace S , calcule la distance entre la requête et le point. On suppose que D_q est une distance pondérée L_p , i.e., pour une valeur donnée de p , la distance entre deux points O_1 et O_2 dans S est définie telle que¹ :

$$D_q = \left(\sum_{j=1}^{d_s} \mu_q^{(j)} (|O_1[j] - O_2[j]|)^p \right)^{1/p} \quad (3.2)$$

où $\mu_q^{(j)}$ est le poids associé à la j^e dimension de l'espace S du type de descripteur d_i ($0 \leq \mu_q^{(j)} \leq 1, \sum_{j=1}^{d_s} \mu_q^{(j)} = 1$).

D_q spécifie quelle mesure L_p utiliser (i.e., la valeur de p) et les valeurs des poids par dimension. Nous utilisons la fonction de distance D_q pour construire la fonction de distance agrégée $D_q(I_q, O)$ entre les objets multiples de la requête O_q et l'objet O (dans S). $D_q(I_q, O)$ est une fonction agrégée des distances entre O et les objets $O_q^{(i)} \in O_q$ dans l'espace S des descripteurs de type d_i telle que :

$$D_q(I_q, O) = \sum_{i=1}^{n_q} w_q^{(i)} D_q(O_q^{(i)}, O) \quad (3.3)$$

¹Cette hypothèse est générale pour la plupart des fonctions de distance utilisées. La distance de Manhattan, la distance Euclidienne, et la distance au rectangle englobant sont des cas particuliers de distance L_p .

Nous utilisons une somme pondérée comme fonction d'agrégation. Toute autre fonction peut être utilisée tant qu'elle est pondérée et monotone. La recherche se fait dans l'espace multidimensionnel des vecteurs d'un type de descripteur donné et la similarité entre les images est traduite par une distance dans cet espace multidimensionnel.

Définition 3.4 (Sous-requête) *Une sous-requête est une recherche de plus proches voisins dans un sous-ensemble de clusters, quel que soit le type de descripteur concerné. Si le sous-ensemble se réduit à un seul cluster, nous parlons de sous-requête minimale.*

3.4 Principe de sélection des règles d'association

Dans le cas d'une recherche par l'exemple (*QBE - Query-By-Example*), l'utilisateur soumet au système une image-requête. Son but est de retrouver toutes les images similaires à l'image-requête selon certains critères visuels. L'utilisateur peut savoir choisir (voire pondérer) les descripteurs visuels tels que ceux proposés dans la norme *MPEG-7* comme critères de recherche. Mais, il arrive le plus souvent qu'il n'ait aucune idée des descripteurs les plus discriminants pour sa requête. Comme nous l'avons déjà mentionné, dans bien des cas, il s'avère inutile de faire la recherche sur tous les critères (en supposant qu'ils aient été déjà précalculés) sur l'ensemble des images. Pour ce type de recherche, notre travail est, en particulier, de proposer les descripteurs qui seraient les plus adaptés et de leur affecter des priorités utiles pour optimiser l'exécution de la requête.

Nous définissons une règle d'association entre plusieurs *clusters* notés $C_{i_k j_k}$, $i_k = 1..m$, $j_k = 1..n_c(d_{i_k})$ avec $k = 0..p$, $p \leq m$ (m étant le nombre total de types de descripteurs et $n_c(d_{i_k})$ le nombre de *clusters* pour le type de descripteur d_{i_k}) de la façon suivante :

$$r : C_{i_1 j_1} \wedge C_{i_2 j_2} \wedge \dots \wedge C_{i_p j_p} \rightarrow C_{i_0 j_0} < \text{supp}, \text{conf} > \quad (3.4)$$

avec $i_k \neq i_{k'}$ si $k \neq k'$; $k, k' \in \{0, 1, 2, \dots, p\}$

La règle (3.4) signifie qu'une image appartenant à la fois aux *clusters* $C_{i_1 j_1}$, $C_{i_2 j_2}$, \dots , et $C_{i_p j_p}$ se trouve aussi dans $C_{i_0 j_0}$ avec les taux *supp* et *conf* respectifs de support et de confiance.

Les plus proches voisins d'une image-requête I_q sont calculés à partir des *clusters* les plus proches de I_q pour chaque type de descripteur. Notons C_S l'ensemble des *clusters* les plus proches de I_q . Le nombre $\text{card}(C_S)$ de *clusters* de C_S dépend du nombre de plus proches voisins recherchés. Le but des règles d'association est de trouver des relations entre les *clusters* de C_S de façon à éliminer de la recherche les *clusters* les moins pertinents ainsi que les types de descripteurs correspondants. Soit R l'ensemble des règles calculées au cours de l'organisation de la base d'images (phase hors-ligne). Pour tout $r \in R$, notons C_r l'ensemble des *clusters* présents dans l'expression de la règle r . La liste des notations utilisées est regroupée dans le tableau 3.1.

La règle r est sélectionnée si $C_r \subseteq C_S$, ce qui signifie que tous les *clusters* constituant l'expression de r sont parmi les plus proches de l'image-requête I_q .

Dans l'exemple de la figure 3.2, l'ensemble des *clusters* sélectionnés pour l'image-requête I_q est $C_S = \{C_{12}, C_{23}, C_{33}\}$ et l'ensemble des règles d'association du système

I_q	Image-requête
C_{ij}	Cluster j du type de descripteur i
C_S	Ensemble des <i>clusters</i> les plus proches de I_q
$card(C_S)$	Nombre d'éléments de C_S
r	Règle d'association
C_r	Ensemble des <i>clusters</i> présents dans l'expression de la règle d'association r
R	Ensemble des règles d'association calculées hors-ligne

TAB. 3.1 – Rappel des notations utilisées.

est $R = \{r_1 : C_{33} \wedge C_{23} \rightarrow C_{12}; r_2 : C_{13} \wedge C_{21} \rightarrow C_{32}\}$. On a $C_{r_1} = \{C_{12}, C_{23}, C_{33}\}$ et $C_{r_2} = \{C_{13}, C_{21}, C_{32}\}$. On constate que $C_{r_1} \subseteq C_S$. La règle r_1 est donc sélectionnée et la recherche se fait uniquement dans les *clusters* C_{23} et C_{33} (voir figure 3.2). Ainsi, une application pratique dans la recherche par le contenu sera par exemple d'ignorer les *clusters* en partie droite des règles sélectionnées. On évite ainsi d'examiner plusieurs fois les mêmes images de la base.

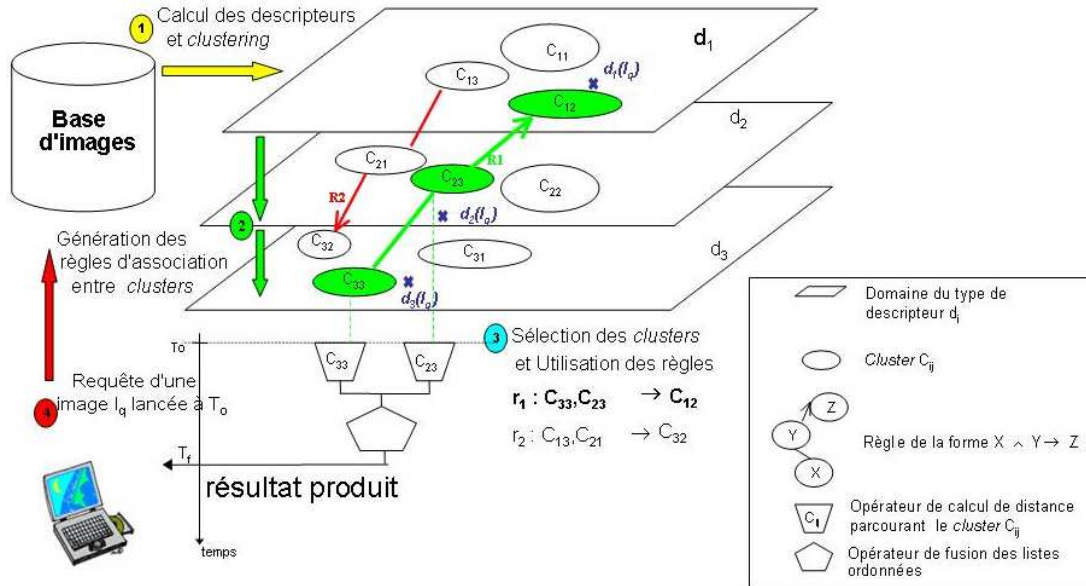


FIG. 3.2 – Exemple de sélection des règles d'association.

Le calcul de l'ensemble R des règles d'association a été détaillé au chapitre 2. Il reste à chercher comment déterminer C_S , ensemble des *clusters* les plus proches de I_q . Pour une recherche de plus proches voisins, C_S doit inclure les *clusters* susceptibles de contenir les plus proches voisins de I_q . Il pourrait aussi inclure tous les *clusters* de tous les types de descripteurs. La question est donc de savoir par quel sous-ensemble de *clusters* commencer la recherche. Dans tous les cas, les *clusters* de C_S doivent être

parcourus dans un certain ordre et il est intéressant de trouver l'ordre qui accélère l'obtention des plus proches voisins de I_q . Nous avons été amenés à définir une stratégie de calcul progressif des *clusters* de C_S .

3.5 Recherche progressive par le contenu

Le parcours progressif d'une base d'images préalablement organisée en *clusters* de descripteurs consiste à agréger régulièrement des résultats intermédiaires et à les retourner à l'utilisateur au fur et à mesure de la progression de la recherche. La recherche par le contenu nécessite généralement le parcours exhaustif de toute la base d'images avant de pouvoir fournir un résultat à l'utilisateur. Face à ce constat, notre objectif est d'améliorer le traitement d'une requête par le contenu en proposant :

1. une stratégie de planification des requêtes qui définisse un ordre optimal de parcours des *clusters* d'images,
2. la possibilité de fournir à l'utilisateur des résultats intermédiaires qui soient rafraîchis au fur et à mesure du parcours progressif de la base,
3. et la possibilité à l'utilisateur de stopper l'exécution de la requête s'il juge que le dernier résultat intermédiaire retourné est pertinent pour sa recherche.

Le principe général d'une recherche progressive, illustré à la figure 3.3, consiste à soumettre, à l'instant initial T_o , une image-requête I_q qui est décomposée en une série de sous-requêtes sur des portions de la base d'images (c'est-à-dire des *clusters* de descripteurs) ciblées pour optimiser le processus de la recherche selon les priorités éventuellement données par les règles d'association qui ont été préalablement extraites lors de la phase d'indexation.

Soit une image-requête I_q . Supposons un ordre partiel sur les instants initial, intermédiaires et final de l'exécution de la requête tel que : $T_0 < T_1 < T_2 < \dots < T_f$.

Définition 3.5 (Requête progressive) Une requête progressive, notée $\vec{Q}(I_q, T_0, T_f)$, soumise à l'instant initial T_0 , et terminée à l'instant T_f , est une requête par l'exemple dont les résultats intermédiaires produits par chaque sous-requête (appelés résultats instantanés) sont régulièrement fusionnés et envoyés à l'utilisateur aux instants $T_i \in]T_0, T_f[$ tout au long de l'exécution de la requête.

Définition 3.6 (Résultat intermédiaire) Un résultat intermédiaire noté $RI(I_q, T_i)$ d'une requête progressive est la réponse à l'instant $T_i \in]T_0, T_f[$ du système de recherche après le parcours d'un sous-ensemble donné de *clusters*.

Exécuter une requête progressive $\vec{Q}(I_q, T_0, T_f)$ revient à retourner à l'instant T_1 un résultat intermédiaire noté $RI(I_q, T_1)$ et à exécuter la requête progressive $\vec{Q}(I_q, T_1, T_f)$. Nous écrivons dans ce cas :

$$\vec{Q}(I_q, T_0, T_f) \iff RI(I_q, T_1), \vec{Q}(I_q, T_1, T_f)$$

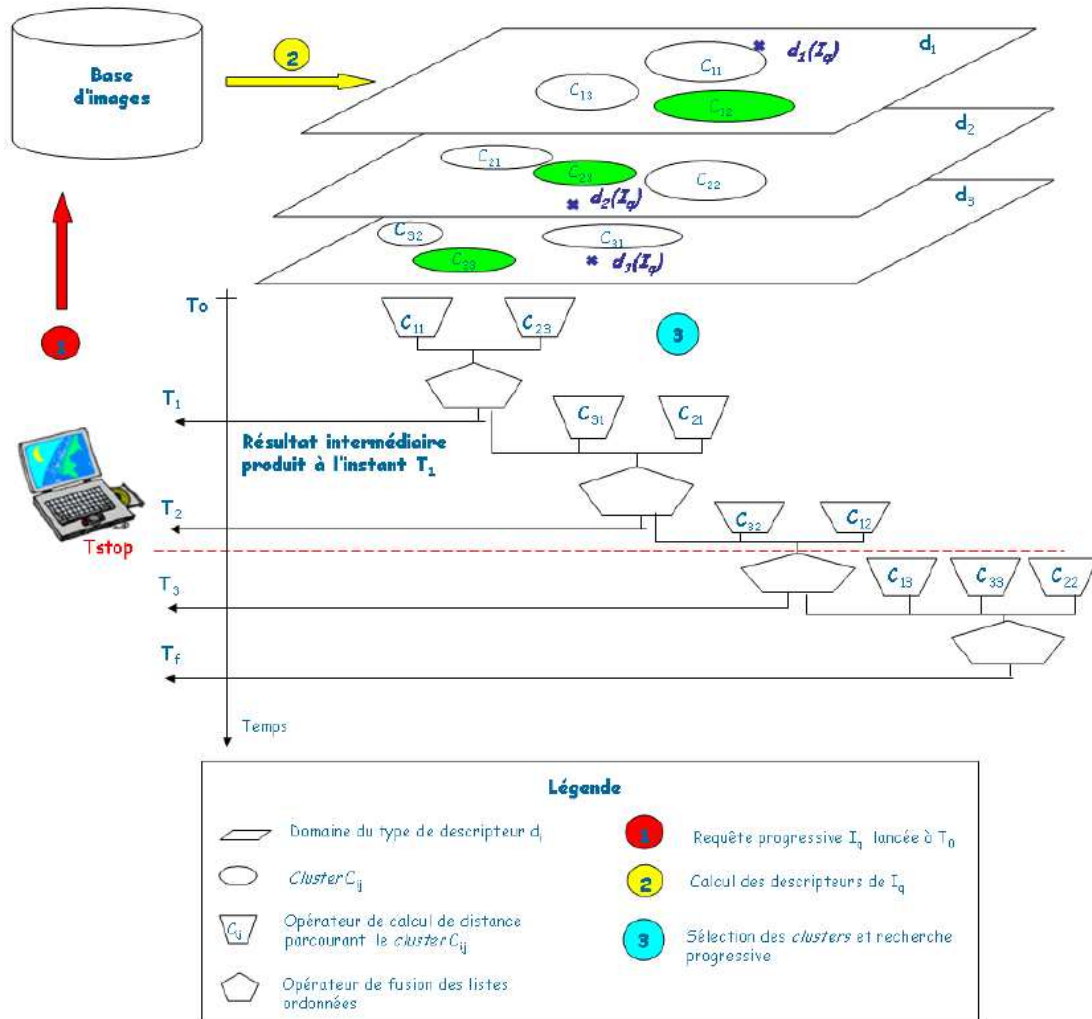


FIG. 3.3 – Exemple de Recherche progressive par le contenu.

Plus généralement, nous avons :

$$\vec{Q}(I_q, T_0, T_f) \iff RI(I_q, T_1), RI(I_q, T_2), \dots, RI(I_q, T_i), \vec{Q}(I_q, T_i, T_f) \quad (3.5)$$

où $RI(I_q, T_i)$, $T_i \in]T_0, T_f[$ est le résultat intermédiaire retourné à l'instant T_i .

Les sous-requêtes fournissent des résultats intermédiaires qui sont progressivement mis à jour et raffinés au cours du temps global d'exécution de la requête sur la totalité de la base. L'avantage est ici de fournir, sans attendre, des résultats intermédiaires à l'utilisateur au cours de l'exécution de la requête, qu'il pourra d'ailleurs stopper (T_{stop}). Après la fusion des résultats de recherche sur chaque *cluster*, les résultats intermédiaires de chaque sous-requête sont envoyés à T_1, T_2, T_3, \dots (voir figure 3.3). Lorsque toute la base a été parcourue, le dernier résultat intermédiaire est fusionné avec les résultats des derniers *clusters* parcourus et il est envoyé comme résultat final à l'utilisateur (à T_f).

Les résultats intermédiaires de la requête progressive $\vec{Q}(I_q, T_0, T_f)$ dépendent de l'ordre d'exécution des sous-requêtes et des instants intermédiaires de réponse du système de recherche. Par contre, le résultat final de $\vec{Q}(I_q, T_0, T_f)$ est le résultat obtenu soit lorsque l'intégralité de la base d'images a été parcourue pour répondre à la requête initiale (c'est-à-dire quand tous les *clusters* ont été parcourus), soit par l'arrêt volontaire de la requête par l'utilisateur.

L'objectif de la section suivante est de décrire les stratégies d'exécution d'une requête progressive par l'exemple, puis de préciser les principaux facteurs qui rentrent dans la définition d'un ordre d'exécution des sous-requêtes.

3.5.1 Stratégies d'exécution d'une requête progressive par l'exemple

Rappelons que $D = \{d_1, d_2, \dots, d_m\}$ est l'ensemble des types de descripteurs et qu'une base d'images B est organisée en *clusters* selon une méthode décrite en section 2.4.1 et suivant chaque type de descripteur.

Définition 3.7 (Plan de requête) *Un plan de requête P pour une requête par l'exemple I_q est une liste de *clusters* C_{ij} à parcourir pour retrouver les images les plus similaires selon des priorités définies pour le traitement de l'ensemble des types de descripteurs D de la base B .*

Les priorités de traitement des types de descripteurs peuvent être fixées en fonction du pouvoir discriminant de ceux-ci. Elles dépendent aussi de l'image-requête. On pourra commencer la recherche avec des descripteurs de couleur pour retrouver les images d'un ciel bleu. Les descripteurs de texture permettront ensuite d'affiner la recherche pour ne retenir que les images présentant des régions homogènes.

Un plan de requête se décompose en plusieurs sous-plan P_i tels que :

$$P(I_q, D) \leftarrow P_1(I_q, d_1) \circ P_2(I_q, d_2) \circ \dots \circ P_i(I_q, d_i) \circ \dots \circ P_m(I_q, d_m) \quad (3.6)$$

où $P_i(I_q, d_i)$ correspond à un sous-plan de traitement des *clusters* du type de descripteur d_i (c'est-à-dire à la recherche des plus proches voisins de I_q dans chaque *cluster* correspondant au type de descripteur d_i). \circ un opérateur de fusion ou d'agrégation des résultats.

Définition 3.8 (Opérateur de fusion) *Un opérateur de fusion construit un résultat unique à partir des résultats issus de plusieurs sous-plans de requête en entrée. Le résultat d'un sous-plan de requête étant une liste ordonnée, l'opérateur de fusion se réduit à une fusion de listes ordonnées si tous les sous-plans en entrée sont rattachés aux descripteurs de même type.*

$P(I_q, D)$ peut être réécrit au niveau des ensembles des *clusters* pour chaque type de descripteur comme suit :

$$P(I_q, D) \leftarrow P_1(I_q, \{C_{11}, \dots, C_{1n_c(d_1)}\}) \circ \dots \circ P_m(I_q, \{C_{m1}, \dots, C_{mn_c(d_m)}\})$$

avec $n_c(d_i)$ le nombre de *clusters* de type de descripteur d_i . Nous en déduisons l'écriture du sous-plan P_i à un niveau de granularité plus fin, c'est-à-dire au niveau des *clusters* de descripteurs :

$$\begin{aligned} P_i(I_q, d_i) &\leftarrow P_i(I_q, \{C_{i1}, \dots, C_{in_c(d_i)}\}) \\ &\leftarrow P_i(I_q, C_{i1}) \circ \dots \circ P_i(I_q, C_{in_c(d_i)}) \end{aligned}$$

où $P_i(I_q, C_{ik})$, $k = 1..n_c(d_i)$ correspond au parcours du *cluster* C_{ik} lors de la recherche par l'exemple avec la requête I_q . Il s'agit là d'une phase élémentaire de recherche.

Définition 3.9 (Phase élémentaire de recherche) *Une phase élémentaire de recherche est l'exécution d'une sous-requête minimale.*

L'écriture (3.6) montre qu'il existe, pour une même requête par l'exemple, plusieurs plans d'exécution possibles. En effet, avec $n_c(d_i)$ *clusters* de type de descripteur d_i et m types de descripteurs, on dénombre jusqu'à $(\sum_{i=1}^m n_c(d_i))!$ plans. Dans la figure 3.3 par exemple, on a au total 9 *clusters*, soit $9! = 362880$ plans possibles, pourtant difficile à énumérer de façon exhaustive sans retarder la durée de la recherche. D'autre part, il n'est pas nécessaire de parcourir tous les *clusters* de tous les types de descripteurs pour obtenir les plus proches voisins de la requête par l'exemple. Une exécution partielle du plan de requête (3.6) pourrait suffire à obtenir un résultat équivalent. Face à ce constat, nous proposons d'élaborer les critères de choix des meilleurs plans possibles d'exécution. Ces critères devront permettre de définir un ordre d'exécution des sous-requêtes minimales qui garantit l'obtention rapide des résultats intermédiaires et l'exécution partielle la plus courte de (3.6).

Dans les sous-sections suivantes, nous définissons la notion d'éligibilité de *cluster* et celle de règle de filtrage. Nous formalisons également l'utilisation progressive des règles d'association.

3.5.2 Éligibilité des *clusters* de descripteurs

Un *cluster* est un groupe de vecteurs représentant des descripteurs visuels. Ses caractéristiques peuvent être dépendantes ou non d'une requête par l'exemple. Elles définissent les paramètres du *cluster* tels que sa forme géométrique, sa taille, la distribution

statistique de ses vecteurs, et la distance de son centroïde à une image-requête. Désignons par $Dep_{(k)}(I_q, C_{ij})$, $k = 1..n_{dep}$ la famille des mesures des caractéristiques de C_{ij} dépendantes de la requête I_q et par $Indep_{(k)}(C_{ij})$, $k = 1..n_{indep}$ la famille des mesures des caractéristiques de C_{ij} indépendantes de la requête I_q . Les mesures pouvant être d'échelles très différentes, nous les normalisons pour les rendre comparables. La normalisation est définie par l'équation (3.7) :

$$NormX = \frac{X - E(X)}{\sigma(X)} \text{ avec } \begin{cases} X = Dep_{(k)}(I_q, C_{ij}) \text{ ou} \\ X = Indep_{(k)}(C_{ij}) \end{cases} \quad (3.7)$$

où $E(X)$ et $\sigma(X)$ sont respectivement estimés par la moyenne et l'écart-type de X pour le descripteur d_i .

Nous proposons la notion d'éligibilité d'un *cluster* C_{ij} comme une grandeur réelle qui résulte de la combinaison de toutes les mesures normalisées des caractéristiques de C_{ij} . L'éligibilité de C_{ij} , notée $eligibility(I_q, C_{ij})$ pour la requête par l'exemple I_q , se définit comme suit :

$$eligibility(I_q, C_{ij}) = \sum_{k=1}^{n_{dep}+n_{indep}} \alpha_k \left| NormX - ExtrNormX_{\substack{i=1..m \\ j=1..n_c(d_i)}} \right| \quad (3.8)$$

$$\text{avec } \sum_{k=1}^{n_{dep}+n_{indep}} \alpha_k = 1, \quad X = Dep_{(k)}(I_q, C_{ij}) \text{ ou } X = Indep_{(k)}(C_{ij})$$

$ExtrNormX_{\substack{i=1..m \\ j=1..n_c(d_i)}}$ est la valeur extrême (maximum ou minimum) de $NormX$ pour la caractéristique X en fonction du sens de l'optimisation. Si X est la distance du *cluster* à l'image-requête, $ExtrNormX$ sera la plus petite valeur possible de $NormX$, $i = 1..m$ et $j = 1..n_c(d_i)$.

D'après l'équation (3.8), plus $eligibility(I_q, C_{ij})$ est petit, plus le *cluster* C_{ij} est éligible. Une base d'images fixes pouvant être décrite par plusieurs types de descripteurs à la fois, l'éligibilité est donc définie pour induire un ordre sur l'ensemble de tous les *clusters* pour tous les types de descripteurs. L'utilisation progressive des types de descripteurs consiste donc à sélectionner des *clusters* suivant un ordre sur leur éligibilité. Une image-requête étant donnée, le parcours des *clusters* dans un tel ordre définit un plan d'exécution de la requête par l'exemple dont une phase élémentaire correspond à une recherche par similarité dans un *cluster*, peu importe le type de descripteur.

3.5.3 Utilisation des règles d'association dans un contexte progressif

Nous nous plaçons dans un contexte de recherche progressive par le contenu où des résultats intermédiaires sont retournés au fur et à mesure de la progression de la recherche et à la fréquence fixée par l'utilisateur. Nous désignons par phase observable de recherche l'étape de traitement dont la fin correspond à l'envoi d'un résultat intermédiaire à l'utilisateur. Une phase observable est donc une succession de phases élémentaires. La numérotation des phases observables commence à 1. Étant donnée une image-requête I_q , le système calcule, pour chaque type de descripteur d_i , la mesure

d'éligibilité des *clusters* (Voir section 3.5.2), puis détermine, à une phase donnée (élémentaire ou observable), le *cluster* dans lequel rechercher les plus proches voisins de I_q .

Notons $TopC(M, j)$ l'ensemble des M *clusters* de meilleure éligibilité sélectionnés à la phase observable j . Il s'agit ensuite d'exploiter les règles d'association entre les *clusters* de $TopC(M, j)$ et ceux des phases observables précédentes de façon à éliminer de $TopC(M, j)$ certains *clusters* et les types de descripteurs correspondants pour accélérer les premiers temps de la recherche. Soit R l'ensemble des règles calculées lors du processus d'organisation d'images (chapitre 2). Pour toute règle $r \in R$, notons C_r l'ensemble des *clusters* présents dans l'expression de la règle r (à la fois en partie droite et gauche). Notre hypothèse initiale est la suivante : pour les règles dont le support et la confiance sont supérieurs à un seuil donné, à une phase observable k de la recherche, une règle r est sélectionnée si :

$$C_r \subseteq \bigcup_{j=1..k} TopC(M, j) \quad (3.9)$$

Intuitivement, à l'issue du processus de calcul des *clusters* et d'extraction des règles décrit au chapitre 2, une règle de la forme (3.4) exprime, si elle est sélectionnée, que les plus proches voisins de l'image-requête I_q appartenant à tous les *clusters* de la partie gauche de la règle (3.4) sont probablement aussi présents dans la partie droite de la règle. Les expérimentations faites au chapitre 4 confirment bien cette hypothèse. Notre stratégie de sélection des types de descripteurs consiste donc, en priorité, à ignorer systématiquement pour la recherche toutes les parties droites des règles sélectionnées et donc, tous les *clusters* de descripteurs correspondants.

3.5.4 Règles de filtrage

Les algorithmes de recherche de plus proches voisins utilisent les propriétés géométriques des *clusters* issus de la classification automatique pour accroître les performances de la recherche. Le principe général consiste à définir des règles qui, d'une part, évitent le parcours séquentiel inutile des *clusters*, et d'autre part, arrêtent la recherche dès que tous les *clusters* restant à parcourir ne sont pas pertinents. Ces règles garantissent en général l'exactitude des résultats obtenus mais leur capacité à filtrer les *clusters* pertinents dépend de la structure de ceux-ci et de leurs dispositions relatives dans l'espace multidimensionnel.

Les règles de filtrage peuvent être adaptées aux particularités de certaines structures d'index comme les structures arborescentes. C'est le cas par exemple dans les travaux de Hjaltason et Samet [Hjaltason et Samet, 1995] où précisément un critère lié à la distance permet d'éviter le parcours en profondeur ou en largeur de l'arbre d'index. Rousopoulos *et al.* [Rousopoulos *et al.*, 1995] exploitent les propriétés des hyper-rectangles englobants pour renforcer l'efficacité des règles de filtrage sur les index de type *R-tree* [Guttman, 1984, Sellis *et al.*, 1987, Bechmann *et al.*, 1990].

Nous reprenons la règle classique qui consiste à éliminer systématiquement tous les *clusters* dont la distance minimum à l'image-requête est supérieure à un seuil calculé et

non nécessairement nul. Ce seuil tient compte de la distance entre l'image-requête et le plus proche voisin courant. Il dépend aussi des *clusters* dont la taille est au moins égale au nombre de plus proches voisins recherchés.

Plus formellement, pour un type de descripteur d_i , si R_{ij} est le rayon du *cluster* C_{ij} , il est égal à la distance entre le centroïde de C_{ij} et son point le plus éloigné. Nous définissons la distance minimum d_{min} entre une image-requête I_q et un *cluster* C_{ij} par :

$$d_{min}(I_q, C_{ij}) = \max(\text{Dist}(I_q, C_{ij}) - R_{ij}, 0)$$

où $\text{Dist}(I_q, C_{ij})$ est la distance entre le centroïde de C_{ij} et la requête I_q . La règle classique s'énonce donc comme suit :

$$\text{Rule}_1 : \text{Si } d_{min}(I_q, C_{ij}) \geq \min_{\substack{p=1..n_c(d_i) \\ \text{card}(C_{ip}) \geq k}} (\text{Dist}(I_q, C_{ip}) + R_{ip}, d(I_q, ppv_k)) \text{ alors éliminer } C_{ij} \quad (3.10)$$

$\text{card}(C_{ip})$ est le nombre de vecteurs du *cluster* C_{ip} ($p = 1..n_c(d_i)$) et k le nombre de plus proches voisins recherchés. $d(I_q, ppv_k)$ est la distance entre I_q et le $k^{\text{ième}}$ plus proche voisin courant.

La règle Rule_1 (inéquation 3.10) exploite tout simplement la forme des *clusters* et assure que les résultats retrouvés restent corrects par rapport à une recherche séquentielle. Il est cependant possible d'utiliser une règle plus forte que Rule_1 qui tienne compte de la sensibilité de la méthode de *clustering* aux valeurs aberrantes (*outliers*) pour éliminer de la recherche tous les *clusters* dont la distance minimum à l'image-requête est non nulle. On propose alors la deuxième règle suivante :

$$\text{Rule}_2 : \text{Si } d_{min}(I_q, C_{ij}) > 0 \text{ alors éliminer } C_{ij} \quad (3.11)$$

Rule_2 est particulièrement intéressante si la requête par l'exemple appartient à la base d'images. Dans ce cas, elle appartient à un *cluster* pour un type de descripteur donné et intuitivement, le *cluster* contient quelques voisins de la requête. D'après la définition de d_{min} , la règle Rule_2 (inéquation 3.11) indique donc la condition sous laquelle un *cluster* pourrait être considéré comme éloigné de la requête, et donc non pertinent pour la recherche.

Les règles étudiées dans cette section sont indépendantes et nous les avons utilisées séparément. Elles peuvent néanmoins être combinées pour améliorer, d'une part, l'efficacité de la recherche en terme de temps de réponse, et d'autre part, la précision des résultats. Il s'agit, par exemple, d'appliquer la règle (3.11) lors d'un premier parcours des *clusters*. Dans un second parcours, la règle (3.10) pourra ensuite être appliquée aux *clusters* filtrés par la règle (3.11). La règle (3.11) sera ainsi plus efficace que si elle avait été appliquée dès le premier parcours des *clusters* puisque la liste des plus proches voisins courants au début du second parcours sera déjà suffisamment proche du résultat attendu.

3.6 Fusion des résultats intermédiaires

Notre système de recherche d'images fixes par le contenu utilise progressivement les types de descripteurs disponibles. À une phase de recherche donnée, des résultats

intermédiaires sont produits et doivent être intégrés à ceux des phases précédentes puis présentés à l'utilisateur sous forme d'une liste ordonnée. La fusion des listes de résultats issues de la comparaison de descripteurs hétérogènes est un problème complexe [Fagin *et al.*, 2003, Berretti *et al.*, 2004]. Nous introduisons néanmoins une notion de traitement et de fusion des résultats intermédiaires permettant de renforcer les résultats produits par utilisation de chaque type de descripteur.

L'intuition est la suivante : si une image apparaît dans plusieurs listes et parmi les premières de chacune de ces listes, alors elle aura un score élevé dans la liste fusionnée. Cette intuition est à l'origine de la définition de notre mesure de score.

Supposons que la recherche se fait, à une phase donnée p , dans le *cluster* C_{i_0j} du type de descripteur d_{i_0} . Sans nuire à la généralité, notons $\{d_1, d_2, \dots, d_{m_1}\}$ le sous-ensemble de m_1 types descripteurs correspondant aux *clusters* déjà parcourus au terme de la phase de recherche p . Ces types de descripteurs sont sélectionnés parmi les m disponibles dans la base d'images ($m_1 \leq m$). Soit l_i les listes ordonnées d'images, résultats de la recherche suivant les descripteurs sélectionnés d_i , avec $1 \leq i \leq m_1$. À la phase de recherche p , la liste d'images retournées l_{i_0} est mise à jour après le parcours du *cluster* C_{i_0j} puis fusionnée avec les autres listes l_i ($1 \leq i \leq m_1, i \neq i_0$) pour obtenir le résultat final également sous forme d'une liste ordonnée. Nous définissons pour une image I un score de fusion par l'équation suivante :

$$S_f(I) = \frac{1}{2} (\sum_{i=1}^{m_1} \lambda_i (f_{d_i}(I) + S_{d_i}(I)))$$

$$\text{avec } S_{d_i}(I) \in [0, 1], \quad f_{d_i}(I) = \begin{cases} 1 & \text{si } I \in l_i \\ 0 & \text{sinon} \end{cases} \quad \text{et } \sum_{i=1}^{m_1} \lambda_i = 1 \quad (3.12)$$

$S_{d_i}(I)$ est le score de I pour le type de descripteur d_i . Il est proportionnel à la similarité entre I et l'image-requête I_q , la similarité entre deux images étant propre à chaque type de descripteur. Le principal avantage de la fonction définie par l'équation 3.12 est sa granularité permettant la fusion des listes intermédiaires après chaque parcours de *cluster*, et ceci sans tenir compte du type de descripteur.

3.7 Synthèse

Nous avons décrit dans ce chapitre notre approche de recherche progressive par le contenu. Cette approche suppose qu'une technique de *clustering* ait été préalablement appliquée à la base et que des règles d'association aient été découvertes sur les *clusters* calculés. Les règles d'association sont découvertes hors-ligne. En-ligne, elles exploitent les relations entre *clusters* pour éviter de parcourir certains *clusters* si ceux-ci sont jugés « peu intéressants » au sens de la relation qui les lie aux *clusters* déjà parcourus. Les règles de filtrage interviennent uniquement dans le processus de recherche en-ligne. Elles s'appuient essentiellement sur les propriétés géométriques des *clusters* pour éliminer de la recherche ceux qui ne sont pas susceptibles de contenir les plus proches voisins. La mesure d'éligibilité proposée dans ce chapitre est un moyen de combiner toutes les caractéristiques mesurables des *clusters*, qu'elles soient dépendantes ou non de l'image-requête. Les *clusters* peuvent ainsi être ordonnés et parcourus, puis les résultats retour-

nés progressivement à l'utilisateur. Une mesure de calcul des scores permet d'agrèger les résultats issus de la recherche sur plusieurs types de descripteurs. L'évaluation de toutes les notions présentées dans ce chapitre font l'objet du chapitre expérimental qui suit.

Chapitre 4

Évaluations expérimentales

Le but de ce chapitre est d'évaluer notre approche d'organisation et de recherche d'images fixes par le contenu proposée aux chapitres 2 et 3. Les évaluations concernent à la fois l'efficacité et la qualité des résultats retournés à l'utilisateur. Après avoir précisé l'environnement expérimental et décrit nos données, nous nous intéressons, dans un premier temps, à l'étude des paramètres d'indexation et, plus précisément, à la génération des règles d'association que nous confortons par les techniques de l'analyse des correspondances multiples. Nous évaluons ensuite les performances de la recherche sur des bases de plusieurs milliers d'images fixes caractérisées par des descripteurs globaux. Nous discutons enfin de l'intérêt de la combinaison des descripteurs pour la recherche d'images par le contenu.

4.1 Introduction

Nous présentons dans ce chapitre une étude expérimentale du système d'organisation et de recherche d'images fixes proposé. Cette étude vise principalement à évaluer la rapidité ainsi que la qualité de la recherche d'images par le contenu. Les temps de réponse de notre approche de recherche progressive sont comparés à ceux d'une approche séquentielle menée à l'aveugle. Celle-ci s'avère être plus performante que les méthodes d'indexation multidimensionnelle en grande dimension [Weber *et al.*, 1998, Beyer *et al.*, 1999]. Quant aux résultats intermédiaires de la recherche progressive, ils sont dans un premier temps comparés aux résultats d'une recherche séquentielle. Cet indicateur relatif de qualité, destiné à croître avec la progression de la recherche, permet de fixer une base pour mieux comparer les temps partiels au temps d'une recherche séquentielle exhaustive. Dans l'étude de l'intérêt de la combinaison des descripteurs globaux, nous revenons aux mesures classiques de rappel et de précision pour évaluer la qualité des résultats.

Ce chapitre s'articule autour de trois principaux points. Nous montrons d'abord l'intérêt des règles d'association d'une part comme moyen d'organisation des images et d'explicitation des relations entre groupes d'images, et d'autre part, comme stratégie d'exécution d'une requête par l'exemple pour réduire les temps de recherche. Nous étudions ensuite les performances de la recherche progressive. Dans ce cadre, une analyse

fine nous permet de montrer l'impact de l'ordonnement des *clusters* d'images sur les résultats partiels. Une étude comparative des règles de filtrage présentées au chapitre 3 est également proposée. Le dernier aspect abordé dans ce chapitre concerne la combinaison des descripteurs globaux dont nous montrons la valeur ajoutée dans une recherche par le contenu. Toutes les expérimentations sont faites sur des bases d'images réelles et la recherche effectuée est celle des 15 plus proches voisins.

4.2 Environnement expérimental

Les algorithmes de *clustering* et de découverte des règles d'association sont implémentés en C++. Nous réutilisons le modèle expérimental (*MPEG-7 XM Software*) développé à l'Institut des Systèmes Intégrés de Munich¹ pour la description *MPEG-7* de nos images. Tous les tests de notre méthode de recherche sont effectués sur PC sous Linux (Pentium 4, 3GHz, 1Go de mémoire centrale). Les temps d'exécution des algorithmes sont mesurés à l'aide de la fonction `getrusage()`.

4.3 Description des données

Le système que nous évaluons se compte parmi les systèmes de recherche par renforcement de critères ou de descripteurs visuels. Son objectif est, d'une part, de bien organiser les images pour en faciliter la recherche par l'exemple et, d'autre part, de combiner les descripteurs à disposition pour améliorer le rappel et la précision de la recherche. Pour évaluer le temps de recherche de notre système, nous travaillons principalement avec trois bases deux-à-deux disjointes (c'est-à-dire sans images communes) de 7727, 30411, et 110291 images fixes que nous notons respectivement B_1 , B_2 , et B_3 . Les images de ces trois bases sont fournies par une agence photographique et sont toutes décrites par cinq types de descripteurs *MPEG-7* dont deux de couleur (*Color Layout Descriptor*, *Scalable Color Descriptor*), deux de texture (*Homogeneous Texture Descriptor*, *Edge Histogram Descriptor*), et un de forme (*Region-based Shape Descriptor*). Le tableau 4.1 précise pour chaque type de descripteur sa notation abrégée, la dimension de ses vecteurs et le type de mesure de dissimilarité entre deux vecteurs. L'annexe A donne une description détaillée de ces types de descripteurs.

Type de descripteur	Notation abrégée	Dimension des vecteurs	Espace de similarité
<i>Color Layout</i>	CLD	12	Distance L_2
<i>Scalable Color</i>	SCD	32	Distance L_1
<i>Homogeneous Texture</i>	HTD	62	Distance L_1
<i>Edge Histogram</i>	EHD	80	Distance L_1
<i>Region-based Shape</i>	RSD	35	Distance L_1

TAB. 4.1 – Résumé des types de descripteurs utilisés.

¹http://www.lis.ei.tum.de/research/bv/topics/mmdb/e_mpeg7.html

Dans nos expérimentations sur l'intérêt de la combinaison des descripteurs globaux, nous utilisons les images provenant de la base MOVI² de 610 images réparties en 32 séquences. Chaque séquence représente la même scène (2 ou 3D) prise dans des conditions différentes (variation d'intensité de la source lumineuse, rotation, translation, occultation, *etc.*). Certaines séquences ont été générées en modifiant la composition de la scène ou en déplaçant la caméra autour de la scène selon des mouvements plus ou moins complexes. Une présentation plus détaillée de MOVI est donnée dans [Amsaleg et Gros, 2001]. Les images de MOVI sont rajoutées à celles de B_3 pour constituer une nouvelle base que nous notons B_4 . B_4 contient donc $110291 + 610 = 110901$ images également décrites par les 5 descripteurs *MPEG-7* de couleur (CLD et SCD), de texture (HTD et EHD), et de forme (RSD). Cette base nous a permis d'évaluer le taux de reconnaissance des descripteurs globaux utilisés et de leurs combinaisons à travers les mesures de rappel/précision.

4.4 Intérêt des règles d'association

Les expérimentations sur l'intérêt des règles d'association sont essentiellement menées sur les bases B_1 et B_2 pour lesquelles des *clusters* sont calculés avec l'algorithme *k-means*. Le problème du calcul automatique du nombre k de *clusters* reste entier [Nakache et Confais, 2005]. Dans ce travail, nous adoptons une approche expérimentale dans laquelle le nombre choisi assure que le système produise des règles dont le support est supérieur à un seuil donné. La détermination du nombre de *clusters* reste difficile en raison d'un compromis à trouver entre ce nombre et la qualité des *clusters* calculés. Une tentative de résolution du problème est présentée dans [Fernandez *et al.*, 2002], mais la complexité du regroupement en *clusters* est plus grande puisque la méthode teste plusieurs valeurs de k avant de choisir la meilleure selon un critère statistique défini (critère de variance). Dans cette section, nous nous intéressons plus précisément à la génération et à l'utilisation effective des règles améliorant la recherche par le contenu dans le but d'évaluer la stratégie de sélection définie au chapitre 3. Les règles sont découvertes avec l'algorithme *Apriori* [Agrawal *et al.*, 1993].

4.4.1 Génération des règles d'association : étude des paramètres

Rappelons qu'une règle d'association entre *clusters* est une implication de la forme (3.4) (voir page 59). Une image qui appartient à tous les *clusters* de la partie gauche de l'implication se trouve aussi dans le *cluster* de la partie droite avec les taux *supp* et *conf* respectifs de support et de confiance. Notre stratégie d'utilisation des règles consiste à ignorer le *cluster* de la partie droite des règles sélectionnées lors de la recherche.

Nous avons maintenu le seuil de la confiance à 50% pour le calcul des règles afin d'avoir des implications fortes. Les expérimentations sont faites dans un premier temps sur la base B_1 de 7727 images fixes. La figure 4.1 montre les variations du nombre de règles en fonction du nombre de *clusters* pour les trois valeurs suivantes du seuil du

²disponible à l'adresse : http://www.irisa.fr/texmex/base_images/index.html

support : 0, 1%, 1%, et 10%. Nous avons travaillé avec le même nombre de *clusters* pour tous les 5 types de descripteurs. L'expérimentation peut être étendue en considérant un nombre de *clusters* différent pour chaque type de descripteur. On constate que plus le nombre de *clusters* est élevé, moins on obtient de règles. La figure 4.2 confirme cet aspect. En particulier, le système ne produit pratiquement plus de règles à partir de 7 *clusters* pour chacun des types de descripteurs. On voit aussi, d'après la figure 4.2, que pour des valeurs petites du nombre de *clusters*, le système produit des règles pour toutes les valeurs spécifiées du seuil du support. Le nombre de *clusters* doit donc être judicieusement choisi pour assurer un bon compromis entre le nombre de règles produites et le seuil du support des règles.

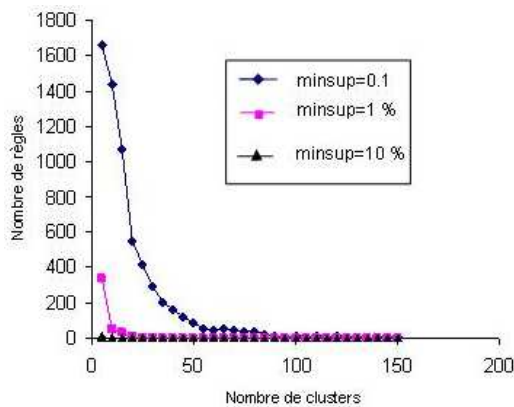


FIG. 4.1 – Variation du nombre de règles en fonction du nombre de *clusters* pour 3 valeurs du seuil du support sur B_1 .

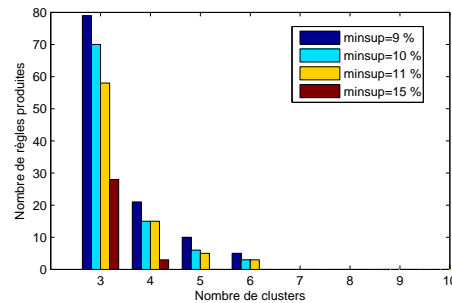


FIG. 4.2 – Diagramme à bâtons du nombre de règles en fonction du nombre de *clusters* pour 4 valeurs du seuil du support sur B_1 .

Dans la suite des expérimentations sur la base B_1 , nous avons fixé le nombre de *clusters* à 5 pour chaque type de descripteur et le seuil du support à 10%. Dans ces conditions, le système produit 6 règles dont le support varie entre 10% et 13% (voir figure 4.3). Ce support relativement faible s'explique. En effet, la valeur du support est une fonction décroissante du nombre de *clusters* choisi par type de descripteur. Si l'on suppose par exemple une répartition uniforme des images dans chacun des 5 *clusters* pour chacun des types de descripteurs, alors le support des règles est majoré par 20%.

(1, <i>CLD</i>)	→	(3, <i>RSD</i>)	(11% 69, 2%)
(2, <i>CLD</i>)	→	(2, <i>SCD</i>)	(12.3% 67, 4%)
(2, <i>SCD</i>)	→	(2, <i>CLD</i>)	(12.3% 55, 1%)
(2, <i>CLD</i>)	→	(0, <i>HTD</i>)	(10% 55, 1%)
(1, <i>SCD</i>)	→	(4, <i>CLD</i>)	(13% 52, 6%)
(1, <i>SCD</i>)	→	(4, <i>EHD</i>)	(12.8% 51, 5%)

FIG. 4.3 – Règles d'association découvertes avec *Apriori* sur la base B_1 .

Nous réalisons des tests sur la base B_2 de 30411 pour confirmer les observations faites sur la base B_1 . Le seuil de la confiance reste maintenu à 50% pour les même

raisons que celles évoquées avec B_1 . L'homogénéité sur la taille des *clusters* est un paramètre très important car les *clusters* de très grande taille ont tendance à apparaître en partie droite des règles de façon à maximiser la confiance. Pour éviter ce biais dans le calcul des règles, nous faisons des tests pour obtenir des *clusters* dont la différence de taille est la plus réduite possible. Ce faisant, nous calculons pour une valeur fixée du nombre de *clusters*, les variations relatives de la taille de tous les *clusters* par rapport au nombre moyen d'images par *cluster*. Ces variations sont exprimées en nombre d'images. La figure 4.4(a) présente la variation moyenne de la taille des *clusters* par rapport au nombre de *clusters* pour chacun des 5 types de descripteurs utilisés. Elle montre que pour des valeurs inférieures à 10 du nombre de *clusters*, on obtient des *clusters* très disproportionnés en taille. Le nombre de *cluster* doit être suffisamment grand pour que la taille ne s'écarte pas trop de la moyenne. Ce constat est vrai pour tous les types de descripteurs qui se comportent de la même manière dans la répartition des images de chaque *cluster*. Le nombre de *clusters* pour tous les types de descripteurs, qu'il soit le même ou pas, ne modifie pas notre principe d'organisation et de recherche d'images fixes.

Nous pouvons donc poser comme hypothèse de travail, sans nuire à la généralité, que le nombre de *clusters* est le même pour tous les types de descripteurs. Dans la suite de nos expérimentations, nous le fixons à 30, car d'après l'expérience reportée dans la figure 4.4(a), 30 est la valeur pour laquelle la variation de la taille des *clusters* se stabilise pour l'ensemble des types de descripteurs.

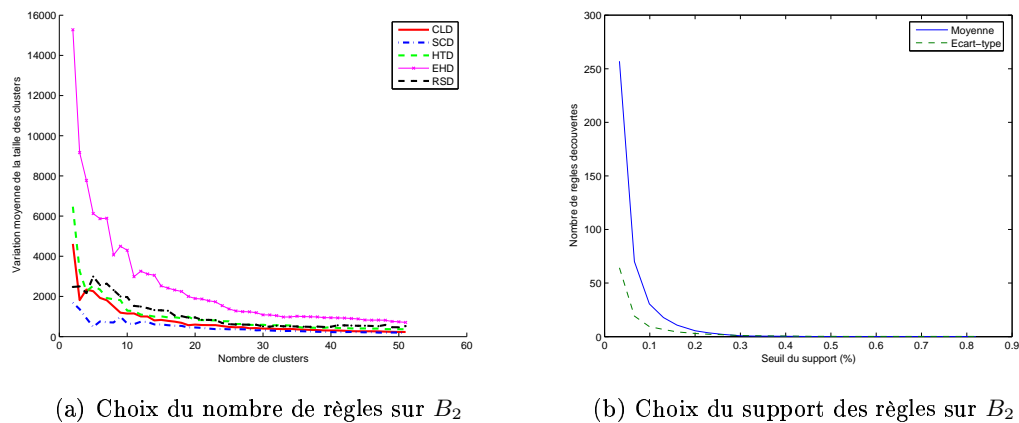


FIG. 4.4 – Choix du nombre et du support des règles sur B_2 .

Le seuil du support est aussi choisi expérimentalement. La figure 4.4(b) donne les variations du nombre de règles calculées en fonction du seuil du support. On obtient en général des règles de support très faible. Un nombre élevé de *clusters* entraîne la réduction du support des règles. Si l'on suppose par exemple que tous les 30 *clusters* contiennent exactement le même nombre d'images, c'est-à-dire $\frac{30411}{30}$, alors le support d'une règle sera au plus de $\frac{\frac{30411}{30} \times 100}{30411} \approx 3,33\%$ sous l'hypothèse que les parties gauche et

droite des règles contiennent les mêmes images. Les règles donnent néanmoins une idée du nombre d'images qui se retrouvent dans un même *cluster* pour plusieurs types de descripteurs. Nous réalisons des tests avec les 3 valeurs suivantes du seuil du support : 0,033, 0,066, et 0,1. Ces valeurs correspondent respectivement à au moins 10, 20, et 30 images communes pour les parties gauche et droite des règles. La confiance d'une règle retenue est bien entendue supérieure ou égale à 50%.

Le tableau 4.2 résume les paramètres déterminés expérimentalement ainsi que nos hypothèses de travail.

Paramètres déterminés expérimentalement	Hypothèses	Incidence des paramètres et des hypothèses
Nombre de <i>clusters</i>	Le même pour tous les types de descripteurs	Nombre de règles produites Calcul des règles Taille des <i>clusters</i>
Support des règles		Nombre de règles produites
Confiance des règles	Avoir des implications fortes	Nombre de règles produites

TAB. 4.2 – Paramètres déterminés expérimentalement.

Le tableau 4.3 récapitule les valeurs de tous les paramètres que nous avons pu déterminer expérimentalement et que nous utilisons dans la suite des expérimentations sur les bases B_1 et B_2 . Il indique aussi le nombre de règles découvertes pour chacune des deux bases.

Base d'images	Nombre de <i>clusters</i> par type de descripteur	Support min. des règles (%)	Confiance min. des règles (%)	Nombre de règles
B_1	5	10	50	6
B_2	30	0,033	50	722
B_2	30	0,066	50	235
B_2	30	0,1	50	127

TAB. 4.3 – Paramètres fixés et nombre de règles découvertes à l'organisation des images.

L'utilisation de *k-means* comme algorithme de calcul des *clusters* implique effectivement le choix préalable du nombre de *clusters*. Ce choix peut cependant être guidé par les règles d'association dont la découverte et la qualité dépendent de la taille des *clusters*.

4.4.2 Confirmation des règles par l'analyse des correspondances multiples

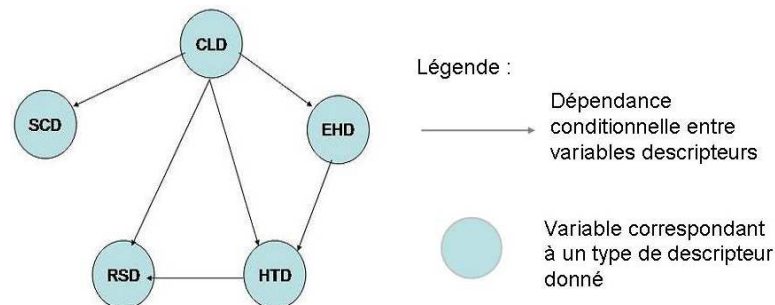
Pour compléter la recherche de règles, nous avons utilisé l'analyse des correspondances multiples (ACM) sur la base B_1 . Chacune des 7727 images de B_1 est décrite par cinq variables notées CLD, SCD, HTD, EHD, et RSD dont chacune correspond à un des 5 types de descripteurs utilisés pour décrire B_1 (voir tableau 4.1). Nous disposons

pour chaque variable et chaque individu (image) de la classe d'affectation de l'individu pour la variable en question. Ces valeurs vont de 1 à 5. Nous transformons ce tableau en tableau disjonctif complet. En effet, l'ACM est équivalente à l'analyse factorielle des correspondances simples sur un tableau obtenu en remplaçant dans le tableau d'origine chaque variable qualitative par l'ensemble des variables indicatrices des différentes modalités de cette variable. Nous travaillons donc sur un tableau de 7727 lignes et 25 variables indicatrices. Notre objectif est de comparer les résultats de l'ACM et ceux de la technique des règles d'association. Les résultats les plus intéressants sont présentés au tableau 4.4, les nombres entre parenthèses indiquent la confiance de la règle.

Association forte entre modalités de variables	Modalités induites
$(2, CLD) \wedge (2, SCD)$	$(0, HTD)$ (54, 4%)
$(1, CLD) \vee (3, CLD) \wedge (2, HTD)$	$(0, SCD)$ (59, 7%) \wedge / \vee $(4, EHD)$ (51, 3%)
$(0, CLD) \wedge (3, HTD)$	$(0, SCD)$ (47, 9%) \vee $(3, SCD)$ (45, 7%)
$(4, CLD) \wedge (1, SCD)$	$(4, EHD)$ (52, 6%)
$(3, CLD) \wedge (0, SCD)$	$(2, EHD)$ (52, 2%) \wedge / \vee $(3, RSD)$ (54, 4%)

TAB. 4.4 – Règles produites par l'ACM.

En plus des règles déterminées par la technique des règles d'association (figure 4.3), on retrouve des règles plus complexes mettant en jeu plusieurs variables. On constate que le descripteur de couleur CLD permet à lui seul d'induire d'autres valeurs de descripteurs. Nous n'avons conservé que les valeurs de confiance supérieures à 50% sauf pour la règle qui conduit à induire deux valeurs des modalités de SCD. Cette permanence d'associations impliquant CLD nous a conduit à estimer la topologie d'un réseau bayésien entre les cinq variables (voir figure 4.5). On note la position à l'origine du réseau de CLD. Il existe des dépendances conditionnelles entre les variables SCD, HTD, EHD, RSD et CLD.

FIG. 4.5 – Estimation topologique des relations entre les types de descripteurs sur B_1 .

Les méthodes statistiques confirment et complètent les règles d'association. Pour

choisir le nombre de classes par type de descripteur, un examen plus fin de chacun d'eux est envisageable. Cependant, un des problèmes est l'absence de sémantique et de similarité visuelle fournies par la classification.

4.4.3 Utilisation effective des règles pour la recherche par le contenu

Dans cette sous-section, nous évaluons notre principe d'utilisation des règles d'association sur les bases B_1 et B_2 . L'aspect progressif est explicitement pris en compte au cours de l'évaluation sur la base B_2 pour laquelle les *clusters* sont ordonnés en fonction de la distance de leur centre à une image-requête donnée. On rappelle que les requêtes progressives consistent à agréger régulièrement des résultats intermédiaires et à les retourner à l'utilisateur au fur et à mesure de la progression de la recherche.

Test sur la base B_1 de 7727 images. Le but de cette expérimentation est de tester le critère de sélection des règles formulé au chapitre 3. Le système a été conçu pour interroger par 500 images-requêtes (toutes déjà présentes) la base de 7727 images. Pour 165 d'entre elles, le système fait usage des règles d'association calculées, soit un taux d'utilisation de 33%. Le diagramme d'utilisation des règles est présenté à la figure 4.6. On constate que le système utilise rarement plus de trois règles dans le traitement d'une requête. D'autre part, rappelons qu'il n'y a que 6 règles qui ont été découvertes avec les paramètres que nous nous sommes fixés (figure 4.3). Ces règles ne font intervenir que 8 *clusters* sur l'ensemble des 25 *clusters* calculés (5 *clusters* pour chaque type de descripteur). Le système n'utilise pas de règle sur les images-requêtes pour lesquelles aucun *cluster* sélectionné n'est parmi les 8 de la figure 4.3. Il est donc normal que le taux d'utilisation des règles ne soit pas élevé.

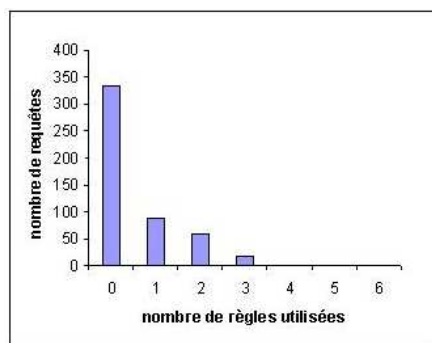
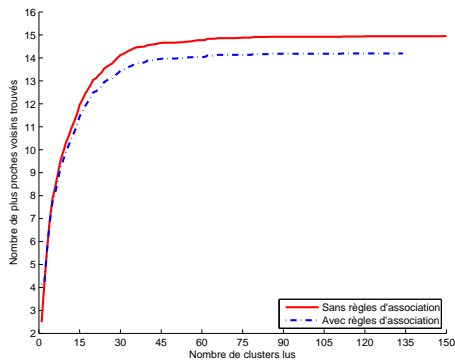


FIG. 4.6 – Diagramme d'utilisation des règles sur la base de 7727 images (base B_1).

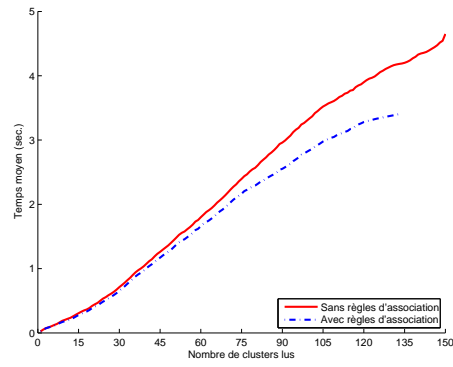
Tests sur la base B_2 de 30411 images. Dans cette série de tests, nous nous intéressons à la fois à la qualité des résultats d'une requête et au temps de recherche. Le but est de quantifier expérimentalement le gain de temps obtenu par l'utilisation des règles d'association. Nous n'avons pas de détails sur la provenance des images pour envisager

un plan robuste d'échantillonnage. Les images sont « du tout venant ». Pour cette raison, nous choisissons aléatoirement 100 images-requêtes non nécessairement présentes dans la base B_2 de test et nous recherchons pour chacune d'elles les 15 plus proches voisins, puis nous mesurons la qualité et le temps moyens de la recherche. Tous les résultats présentés dans cette étude sont comparés à ceux obtenus par une recherche séquentielle menée à l'aveugle. A l'heure actuelle, il est très difficile d'évaluer, « dans l'absolu », la qualité des résultats d'une recherche par l'exemple sur des images, car elle dépend en grande partie d'objectifs implicites et d'applications subjectives de l'utilisateur dont l'image-requête n'est pas toujours porteuse de façon auto-suffisante. C'est pourquoi, d'abord motivés par les problèmes de performance, nous définissons la notion de qualité de la recherche, par rapport à ce que l'on obtient par une recherche séquentielle, en comptant le nombre d'images communes entre chaque résultat intermédiaires obtenu par notre approche et le résultat final obtenu au terme d'une recherche séquentielle exhaustive sur toute la base d'images et suivant tous les types de descripteurs.

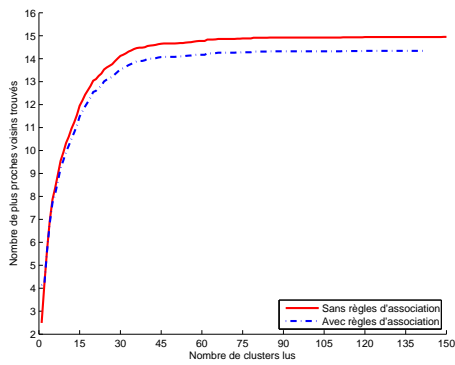
Les tests sont réalisés pour les seuils suivants du support : 0,033, 0,066, et 0,1. Le tableau 4.3 donne pour chacun des seuils choisis le nombre correspondant de règles découvertes. L'utilisation des règles est dynamique en ce sens qu'elle dépend des *clusters* déjà parcourus. Le traitement de chaque *cluster* requiert le parcours de tout l'ensemble des règles d'association pour déterminer si le *cluster* doit être parcouru ou pas. Les figures 4.7 et 4.8 indiquent les variations de la qualité relative des résultats et du temps de recherche en fonction du nombre de *clusters* parcourus pour chacune des 3 valeurs du seuil du support des règles d'association. Les graphiques 4.7(a), 4.7(b), et 4.7(c) montrent une perte de qualité relative de la recherche évaluée en moyenne à une image sur quinze, soit une perte d'environ 6,67%. Par contre, on constate que le gain en temps dépend du seuil du support choisi, et donc de la qualité et du nombre des règles d'association. Lorsque le seuil du support est fixé à 0,033%, on a un gain en temps de $\frac{(4,60-3,30) \times 100}{4,60} \approx 28,26\%$ (voir figure 4.8(a)). Le gain en temps est de $\frac{(4,60-3,90) \times 100}{4,60} \approx 15,22\%$ lorsque le seuil du support est égal à 0,1% (voir figure 4.8(c)). Cette différence est essentiellement due au nombre de règles découvertes. En effet, plus ce nombre est grand, plus la sélection des règles à un moment donné de la recherche est probable, entraînant ainsi une réduction du nombre de *clusters* à parcourir et donc un gain en temps plus important. D'autre part, plus on parcourt de *clusters*, plus on accroît la possibilité de sélectionner des règles, d'où l'écart important entre la courbe en trait fort (parcours séquentiel) et la courbe en pointillés (parcours mené selon les règles d'association) lorsque le nombre de *clusters* lus est important (figure 4.8). La perte de la qualité relative due à l'utilisation des règles d'association n'est pas aussi dépendante du seuil du support que le gain en temps de recherche (figure 4.7). Le phénomène peut s'expliquer par le fait qu'après avoir parcouru les *clusters* dans lesquels les plus proches voisins de l'image-requête se trouvent, l'élimination des autres *clusters* (présents en partie droite) par l'utilisation des règles d'association n'a qu'une très faible incidence sur la qualité du résultat final. Ainsi, pour utiliser efficacement les règles d'association, il est nécessaire de proposer en premier lieu un bon ordre de parcours des *clusters*. La proposition d'un tel ordre fait partie des stratégies de recherche dont l'étude est abordée à la section 4.5.



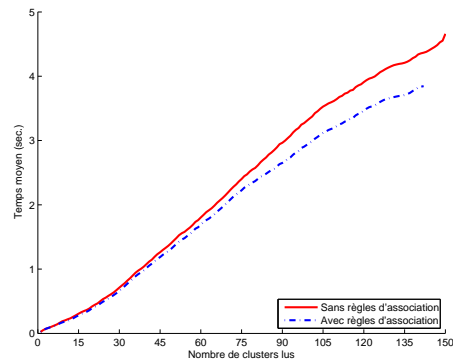
(a) support = 0,033%



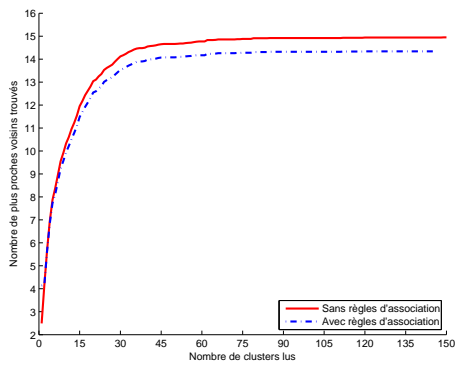
(a) support = 0,033%



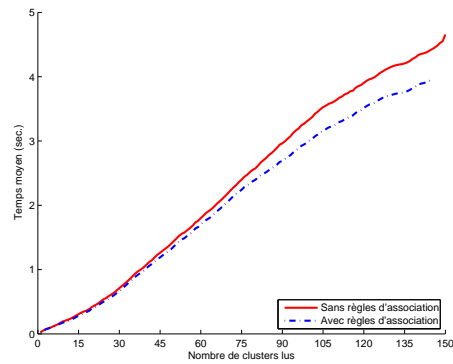
(b) support = 0,066%



(b) support = 0,066%



(c) support = 0,1%



(c) support = 0,1%

FIG. 4.7 – Variation de la qualité relative des résultats en fonction du nombre de *clusters* parcourus sur B_2 .

FIG. 4.8 – Variation du temps de recherche en fonction du nombre de *clusters* parcourus sur B_2 .

4.5 Performances de la recherche progressive

Nous nous proposons dans cette section d'évaluer les performances de la recherche progressive par le contenu, c'est-à-dire du parcours ordonné des *clusters* suivi de l'agrégation régulière des résultats intermédiaires. Pour cela, nous observons à travers la mesure d'éligibilité proposée à la section 3.5.2, l'influence des principaux critères de sélection des *clusters* sur les résultats et le temps de la recherche. Nous étudions ensuite, et de façon comparative, l'efficacité des deux règles de filtrage définies par les inéquations (3.10) et (3.11) de la section 3.5.4. Nous montrons, dans le cadre du passage à l'échelle de notre méthode, que l'utilisation de ces règles conduit à des performances meilleures qu'une recherche séquentielle exhaustive lorsque la taille de la base d'images varie. Les évaluations sont essentiellement menées sur la base B_3 de 110291 images. Les deux méthodes *k-means* et *M-Tree* de calcul des *clusters* sont utilisées. Dans tous les tests, le protocole d'interrogation de la base B_3 est le même que celui utilisé pour la base B_2 (voir section 4.4.3) avec 100 images requêtes n'appartenant pas nécessairement à B_3 , la recherche des 15 plus proches voisins est évaluée sur la qualité et temps de recherche comparés à ceux d'une recherche séquentielle exhaustive. Les graphiques sont tracés sur la base de la moyenne des valeurs obtenues sur les 100 images requêtes choisies.

4.5.1 Critères de sélection des *clusters*

Nous faisons une étude expérimentale de l'équation (3.8) proposée à la section 3.5.2 pour chacune des 3 valeurs suivantes du nombre de *clusters* de chaque type de descripteur : 10, 50, et 100. Les *clusters* sont calculés avec *k-means*. Les caractéristiques des *clusters* dépendantes de l'image-requête sont calculées au moment de la recherche. Nous choisissons d'évaluer pour chaque *cluster* C_{ij} la distance de son centre à l'image-requête I_q soumise. Cette distance sera notée : $Dist(I_q, C_{ij})$. Le principe de calcul de la distance dépend du type de descripteur. Dans le cas des descripteurs *MPEG-7* utilisés dans nos expérimentations, la distance est en général une variante des distances L_1 ou L_2 avec pondération des composantes des vecteurs (voir tableau 4.1).

Certaines caractéristiques de *clusters* indépendantes d'une image-requête, mais rentrant en jeu dans l'amélioration des performances de la recherche, peuvent aussi être évaluées. L'évaluation se fait, dans ce cas, au moment de l'organisation des images. D'abord focalisés sur l'optimisation du temps de recherche, nous avons choisi de calculer le coût de parcours d'un *cluster* donné au cours d'une recherche. Nous estimons ce coût proportionnel à la taille du *cluster* et à la dimension du vecteur correspondant au type de descripteur visuel pour lequel le *cluster* est calculé. Notons $dim(d_i)$ la dimension d'un vecteur du type de descripteur d_i et $card(C_{ij})$ le nombre de vecteurs du *cluster* C_{ij} . Le coût de parcours $Cost(C_{ij})$ du *cluster* C_{ij} est défini par : $Cost(C_{ij}) = dim(d_i) \times card(C_{ij})$. L'équation (3.7) de la section 3.5.2 devient donc :

$$NormX = \frac{X - E(X)}{\sigma(X)} \text{ avec } \begin{cases} X = Dist(I_q, C_{ij}) \text{ ou} \\ X = Cost(C_{ij}) \end{cases} \quad (4.1)$$

et l'équation (3.8) devient :

$$\begin{aligned} eligibility(I_q, C_{ij}) = & \alpha_1 \left| NormDist(I_q, C_{ij}) - \min_{j=1..n_c(d_i)} NormDist(I_q, C_{ij})_{i=1..m} \right| \\ & + \alpha_2 \left| NormCost(C_{ij}) - \min_{j=1..n_c(d_i)} NormCost(C_{ij})_{i=1..m} \right| \end{aligned} \quad (4.2)$$

avec $\alpha_1 + \alpha_2 = 1$

Les graphiques des figures 4.9 et 4.10 représentent, pour chaque nombre de *clusters* choisi, les variations de la qualité et du temps en fonction du nombre de *clusters* lus (phases de recherche). Dans la légende, les nombres entre parenthèses sont dans l'ordre les valeurs de α_1 et de α_2 . A titre d'exemple, (1,0) signifie que seule la distance du centre d'un *cluster* à une image-requête est utilisée comme caractéristique.

Les graphiques 4.9(a), 4.9(b), et 4.9(c) montrent que le système de recherche se comporte de la même manière pour les petits et les grands nombres de *clusters* en terme de qualité des résultats. La distance du centre des *clusters* à une image-requête, seule caractéristique utilisée et dépendante de la requête permet de se rapprocher plus rapidement du résultat final séquentiel. D'après les graphiques 4.10(a), 4.10(b), et 4.10(c), les trois courbes de variations du temps partiel (temps écoulé entre le début de la recherche et l'instant où un résultat intermédiaire est renvoyé) en fonction du nombre de *clusters* lus tendent à se rapprocher d'une courbe linéaire lorsque le nombre de *clusters* augmente. Une explication à ce phénomène est que la taille des *clusters* tend vers la moyenne lorsque le nombre de *clusters* croît. Ainsi, toutes les mesures induites par l'équation (4.2) en fonction des paramètres α_1 et α_2 produisent des variations similaires du temps partiel de réponse lorsque le nombre de *clusters* est grand. Dans cette situation, il suffira donc en pratique de choisir α_1 et α_2 de manière à obtenir une progression rapide de la qualité des résultats en fonction du nombre de *clusters* lus. Avec 100 *clusters* par type de descripteur, on peut par exemple choisir $\alpha_1 = 1$ et $\alpha_2 = 0$ (voir figure 4.9).

Les tableaux récapitulatifs 4.5, 4.6, et 4.7 indiquent pour chaque type de descripteur, le nombre de *clusters* parcourus pour avoir un résultat identique à celui d'une recherche séquentielle exhaustive. Ils montrent qu'il n'est pas nécessaire de parcourir tous les *clusters*. D'après le tableau 4.7, pour 100 *clusters* par type de descripteur, il suffit de parcourir 121 *clusters* sur un total de 500 lorsque $\alpha_1 = 1$ et $\alpha_2 = 0$. Ce parcours se fait en moyenne en 3,75 secondes au lieu de 15 secondes, temps moyen d'une recherche séquentielle exhaustive (figure 4.10(c)).

4.5.2 Gestion des résultats intermédiaires

Nous avons étudié comment l'ordre des *clusters* affecte le temps et la qualité des résultats partiels. La suite logique de l'étude est de se demander, un ordre de *clusters* étant choisi, si l'on peut avoir le résultat séquentiel final sans nécessairement parcourir tous les *clusters* disponibles. D'autre part, avec 100 *clusters* par type de descripteur et pour 5 types de descripteurs, le système effectue 500 phases de recherche (correspondant à 500 parcours de *clusters*). Il serait ennuyeux pour l'utilisateur de se voir proposer un résultat pas forcément différent à chaque phase de recherche. Pour répondre à ces

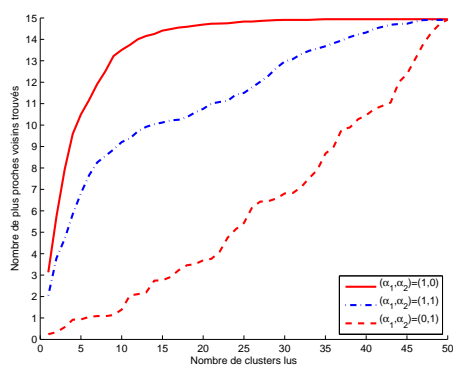
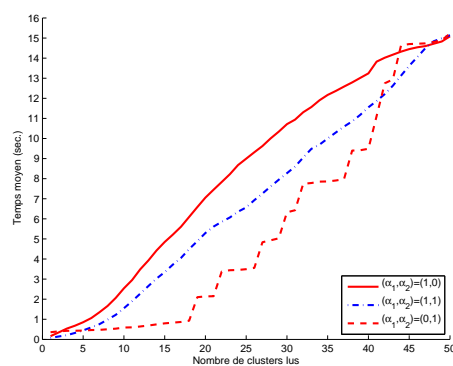
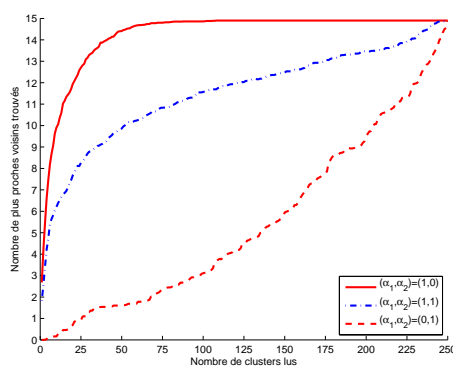
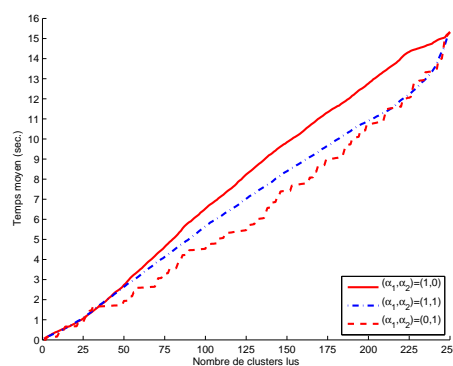
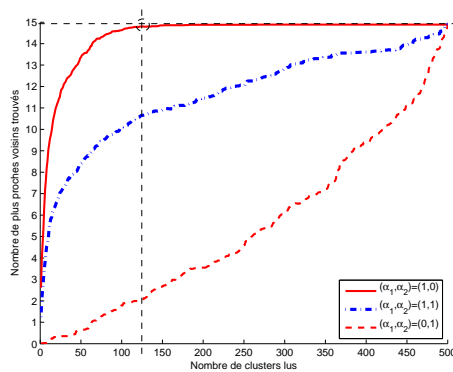
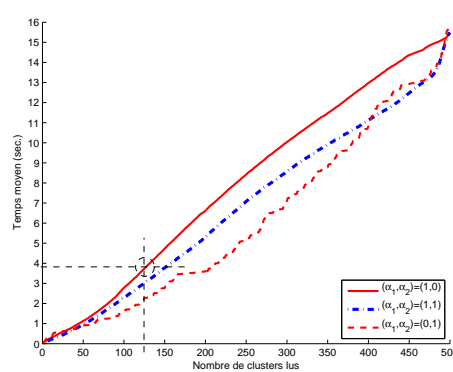
(a) nombre de *clusters* = 10(a) nombre de *clusters* = 10(b) nombre de *clusters* = 50(b) nombre de *clusters* = 50(c) nombre de *clusters* = 100(c) nombre de *clusters* = 100

FIG. 4.9 – Étude de la qualité des résultats intermédiaires en fonction de l'éligibilité des *clusters* sur B_3 .

FIG. 4.10 – Étude des temps intermédiaires de recherche en fonction de l'éligibilité des *clusters* sur B_3 .

Valeurs de α_1 et de α_2	Type de descripteur					Nombre total de <i>clusters</i> parcourus
	CLD	SCD	HTD	EHD	RSD	
$(\alpha_1, \alpha_2) = (1, 0)$	4	3	4	4	4	19
$(\alpha_1, \alpha_2) = (1, 1)$	7	7	8	8	8	38
$(\alpha_1, \alpha_2) = (0, 1)$	10	10	10	10	10	50

TAB. 4.5 – Nombre moyen de *clusters* parcourus pour avoir le résultat séquentiel final : cas de 10 *clusters* par type de descripteur (base B_3).

Valeurs de α_1 et de α_2	Type de descripteur					Nombre total de <i>clusters</i> parcourus
	CLD	SCD	HTD	EHD	RSD	
$(\alpha_1, \alpha_2) = (1, 0)$	13	14	10	14	12	63
$(\alpha_1, \alpha_2) = (1, 1)$	38	39	39	39	38	193
$(\alpha_1, \alpha_2) = (0, 1)$	48	49	48	48	48	241

TAB. 4.6 – Nombre moyen de *clusters* parcourus pour avoir le résultat séquentiel final : cas de 50 *clusters* par type de descripteur (base B_3).

Valeurs de α_1 et de α_2	Type de descripteur					Nombre total de <i>clusters</i> parcourus
	CLD	SCD	HTD	EHD	RSD	
$(\alpha_1, \alpha_2) = (1, 0)$	25	26	20	26	24	121
$(\alpha_1, \alpha_2) = (1, 1)$	74	75	76	76	75	376
$(\alpha_1, \alpha_2) = (0, 1)$	95	99	96	97	96	483

TAB. 4.7 – Nombre moyen de *clusters* parcourus pour avoir le résultat séquentiel final : cas de 100 *clusters* par type de descripteur (base B_3).

interrogations, nous déterminons à chaque phase de recherche le nombre de plus proches voisins de la liste finale retrouvés dans l'ordre.

Les graphiques 4.11(a), 4.11(b), et 4.11(c) présentent les diagrammes à barres qui indiquent, un nombre de plus proches voisins étant donné en abscisse, le pourcentage de requêtes pour lequel les plus proches voisins sont retrouvés en au plus 100 ($[0, 100[$), 200 ($[100, 200[$), 300 ($[200, 300[$), 400 ($[300, 400[$) ou 500 ($[400, 500[$) phases de recherche. Nous n'étudions ici que le cas de 100 *clusters* par descripteur, les autres cas pouvant être traités de façon analogue.

D'après le graphique 4.11(a), lorsque la distance des *clusters* à la requête est utilisée comme seule caractéristique, le résultat séquentiel final est obtenu en moins de 100 étapes pour près de 40% de requêtes. Si l'on se reporte au graphique 4.10(c), ce résultat séquentiel est donc obtenu en moins de 3,5 secondes pour ces 40% de requêtes, ce qui représente un gain en temps de plus de 77% par rapport à un parcours séquentiel exhaustif de la base d'images (3,5 secondes au lieu de 15). Le graphique 4.11(b) montre qu'après 400 *clusters* lus, un résultat identique à celui d'une recherche séquentielle exhaustive est retourné pour plus de 60% d'images-requêtes. Par contre, d'après la figure 4.11(c), un résultat identique à celui d'une recherche séquentielle exhaustive n'est en général retourné qu'après le parcours de plus de 400 *clusters* (pour plus de 60% d'images-requêtes).

Le coût du parcours des *clusters*, ainsi que sa combinaison avec la distance des *clusters* à une image-requête, réduisent les temps partiels mais augmentent le nombre de *clusters* à parcourir pour retrouver le résultat séquentiel. La distance d'une image-requête au centre d'un *cluster* est la caractéristique la plus intéressante étudiée dans ce travail si l'on s'intéresse en priorité à la qualité des résultats intermédiaires retournés. Nous retenons cette caractéristique pour l'évaluation des règles de filtrage à la sous-section 4.5.3.

4.5.3 Évaluation des règles de filtrage selon deux méthodes de construction des *clusters*

Utilisation du *k-means*. L'efficacité des règles de filtrage définies par les inéquations (3.10) et (3.11) de la section 3.5.4 a été comparée avec successivement 10, 100, et 200 *clusters* par type de descripteur. Nous avons ensuite tracé les variations de la qualité relative et du temps partiel correspondant, en fonction du nombre de *clusters* lus pour chacun des 3 cas (figure 4.12 et 4.13). La figure 4.12 confirme que la règle (*Rule*₁) formulée par l'inéquation (3.10) assure l'exactitude des résultats retrouvés. Nos expérimentations montrent en outre que l'approximation introduite par la formulation de la règle (*Rule*₂) à l'inéquation (3.11) est accrue avec le nombre de *clusters* par type de descripteur. La conclusion immédiate est que plus le nombre de *clusters* par type de descripteur est élevé, plus les *clusters* ont un rayon faible, ce qui réduit les risques de chevauchement et entraîne un taux de filtrage élevé avec (*Rule*₂) ainsi qu'une incidence sur la qualité des résultats. S'agissant de l'impact des règles de filtrage sur le temps de recherche, nous constatons d'après les expérimentations que le temps de recherche est de plus en plus faible lorsque le nombre de *clusters* par type de descripteur croît (figures

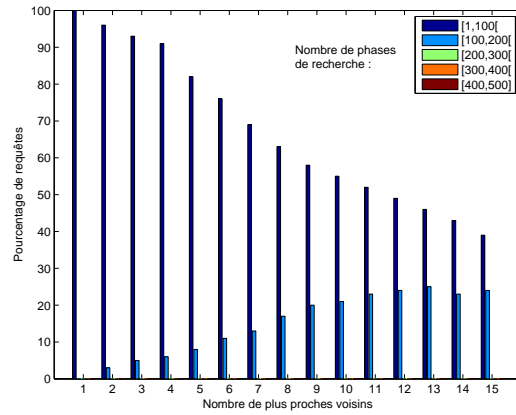
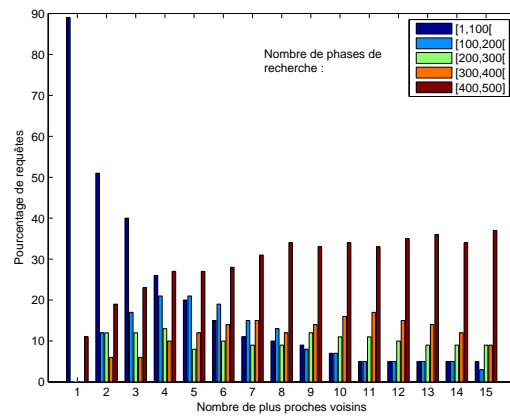
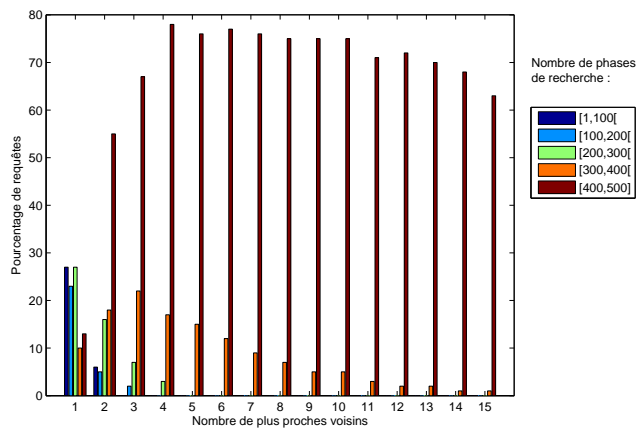
(a) $\alpha_1 = 1, \alpha_2 = 0$ (b) $\alpha_1 = 1, \alpha_2 = 1$ (c) $\alpha_1 = 0, \alpha_2 = 1$

FIG. 4.11 – Étude des résultats intermédiaires en fonction de l'éligibilité des *clusters* pour 100 *clusters* par type de descripteur sur B_3 .

4.13). Les règles ($Rule_1$) et ($Rule_2$) sont donc très performantes pour des petits rayons de *clusters*.

Une simple lecture des figures 4.12 et 4.13 montre par exemple que pour le cas de 200 *clusters* par type de descripteur, une recherche séquentielle exhaustive retourne les 15 plus proches voisins en 16,8 secondes en moyenne. L'utilisation de la règle ($Rule_1$) permet d'avoir les mêmes 15 plus proches voisins en 6,1 secondes en moyenne, soit 2,76 fois plus vite qu'une recherche séquentielle exhaustive. Malgré l'imprécision introduite par la règle ($Rule_2$), celle-ci permet d'avoir jusqu'à 14 plus proches voisins sur 15 en seulement 2,4 secondes en moyenne, soit 7 fois plus vite qu'une recherche séquentielle exhaustive. Nous concluons donc que la règle ($Rule_2$) est plus performante que la règle ($Rule_1$) sur le temps de réponse d'après les figures 4.13.

Nous avons évalué l'impact des règles de filtrage sur le parcours des *clusters* pour chaque type de descripteur. Les tableaux 4.8 et 4.9 présentent pour chaque type de descripteur, le nombre moyen de *clusters* parcourus au terme d'une recherche des 15 plus proches voisins. Chaque ligne correspond à un des trois cas étudiés (10, 100, et 200 *clusters*). Comme on pourrait s'y attendre, beaucoup moins de *clusters* sont parcourus avec ($Rule_2$) qu'avec ($Rule_1$) (voir aussi figures 4.12). De plus, la règle ($Rule_1$) est particulièrement inefficace sur les *clusters* de type EHD et RSD. Les règles sont sensibles au type de descripteur sur lequel elles sont appliquées. Le nombre moyen de *clusters* parcourus avec ($Rule_2$) donne une indication sur le taux de chevauchement des *clusters*. D'après le tableau 4.9, les *clusters* de type EHD et RSD sont ceux qui chevauchent le plus, d'où l'inefficacité constatée de la règle ($Rule_1$) sur ces types de descripteurs. La règle ($Rule_2$) peut être utilisée au début de la recherche progressive pour fournir plus rapidement à l'utilisateur un premier résultat proche du résultat attendu. La règle ($Rule_1$) pourra ensuite être utilisée pour préciser davantage la recherche.

Nombre de <i>clusters</i>	Type de descripteur				
	CLD	SCD	HTD	EHD	RSD
10 × 5	10	10	7	10	10
100 × 5	53	62	44	78	94
200 × 5	83	103	72	144	190

TAB. 4.8 – Nombre de *clusters* parcourus par type de descripteur pour la règle ($Rule_1$) sur B_3 .

Nombre de <i>clusters</i>	Type de descripteur				
	CLD	SCD	HTD	EHD	RSD
10 × 5	8	9	6	8	10
100 × 5	21	26	23	35	78
200 × 5	23	32	27	46	141

TAB. 4.9 – Nombre de *clusters* parcourus par type de descripteur pour la règle ($Rule_2$) sur B_3 .

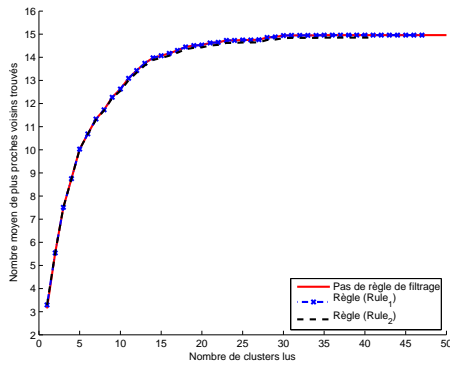
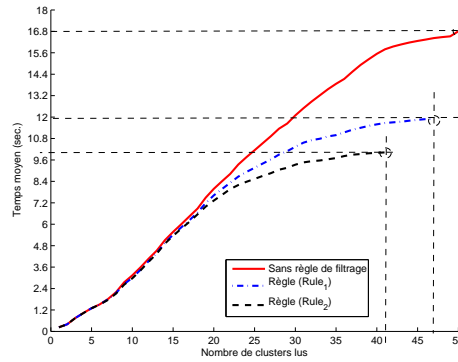
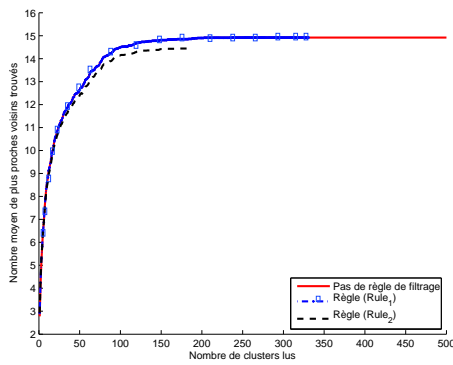
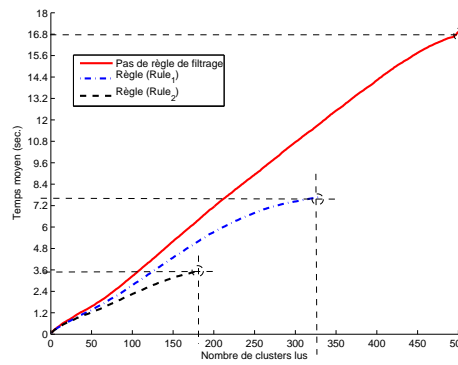
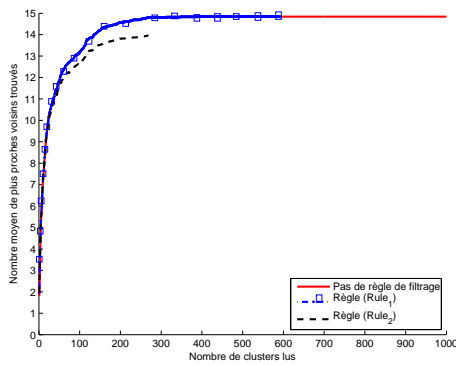
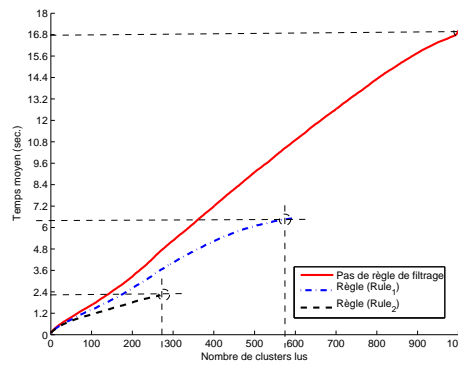
(a) nombre de *clusters* = 10(a) nombre de *clusters* = 10(b) nombre de *clusters* = 100(b) nombre de *clusters* = 100(c) nombre de *clusters* = 200(c) nombre de *clusters* = 200

FIG. 4.12 – Comparaison des règles de filtrage sur la base du nombre moyen de plus proches voisins trouvés (base B_3).

FIG. 4.13 – Comparaison des règles de filtrage sur la base du temps moyen de recherche (base B_3).

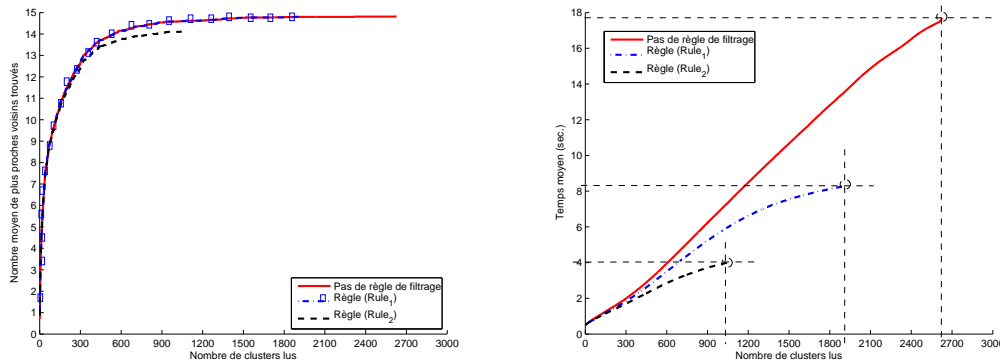
Utilisation du M -Tree. Un inconvénient du k -means est la nécessité de préciser le nombre de *clusters* à calculer. De plus, il n'est pas rare de trouver des *clusters* d'une seule image alors que d'autres sont beaucoup plus grands. Pour contourner ce problème, nous utilisons le M -Tree qui a la particularité de construire des nœuds à partir de leur taille passée en paramètre. Préciser le nombre de *clusters* que l'on souhaite calculer n'est donc plus nécessaire dans ce cas. Le M -Tree, contrairement au k -means, assure la mise à jour incrémentale des descripteurs puisque la construction de l'arbre se fait par insertion successive de vecteurs. D'autre part, les nœuds de grande taille permettent d'avoir des coûts d'entrée/sortie réduits. Dans nos expérimentations, la taille d'une page mémoire est de $4Ko$ et nous avons fixé la taille d'un nœud à $64Ko$ pour chaque type de descripteur.

Le tableau 4.10 indique le nombre moyen de *clusters* parcourus par type de descripteur en séquentiel, puis avec chacune des règles ($Rule_1$) et ($Rule_2$). La première ligne du tableau correspond aussi au nombre de *clusters* calculés avec le M -Tree pour chaque type de descripteur. L'analyse du tableau confirme les observations faites avec dans le cas du k -means : la règle ($Rule_2$) est bien plus sélective que la règle ($Rule_1$). De plus, ($Rule_1$) est beaucoup moins efficace sur les types EHD et RSD que sur les autres types de descripteurs. Ainsi, le M -Tree produit des *clusters* très équilibrés mais le taux de chevauchement reste le plus élevé pour EHD et RSD.

Nombre moyen de <i>clusters</i> parcourus	Type de descripteur				
	CLD	SCD	HTD	EHD	RSD
Séquentiel	216	434	688	867	422
$Rule_1$	163	305	291	755	392
$Rule_2$	96	172	162	280	336

TAB. 4.10 – Nombre moyen de *clusters* parcourus par type de descripteur en séquentiel, puis avec chacune des règles ($Rule_1$) et ($Rule_2$) sur B_3 .

Nous avons réalisé une étude comparative des règles ($Rule_1$) et ($Rule_2$) sur les *clusters* calculés à partir du M -Tree. La figure 4.14 montre qu'avec ($Rule_1$) on va jusqu'à $\frac{17,5}{8,5} \approx 2,06$ fois plus vite qu'une recherche séquentielle alors qu'avec ($Rule_2$) on va $\frac{17,5}{4} \approx 4,38$ fois plus vite pour plus de 14 images retrouvées sur 15. Le M -Tree préserve donc l'efficacité de la règle ($Rule_2$) par rapport à ($Rule_1$). Il est aussi intéressant de remarquer que, malgré un nombre total plus important de *clusters* avec le M -Tree, les performances sont différentes (et dans notre cas, légèrement moins bonnes) qu'avec le k -means. De façon générale, pour un nombre donné de *clusters* parcourus avec notre méthode de recherche, il est possible d'évaluer le nombre de vecteurs comparés à l'image-requête si les *clusters* ont été calculés avec le M -Tree (en multipliant le nombre de *clusters* parcourus par la taille moyenne d'un *cluster*) alors qu'avec le k -means, l'évaluation est plus difficile puisque les vecteurs ne sont pas toujours repartis de façon équilibrée dans les *clusters*.



(a) Qualité moyenne de la recherche en fonction du nombre de *clusters* lus sur B_3 (b) Temps moyen de la recherche en fonction du nombre de *clusters* lus sur B_3

FIG. 4.14 – Comparaison des règles de filtrage : utilisation du *M-Tree* pour le calcul des *clusters*.

4.5.4 Passage à l'échelle de la méthode de recherche

Le but de cette sous-section est d'étudier le comportement de notre méthode de recherche lorsque le nombre d'images de la base varie. Nous nous intéressons au passage à l'échelle sous deux aspects : la gestion des images hors-ligne et la recherche d'images en-ligne. Rappelons que l'organisation des images hors-ligne consiste à calculer les descripteurs globaux des images puis à les regrouper en *clusters*. Comme les descripteurs globaux sont utilisés, la taille des descripteurs d'images est proportionnelle au nombre d'images de la base. Désignons par N le nombre d'images de la base étudiée, par m le nombre de types de descripteurs utilisés, et par N_c est le nombre total de *clusters* pour tous les m types de descripteurs. La taille des descripteurs d'images est donc en $O(mN)$ et le calcul des *clusters* avec *k-means* est en $O(N_c N)$.

La recherche des k plus proches voisins est articulée autour de plusieurs étapes (voir section 3.2, page 56). Nous évaluons le coût de chaque étape comme suit.

1. Le calcul des descripteurs de l'image-requête ne dépend pas du nombre d'images de la base mais uniquement du type de descripteur *MPEG-7* considéré. Il se fait en $O(m)$ où m est le nombre de types de descripteurs.
2. Le calcul des caractéristiques des *clusters* dépend du nombre de *clusters* et des caractéristiques à évaluer. Il se fait en $O(N_c)$ avec N_c le nombre total de *clusters*.
3. Le tri des *clusters* sur la base de la mesure d'éligibilité définie par l'équation (4.2) est en $O(N_c \log N_c)$.
4. L'application des règles de filtrage à chaque *cluster* se fait en $O(N_c)$.
5. Le parcours des *clusters* et la fusion des résultats. La complexité de cette étape dépend du nombre des *clusters* retenus par la règle de filtrage choisie. Soit N_r ce nombre, et $|C|$ le nombre moyen d'images dans un *cluster*. Le parcours de chaque *cluster* d'un type de descripteur donné suivi de la mise à jour de la liste des résultats correspondant au descripteur du même type se fait en $O(k|C|)$ où

le nombre k de plus proches voisins recherchés est aussi la taille de la liste des résultats. L'agrégation des listes est de l'ordre de $O(km)$. Ainsi, la complexité de calcul à chaque phase de recherche est de l'ordre de $O(k|C| + km)$ et pour tous les N_r *clusters* retenus, elle est en $O(N_r(k|C| + km))$.

Le temps mis à retourner les images pertinentes par notre approche dépend essentiellement du nombre de *clusters* parcourus pendant la recherche et indirectement de la taille de la base d'images.

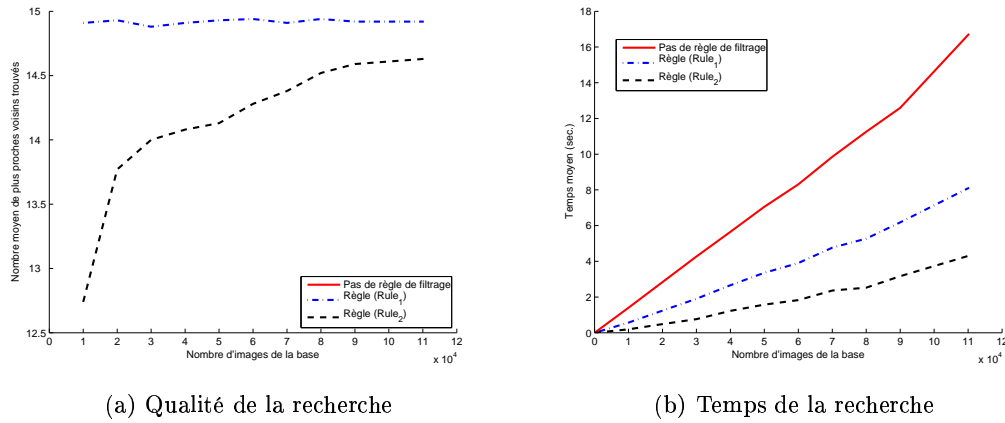


FIG. 4.15 – Influence de la taille de la base d'images sur la qualité et le temps de la recherche : utilisation du k -means pour le calcul des *clusters* sur B_3 .

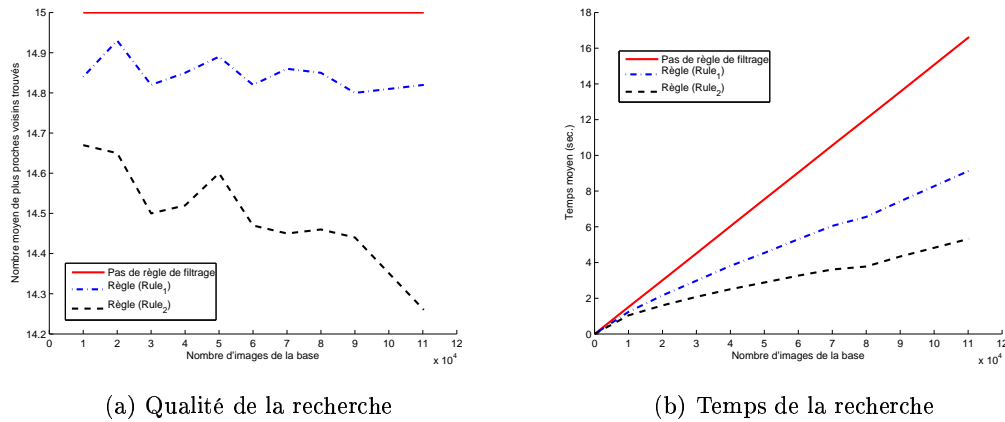


FIG. 4.16 – Influence de la taille de la base d'images sur la qualité et le temps de la recherche : utilisation du M -Tree pour le calcul des *clusters* sur B_3 .

Faisant varier la taille de la base d'images de 10000 à 110291, nous avons noté la qualité et le temps moyens de recherche, d'abord pour un parcours séquentiel exhaustif, puis avec les règles ($Rule_1$) et ($Rule_2$). Deux cas ont été étudiés : le cas où 100 *clusters*

sont calculés avec le *k-means* pour chacun des types de descripteurs et celui où les *clusters* sont calculés avec le *M-Tree* dans les conditions identiques à celles décrites à la sous-section 4.5.3 lors de l'évaluation des règles de filtrage avec le *M-Tree*. Les figures 4.15 et 4.16 illustrent le comportement de notre méthode lorsque le nombre d'images de la base varie. Elles confirment l'efficacité de la règle (*Rule*₂) par rapport à (*Rule*₁) en terme de temps de réponse. La qualité moyenne des résultats obtenus avec la règle (*Rule*₁) n'est rigoureusement pas la même que celle obtenue par une recherche séquentielle exhaustive parce que plusieurs images peuvent avoir le même score et être classées 15^{ième} dans la liste des 15 plus proches voisins. Le système choisit dans ce cas une seule image parmi les *ex aequo*. Cette situation ne remet donc pas en cause le principe de (*Rule*₁) qui assure l'exactitude des résultats retrouvés. La variation de la qualité des résultats avec (*Rule*₂) est beaucoup moins sensible sur les *clusters* du *M-Tree* que sur ceux du *k-means*. L'on observe en particulier qu'avec le *k-means*, la qualité des résultats croît avec le nombre d'images de la base jusqu'à 90000 et se stabilise par la suite (voir figure 4.16(a)). Quant aux résultats obtenus après le parcours des *clusters* du *M-Tree*, leur qualité moyenne est comprise entre 14,2 et 14,7 images sur 15, l'interrogation étant faite avec 100 images-requêtes. Ainsi, les performances de la méthode de recherche restent bonnes lorsque le nombre d'images augmente.

4.6 Intérêt de la combinaison des descripteurs globaux

Nous évaluons dans cette section comment la combinaison de différents types de descripteurs peut être plus intéressante que l'usage d'un seul type de descripteur dans un contexte de recherche par le contenu. La base *B*₄ est utilisée comme base de recherche. Chacune des 610 images de la base MOVI est soumise comme requête au système de recherche et les résultats pour chaque type de descripteur sont ensuite fusionnés. Le calcul du score du résultat retourné à l'utilisateur se fait avec l'équation présentée à la section 3.6 du chapitre 3. Nous calculons ensuite les mesures traditionnelles de rappel, de précision et de *F-mesure* [Rijsbergen, 1979] du résultat retourné.

Il n'existe pas de méthode standard de choix approprié du nombre de plus proches voisins. Comme les séquences de la base MOVI sont de différentes longueurs (en termes de nombre d'images qu'elles contiennent), nous avons choisi de retourner pour chaque requête le nombre de plus proches voisins égal à la longueur de la séquence qui contient la requête. Dans ce cas, le nombre d'images pertinentes est égal au nombre d'images retournées car les 610 images de la base MOVI ont été préalablement noyées dans *B*₃ pour constituer *B*₄ (voir section 4.3). Ainsi, le rappel (*rap*) et la précision (*prec*) ont toujours la même valeur et on a :

$$rap = prec \text{ et } F\text{-mesure} = \frac{2 \times rap \times prec}{rap + prec} = \frac{2 \times rap^2}{2 \times rap} = rap = prec$$

Nous présenterons donc dans la suite tous les résultats sur la base de la *F-mesure*.

D'après le tableau 4.11, le type de descripteur SCD a le meilleur taux de reconnaissance. Le descripteur de couleur est donc le plus performant sur nos données.

Type de descripteur	<i>F-mesure</i>
<i>CLD</i>	0,40
<i>SCD</i>	0,68
<i>HTD</i>	0,24
<i>EHD</i>	0,36
<i>RSD</i>	0,31

TAB. 4.11 – *F-mesure* par type de descripteur.

Un poids proportionnel à la *F-mesure* est ensuite affecté à chaque type de descripteur (voir tableau 4.11). La *F-mesure* du système de recherche est évaluée pour une série de combinaison de plusieurs types de descripteurs. Les résultats consignés dans le tableau 4.12 montrent que la combinaison de 4 types de descripteurs tels que *CLD-SCD-HTD-EHD* ou de 5 types de descripteurs tels que *CLD-SCD-HTD-EHD-RSD* donne un taux moyen de reconnaissance de 70%. Ce taux est bien plus intéressant que celui obtenu en utilisant uniquement le type de descripteur *SCD* (68%).

Combinaison des types de descripteurs	<i>F-mesure</i>
<i>CLD-SCD-HTD-EHD-RSD</i>	0,70
<i>CLD-SCD-HTD-EHD</i>	0,70
<i>CLD-SCD-RSD</i>	0,68
<i>CLD-SCD</i>	0,68
<i>SCD-EHD-RSD</i>	0,67
<i>HTD-EHD-RSD</i>	0,39
<i>HTD-EHD</i>	0,36

TAB. 4.12 – *F-mesure* du système avec combinaison de plusieurs types de descripteurs.

Nous remarquons également que les combinaisons telles que *CLD-SCD* ou *HTD-EHD* ne donnent pas des résultats meilleurs que *SCD* ou *EHD* utilisés seuls. D'autre part, *HTD-EHD-RSD* est meilleur que *HTD-EHD*. À ce stade, on pourrait conclure que dans la combinaison *CLD-SCD-HTD-EHD-RSD*, les types de descripteurs *SCD*, *EHD*, et *RSD* ont les rôles les plus importants en terme de reconnaissance. Les expérimentations conduites sur la combinaison constituée uniquement de *SCD*, *EHD*, et *RSD* montrent cependant que les résultats ne sont pas meilleurs que ceux obtenus sur la combinaison des 5 descripteurs (67% contre 70%; voir tableau 4.12). Il est donc plus avantageux dans la moyenne des cas de combiner au moins 4 types de descripteurs que d'en combiner uniquement 3.

Comme tous les systèmes de recherche d'images par le contenu, le taux de reconnaissance de notre système dépend de l'image-requête. Dans le meilleur des cas cependant, la combinaison de tous les 5 types de descripteurs peut donner des taux de reconnaissance allant jusqu'à 83% contre 72% avec *SCD* seul.

4.7 Synthèse

Nous avons évalué dans ce chapitre la méthode proposée de recherche progressive d'images par le contenu. Tous nos résultats liés aux performances en temps du système sont comparés à ceux d'une recherche séquentielle sur des bases de 7727, 30411, et 110291 images fixes avec 5 types de descripteurs *MPEG-7* dont deux de couleur, deux de texture, et un de forme. Les bases d'images sont organisées en *clusters* (selon deux méthodes : *k-means* et *M-Tree*) dont certaines caractéristiques telles que la taille peuvent être calculées hors-ligne. D'autres caractéristiques, plus dépendantes d'une image-requête, sont évaluées dans la phase de recherche. Nous normalisons toutes ces caractéristiques, puis nous les combinons en une mesure d'éligibilité. La recherche menée selon l'usage progressif des descripteurs visuels permet, lorsque les caractéristiques des *clusters* sont bien choisies, d'avoir plus rapidement la totalité du résultat séquentiel final. Il n'est pas nécessaire dans ce cas de parcourir tous les *clusters* pour tous les descripteurs. L'objectif de la mesure d'éligibilité est avant tout d'induire un ordre qui rende prioritaire le traitement des *clusters* les plus intéressants. D'autre part, l'utilisation des règles d'association altère la qualité des résultats partiels, mais cette altération est compensée par le gain en temps correspondant.

Les expérimentations conduites sur des *clusters* du *k-means* et sur ceux du *M-Tree* montrent que les règles de filtrage (*Rule₁*) et (*Rule₂*) améliorent l'efficacité de la recherche, indépendamment de la technique de *clustering*. Tandis que (*Rule₁*) préserve l'exactitude des résultats retournés, le gain important de temps obtenu par l'utilisation de (*Rule₂*) est compensé par une légère perte de la qualité relative des résultats. L'étude de l'impact du nombre d'images sur la recherche a montré que notre méthode s'adapte bien aux grandes bases d'images. Nos expérimentations ont enfin montré que la combinaison des descripteurs globaux permet d'améliorer le taux de reconnaissance des descripteurs.

Conclusion

Nous nous sommes intéressés dans le cadre de cette thèse à deux principaux aspects de l'amélioration des performances de la recherche d'images fixes par le contenu. Ces aspects sont liés au temps de recherche et à la qualité des résultats. L'objectif de notre travail a été de développer une approche rapide et efficace de sélection automatique des critères visuels destinée à faciliter la tâche à un utilisateur non spécialiste dans son activité de recherche d'images par le contenu. Pour atteindre notre but, nous avons fait un état de l'art sur l'amélioration de la qualité des résultats d'une recherche. Les travaux présentés dans cet état de l'art sont essentiellement basés sur les techniques de description du contenu mais très peu abordent le problème de l'amélioration des résultats sous l'angle de la combinaison de plusieurs techniques de description. Ils se contentent en général de proposer de nouveaux descripteurs et des distances dont les coûts de calculs peuvent être très élevés. Nous avons également rappelé les principaux problèmes liés à l'indexation multidimensionnelle des images et expliqué le phénomène de la malédiction de la dimension. Les travaux présentés dans l'état de l'art contournent ce phénomène par un codage particulier des données ou par un recours aux techniques approximatives qui permettent d'améliorer le temps de recherche.

Notre travail a d'abord consisté à étudier les techniques de fouille de données, puis à choisir celles qui seraient les plus avantageuses pour améliorer l'organisation et la recherche des images par le contenu. Dans ce cadre, nous avons montré comment la classification non supervisée (*clustering*) et la découverte des règles d'association peuvent s'intégrer dans un processus global d'organisation des images fixes pour accroître les performances de la recherche. Nous avons également formalisé la notion de recherche progressive par le contenu et montré comment une base d'images, préalablement organisée en *clusters* de descripteurs, peut être parcourue pour produire à l'utilisateur des résultats régulièrement agrégés en réponse à une requête par l'exemple. La suite de cette conclusion présente dans un premier temps la synthèse des résultats de nos travaux, puis des perspectives.

Synthèse des résultats

Nous partons de l'hypothèse qu'une base d'images peut être décrite par plusieurs types de descripteurs globaux au sein d'un système de recherche. Notre approche se situe donc dans un contexte de recherche par renforcement de ces critères ou descripteurs visuels que nous combinons pour améliorer la qualité de la recherche. Pour organiser

ces types de descripteurs, notre méthode part du principe classique qui consiste à regrouper les vecteurs de ces différents types de descripteurs en *clusters*. Deux techniques de regroupement ont été utilisées. La première, appelée *k-means*, est une méthode de partitionnement de données que nous avons choisie en raison de sa simplicité de mise en œuvre et des temps de calcul réduits. La deuxième est une adaptation de la méthode d'indexation *M-Tree*. L'adaptation consiste à n'utiliser que les feuilles de la structure d'arbre du *M-Tree* comme des *clusters*. Ces *clusters* sont compacts et les mises à jour sont simples puisque l'arbre est construit par insertions successives.

Une figure imposée de notre approche consiste à employer les techniques de fouille de données et de les appliquer aux images. Nous calculons des règles d'association pour mieux expliciter les relations entre les *clusters* constitués. Le processus d'organisation des images se faisant hors-ligne, il est donc absolument indépendant de toute image-requête. En-ligne, lorsqu'une requête par l'exemple est soumise, notre principe d'utilisation des règles d'association consiste à ignorer systématiquement les *clusters* qui apparaissent en partie droite des règles sélectionnées car la sémantique des règles découvertes est la suivante : « une image appartenant à tous les *clusters* de la partie gauche des règles se trouve aussi dans le *cluster* de la partie droite », la validité des règles étant jugée par leur support et leur confiance. Les expérimentations que nous avons conduites montrent que les règles d'association accélèrent notablement les temps de recherche tout en conservant une bonne qualité des résultats.

Nous avons proposé l'éligibilité d'un *cluster* comme une grandeur réelle qui résulte de la combinaison de toutes les caractéristiques du *cluster*, dépendantes ou non de l'image-requête. Nous l'avons définie pour induire un ordre de parcours de l'ensemble de tous les *clusters* (tous types de descripteurs confondus) afin de réduire les temps de recherche. D'après les expérimentations que nous avons conduites sur cinq types de descripteurs *MPEG-7*, il n'est pas nécessaire de parcourir toute la base d'images pour avoir les plus proches voisins recherchés. En particulier, pour 110291 images organisées en 500 *clusters* pour les types de descripteurs utilisés, il suffit de parcourir en moyenne 121 *clusters* pour avoir la totalité des 15 plus proches voisins recherchés. Les caractéristiques des *clusters* dépendantes de l'image-requête soumise contribuent davantage à limiter le parcours des *clusters* que les caractéristiques indépendantes de l'image-requête. Il est ainsi possible de retrouver tous les résultats recherchés en moins d'un tiers du temps de recherche séquentiel. Toutefois, la mesure d'éligibilité ne donne pas le moyen d'arrêter automatiquement la recherche dès que les plus proches voisins ont été retrouvés. Elle permet tout simplement de se rapprocher au plus vite des résultats.

Les deux règles de filtrage étudiées dans cette thèse permettent d'éliminer les *clusters* non pertinents pour la recherche envisagée. La première règle, plus classique, consiste à éliminer systématiquement tous les *clusters* dont la distance minimum à l'image-requête est supérieure à un seuil calculé et non nécessairement nul. Quant à la deuxième règle, elle consiste à ne retenir que les *clusters* dont la distance minimum à l'image-requête est nulle. Cette règle permet une recherche plus rapide que la règle classique mais introduit une approximation dans les résultats de la recherche. Nos expérimentations ont permis de quantifier cette approximation, et surtout, de montrer que l'efficacité relative des règles de filtrage ne dépend pas des techniques de *clustering* que nous avons utilisées

(*k-means*, *M-Tree*). De plus, la règle de filtrage que nous avons proposée donne une minoration du taux de chevauchement des *clusters* dans l'espace multidimensionnel des descripteurs. En effet, les formes englobantes de tous les *clusters* que cette règle retient pour la recherche sont d'intersection non vide (elles ont en commun au moins l'image-requête). Sur des bases d'images de plus en plus grandes (de 10000 à plus de 100000 images), le comportement de notre méthode de recherche a montré que celle-ci reste de loin préférable à une simple recherche séquentielle menée à l'aveugle.

Nous avons enfin montré la valeur ajoutée de la combinaison des descripteurs globaux pour la recherche d'images par le contenu. Pour cela, nous avons conduit les expérimentations sur cinq types de descripteurs *MPEG-7* dont deux de couleur, deux de texture, et un de forme. Les résultats montrent que la combinaison des descripteurs globaux améliorent le taux de reconnaissance du système de recherche. Plus généralement, la combinaison d'un ensemble quelconque de descripteurs globaux peut se faire en évaluant, dans un premier temps, le taux de reconnaissance de chacun des descripteurs sur un ensemble d'images-tests. Dans notre travail, le taux de reconnaissance a été estimé par la *F-mesure*, les images-tests ayant été constituées de sorte que l'on sache distinguer précisément les images pertinentes de celles qui ne le sont pas pour une image-requête. À chaque descripteur rentrant dans une combinaison donnée est ensuite affectée un poids proportionnel à son taux de reconnaissance estimé.

Perspectives

Nos perspectives de travail concernent à la fois les aspects techniques à travers le fonctionnement de notre système de recherche et les aspects applicatifs à travers la gestion intuitive de grandes bases d'images fixes par l'utilisateur. Sur le plan technique, nous envisageons d'étendre notre travail sur l'éligibilité des *clusters* ainsi que sur la gestion des règles de filtrage. Sur le plan applicatif, il est intéressant d'aborder la gestion des mises à jour aussi bien des images que des techniques de description. L'extension du système à la gestion d'autres types de descripteurs devra aussi être considérée.

Éligibilité des *clusters*

Nous avons défini et étudié expérimentalement dans ce travail la notion d'éligibilité comme moyen d'unification de toutes les caractéristiques mesurables des *clusters*. Il est cependant difficile de vouloir établir une liste exhaustive de toutes les caractéristiques envisageables pour les *clusters*. Toutefois, l'étude que nous avons faite montre que les caractéristiques dépendantes de l'image-requête doivent être privilégiées. Une connaissance de la distribution des points (vecteurs) d'un *cluster* par rapport à une requête apporterait sans doute de l'information supplémentaire et permettrait de cibler, en plus du *cluster*, les vecteurs à parcourir en priorité. On passerait ainsi à un niveau d'éligibilité plus fin que celui des *clusters*. Cette piste requiert des techniques d'échantillonnage sur les images-requêtes pour être efficacement mise en œuvre. Elle est cependant difficile à explorer dans le cas des images « tout venant » où l'on a pas toujours des informations

sur la source des requêtes. L'étude sur l'éligibilité peut être approfondie pour examiner en détail les distances entre les vecteurs de descripteurs de façon à identifier les dimensions les plus discriminantes dans la similarité des images.

Gestion des mises à jour du système

Deux aspects de la mise à jour du système doivent être envisagés : celui des images et celui des descripteurs. Les algorithmes *k-means* de calcul des *clusters* et *Apriori* de découverte de motifs fréquents que nous avons utilisés ne sont pas incrémentaux. L'ajout/suppression d'une seule image nécessite de recalculer de nouveau tous les *clusters* et de régénérer les règles d'association pour tenir compte de la modification de la base d'image. Il est bien inefficace de recommencer tout le processus d'organisation des images après une simple opération d'ajout/suppression. L'avantage du *M-Tree* est qu'il procède par insertions successives. Ainsi, il n'est pas nécessaire de recalculer tous les *clusters* si le *M-Tree* est utilisé. Il serait par contre intéressant de s'orienter vers des algorithmes incrémentaux de découverte de règles d'association tels que NFUP³ [Chang *et al.*, 2005].

Un autre aspect de la mise à jour du système concerne la modification des types de descripteurs. En effet, quelle que soit la méthode de *clustering* utilisée (incrémentale ou pas), l'ajout d'un nouveau type de descripteur entraînera systématiquement le parcours de toute la base d'images pour calculer des *clusters* associés au type de descripteur ajouté. Il sera toutefois judicieux de privilégier les techniques de découverte des règles d'association qui gèrent efficacement l'ajout d'un nouvel attribut dans une table de transactions.

Description globale et description locale des images

Les descripteurs locaux sont très précis car ils s'intéressent aux propriétés locales des images. Mais le principe de description et de recherche peut s'avérer plus long comparé aux descriptions globales. Il s'avère donc intéressant de comparer les résultats obtenus avec la méthode proposée à ceux que l'on obtiendrait par recherche sur les descripteurs locaux (voir aussi les temps de recherche des deux approches). Cela permettrait par exemple de privilégier une combinaison de descripteurs globaux aux descripteurs locaux si les qualités de recherche des deux systèmes sont voisines et si la combinaison des descripteurs globaux permet une recherche plus rapide. Les descriptions locales ont l'inconvénient d'accroître la taille des descripteurs d'images comparativement aux descripteurs globaux de même dimension. Elles soulèvent ainsi de nouvelles interrogations telles que la gestion des temps d'accès-disque dont il faudra tenir compte.

³New Fast UPdate algorithm

Annexe A

Descripteurs *MPEG-7*

A.1 Contexte de standardisation

Les contenus multimédias sont désormais accessibles et peuvent être générés par tous grâce à la vulgarisation des caméras numériques, de l'ordinateur et d'Internet. La diversification de ses sources de production et son volume en constante augmentation pose des problèmes liés à la représentation et à la gestion de ces contenus. Les documents numériques sont généralement annotés à la main et sans règle précise. Un traitement de ce type est difficilement envisageable sur de grandes quantités de documents du fait de sa subjectivité et des délais très longs. Les applications qui traitent le contenu n'interopèrent généralement pas, puisqu'elles développent des structures de données et méthodes de description qui leur sont souvent propres. La nécessité de décrire objectivement les données audiovisuelles par des outils capables d'extraire automatiquement les caractéristiques du contenu qui remplacent ou complètent les descriptions manuelles est alors vite apparue. C'est dans ce contexte que le groupe *MPEG* (*Moving Picture Expert Group*) a initié en Juillet 1996 le projet de standardisation *MPEG-7* (*Multimedia Content Description Interface*).

Le principal objectif du projet *MPEG-7* [Manjunath *et al.*, 2002] est de fournir un cadre normatif de description, suffisamment générique et flexible pour s'appliquer à différents types de médias (images, vidéo, parole, musique, *etc.*) et pouvant convenir à un large éventail d'applications. Ce standard, très modulaire, comprend une panoplie d'outils élémentaires qui permettent de décrire le contenu à différents niveaux d'abstraction. *MPEG-7* ne standardise que le format de description du contenu. Il ne spécifie pas comment obtenir automatiquement les caractéristiques à partir des documents multimédias, ni comment utiliser les descriptions produites. Cette flexibilité permet aux développeurs de concevoir de nouvelles applications dont l'interopérabilité est assurée par la représentation standard du contenu qu'offre *MPEG-7*. Les techniques de description *MPEG-7* sont ainsi caractérisées par leur généralité et leur extensibilité grâce au langage de définition de descriptions (*Description Definition Language* ou *DDL*) permettant à un utilisateur de paramétrer *MPEG-7* pour un usage spécifique.

Nous nous intéressons dans cet annexe au principe de description de *MPEG-7*. Nous

décrivons ensuite les différents descripteurs visuels de couleurs de texture et de forme proposés par le standard et que nous avons utilisé dans notre travail de thèse, en particulier pour nos besoins expérimentaux.

A.2 Description du contenu

Une description *MPEG-7* est un fichier de métadonnées détaillant différents aspects du contenu et de la gestion d'un document audiovisuel. Elle est composée de schémas de description et de descripteurs. Un schéma de description (voir Figure A.1) peut avoir un ou plusieurs descripteurs et contenir d'autres schémas de description. Dans la pratique, on peut avoir des structures beaucoup plus complexes, avec des imbrications récursives. En ce qui concerne le contenu par exemple, on pourra retrouver, dans le fichier de description, des informations sur les caractéristiques de bas niveau (couleurs, textures, mouvement, contenu fréquentiel, mélodie, *etc.*) ou des concepts sémantiques de plus haut niveau (objets, événements, interactions entre les objets, lieux de l'action, *etc.*).

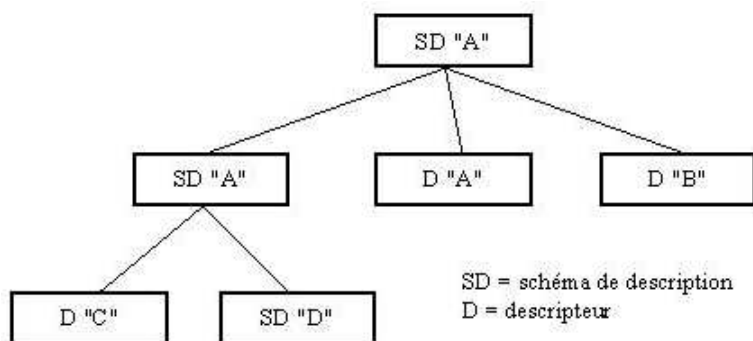


FIG. A.1 – Schéma de description MPEG-7 d'un document multimédia.

De façon plus précise, les descripteurs de contenu sont élaborés à partir :

- **des informations de gestion et de contrôle** permettant une description de haut niveau. On peut par exemple citer les métadonnées générées manuellement qui décrivent le média, sa création et son utilisation ;
- **des informations sémantiques** décrivant les entités sémantiques telles que les objets ou les événements présents dans une scène. La description est en général textuelle et concerne les attributs et les relations entre entités décrites ;
- **des informations de structure** décrivant les segments temporels ou spatiaux du contenu multimédia. Pour une image, on décrira, par exemple, les différentes régions qui la composent. L'extraction de ce type d'information est quasi-automatique (comme la couleur, la texture, et la forme pour les descripteurs visuels).

A.3 Quelques descripteurs MPEG-7 d'images fixes

Le standard *MPEG-7* propose des descripteurs visuels utiles à l'identification et à la catégorisation des images fixes. Ces descripteurs, généralistes ou spécifiques, peuvent être utilisés seuls ou combinés à d'autres types de descripteurs du contenu (descripteurs textuels par exemple). Les descripteurs spécifiques sont liés à une application précise (telle que les détecteurs de visages par exemple). Nous nous intéressons aux descripteurs *MPEG-7* plus généraux classés en trois grandes catégories : les descripteurs de couleur, les descripteurs de texture et ceux de forme. Les procédés d'extraction et de comparaison des descripteurs proposés ne sont pas standardisés, mais *MPEG-7* en donne quelques recommandations que nous abordons dans ce paragraphe. Les détails pourront être trouvés dans [Manjunath *et al.*, 2002]. Nous conserverons par la suite, les dénominations en anglais des différents types de descripteurs présentés. Nous donnerons également, pour chacun de ces types de descripteurs un exemple de description de l'image de la figure A.2.



FIG. A.2 – Exemple d'image à décrire avec *MPEG-7*.

A.3.1 Les descripteurs de couleur

La couleur est un facteur important de la perception visuelle pour l'homme et pour le traitement par ordinateur. Une bonne description de la couleur doit obéir aux principaux critères suivants :

- l'aptitude à caractériser au mieux la similarité perceptuelle de couleur. Cette aptitude est jugée sur la performance des descripteurs de couleur dans la comparaison des images ;
- une faible complexité de calcul et de comparaison des descripteurs ;
- la taille de la description qui joue un rôle important dans l'indexation et la transmission des données ;
- la passage à l'échelle et l'interopérabilité des descripteurs de couleurs.

MPEG-7 utilise deux principaux types de données pour manipuler les descripteurs de couleur :

- la structure *Color Space* est utilisée pour spécifier l'espace de couleur utilisé par un descripteur. Elle permet de définir quatre espaces de couleur : **RGB** (*Red*,

Green, Blue), **YCbCr** (*luminancy, blue, red*), **HSV** (*Hue, Saturation, Value*) et **HMMD** (*Hue-Max-min-Diff*). Un nouvel espace de couleur peut toutefois être défini par spécification d'une transformation linéaire de l'espace RGB.

- la structure *Color Quantization* décrit la quantification de l'espace de couleur utilisé par un type de descripteur donné. La quantification peut être uniforme ou non.

MPEG-7 définit six types de descripteurs de couleur. Nous nous limitons dans cette sous-section à la présentation des deux types de descripteurs effectivement utilisés dans nos expérimentations : le descripteur *Color Layout* et le descripteur *Scalable Color*.

A.3.1.1 Le descripteur *Color Layout*

Le descripteur *Color Layout* (*CLD* pour *Color Layout Descriptor*) caractérise de façon compacte la distribution spatiale d'une couleur dans une image. Il peut être utilisé pour une recherche rapide dans des bases de données et comme filtre dans des applications de diffusion. La figure A.3 donne la description *MPEG-7* de l'image A.2 au format XML à l'aide du descripteur de type *Color Layout*. Toutes les balises encadrant les coefficients du descripteur *CLD* sont préfixées par **Y**, **Cb**, ou **Cr** rappelant ainsi l'espace de description **YCbCr**.

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi =
  "http://www.w3.org/2000/10/XMLSchema-instance">
  <DescriptionUnit xsi:type = "DescriptorCollectionType">
  <Descriptor xsi:type = "ColorLayoutType"><YDCCoeff>34</YDCCoeff>
  <CbDCCoeff>32</CbDCCoeff>
  <CrDCCoeff>34</CrDCCoeff>
  <YACCCoeff5>22 22 6 15 16 </YACCCoeff5>
  <CbACCCoeff2>16 21 </CbACCCoeff2>
  <CrACCCoeff2>16 11 </CrACCCoeff2>
  </Descriptor>
  </DescriptionUnit>
</Mpeg7>
```

FIG. A.3 – Exemple de descripteur *MPEG-7* de type *Color Layout*.

L'extraction du descripteur se fait en quatre étapes :

1. l'image est décomposée en 8×8 blocs ;
2. la couleur représentative de chaque bloc est sélectionnée dans l'espace **YCbCr** (*luminancy, blue, red*). Toutes les méthodes de sélection peuvent être utilisées à cette étape mais en général, la valeur moyenne de la couleur d'un bloc est adoptée comme couleur représentative et le résultat est stocké dans un tableau ;
3. la transformée en cosinus discrète (**DCT** pour *Discrete Cosine Transform*) est appliquée sur le tableau obtenu à l'étape précédente ;

4. les coefficients obtenus sont réorganisés en zig-zag et ceux de plus basse fréquence sont retenus comme paramètres du descripteur. Le nombre de coefficients choisis est variable et précisé dans un champ spécial appelé *CoefficientPattern field*.

La mesure de distance entre deux descripteurs $\{DY, DCb, DCr\}$ et $\{DY', DCb', DCr'\}$ de type CLD est définie par :

$$D = \sqrt{\sum_i w_{yi}(Dy_i - Dy'_i)^2} + \sqrt{\sum_i w_{bi}(Db_i - Db'_i)^2} + \sqrt{\sum_i w_{ri}(Dr_i - Dr'_i)^2}$$

où DY et DY' sont les composantes des descripteurs suivant Y ; DCb et DCb' les composantes des descripteurs suivant Cb ; DCr et DCr' les composantes des descripteurs suivant Cr . w_{yi} , w_{bi} , w_{ri} sont des poids associés aux composantes des descripteurs.

A.3.1.2 Le descripteur *Scalable Color*

Le descripteur *Scalable Color* (*SCD* pour *Scalable Color Descriptor*) est un histogramme de couleurs extrait dans l'espace **HSV** (*Hue, Saturation, Value*) et encodé pour un stockage efficace. Il est utilisé pour la recherche par similarité dans des bases multimédias et pour l'exploration de grandes bases de données. L'extraction du descripteur se décompose en deux principales étapes :

1. le calcul de l'histogramme de couleur composé de 256 partitions dans l'espace HSV dont 16 partitions pour la dimension H et 4 pour les dimensions S et V.
2. l'application à l'histogramme obtenu d'une série de transformations dite transformations 1-D de Haar pour obtenir un histogramme sous une forme beaucoup plus robuste (*scalable*).

La figure A.4 donne la description *MPEG-7* de l'image A.2 au format XML à l'aide du descripteur de type *Scalable Color*. 32 coefficients ont été retenus dans cette description.

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi =
  "http://www.w3.org/2000/10/XMLSchema-instance">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "ScalableColorType" NumberOfCoefficients = "1"
  NumberOfBitplanesDiscarded = "3"><Coefficients>6 0 9 10 0 1 0
  0 3 3 3 -1 -5 0 -3 0 0 0 1 1 0 0 0 0 -1 0 -1 1
  -1 0 0 0 </Coefficients>
</Descriptor>
</DescriptionUnit>
</Mpeg7>
```

FIG. A.4 – Exemple de descripteur *MPEG-7* de type *Scalable Color*.

La norme L_1 est utilisée comme mesure de distance entre deux vecteurs de descripteurs de type SCD.

A.3.2 Les descripteurs de texture

La texture est considérée après la couleur comme un facteur pertinent de la description du contenu. Les applications de ce type de description se retrouvent dans la recherche par similarité, l'exploration et l'indexation. *MPEG-7* propose trois types de descripteurs de texture : *Homogeneous Texture Descriptor* (**HTD**), *Edge Histogram Descriptor* (**EHD**), et *Perceptual Browsing Descriptor* (**PBD**). Nous ne présentons dans cette sous-section que les descripteurs **HTD** et **EHD** qui ont été utilisés.

A.3.2.1 Le descripteur *Homogeneous Texture*

Le descripteur *Homogeneous Texture* (HTD) décrit la distribution statistique de la texture d'une image. C'est un *n-uplet* de la forme :

$$HTD = [f_{DC}, f_{SD}, e_1, e_2, \dots, e_{30}, d_1, d_2, \dots, d_{30}]$$

où f_{DC} et f_{SD} sont respectivement la moyenne et l'écart-type de l'image. Les autres coefficients sont l'énergie (e_i) et l'écart-type de l'énergie (d_i) des réponses aux filtres de Gabor des canaux fréquentiels du domaine représenté à la figure A.5.

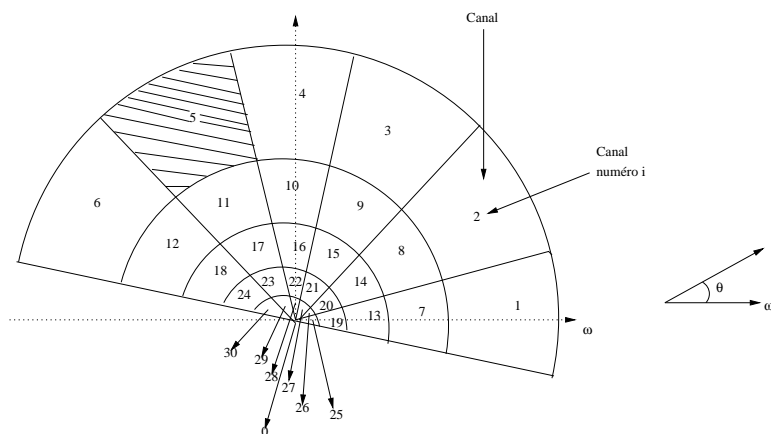


FIG. A.5 – Canaux utilisés dans l'extraction du descripteur Homogeneous Texture (HTD).

La figure A.6 donne un exemple de tous ces coefficients à travers la description *MPEG-7* de l'image A.2 au format XML.

La mesure de dissimilarité entre deux vecteurs V_1 et V_2 de descripteurs de type HTD est définie par :

$$d(V_1, V_2) = \sum_k \left| \frac{V_1(k) - V_2(k)}{\alpha(k)} \right|$$

où $\alpha(k)$ représente le poids de la composante d'ordre k des vecteurs. La valeur recommandée de $\alpha(k)$ est l'écart-type de toutes les composantes d'ordre k des descripteurs d'une base donnée. Mais $\alpha(k)$ peut aussi être fixé par l'utilisateur, par exemple $\alpha(k) = 1$.

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi =
  "http://www.w3.org/2000/10/XMLSchema-instance">
  <DescriptionUnit xsi:type = "DescriptorCollectionType">
  <Descriptor xsi:type = "HomogeneousTextureType"><Average>161</Average>
  <StandardDeviation>73</StandardDeviation>
  <Energy>197 185 185 187 208 210 185 165 169 163 172 167 158
    146 132 156 124 152 131 133 132 136 128 128 143 142 136
    130 141 150 </Energy>
  <EnergyDeviation>195 183 182 184 209 212 175 161 166 156 163
    160 148 138 118 143 107 149 122 112 130 117 123 125 134
    120 119 116 107 140 </EnergyDeviation>
  </Descriptor>
  </DescriptionUnit>
</Mpeg7>
```

FIG. A.6 – Exemple de descripteur MPEG-7 de type *Homogeneous Texture*.

A.3.2.2 Le descripteur *Edge Histogram*

Le descripteur EHD calcule un histogramme qui caractérise l'orientation des principaux contours d'une image. Il décrit les contours de chaque bloc d'image obtenus par subdivision de l'image entière en une grille de taille 4×4 . Les contours sont classés en cinq principaux types : *vertical*, *horizontal*, *45-degrés*, *135-degrés* et *sans-direction*. L'occurrence de chacun de ces types est représentée par une partition de l'histogramme. En définitive, on obtient un histogramme de 80 partitions dont la sémantique est décrite à la figure A.7.

H_E	Sémantique
h(0)	Nombre relatif de pixels de direction verticale dans le bloc (0,0)
h(1)	Nombre relatif de pixels de direction horizontale dans le bloc (0,0)
h(2)	Nombre relatif de pixels de direction 45-degrés dans le bloc (0,0)
h(3)	Nombre relatif de pixels de direction 135-degrés dans le bloc (0,0)
h(4)	Nombre relatif de pixels sans direction dans le bloc (0,0)
⋮	⋮
h(75)	Nombre relatif de pixels de direction verticale dans le bloc (3,3)
h(76)	Nombre relatif de pixels de direction horizontale dans le bloc (3,3)
h(77)	Nombre relatif de pixels de direction 45-degrés dans le bloc (3,3)
h(78)	Nombre relatif de pixels de direction 135-degrés dans le bloc (3,3)
h(79)	Nombre relatif de pixels sans direction dans le bloc (3,3)

FIG. A.7 – Sémantique de l'histogramme de partition du descripteur Edge Histogram (EHD).

La figure A.8 donne un exemple de tous ces coefficients à travers la description

MPEG-7 de l'image A.2 au format XML.

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi =
  "http://www.w3.org/2000/10/XMLSchema-instance">
<DescriptionUnit xsi:type = "DescriptorCollectionType">
<Descriptor xsi:type = "EdgeHistogramType"><BinCounts>1 2 4 5 6
  2 1 2 6 7 2 1 3 4 6 1 1 2 2 6 2 1 4 3 5 5 1 1
  5 6 4 1 1 6 7 1 2 2 4 6 2 5 3 5 5 3 3 3 4 6 5
  2 4 6 6 1 5 3 4 6 1 5 5 4 7 1 6 2 6 5 1 6 4 4
  5 1 3 4 5 6 </BinCounts>
</Descriptor>
</DescriptionUnit>
</Mpeg7>
```

FIG. A.8 – Exemple de descripteur MPEG-7 de type *Edge Histogram*.

La définition de la mesure de dissimilarité entre deux vecteurs A et B de descripteurs de type EHD nécessite un calcul de composantes supplémentaires à partir de celles définies à la figure A.7. Notons $h_A(i)$, $i = 0..79$ (resp. $h_B(i)$, $i = 0..79$) les composantes du descripteur A (resp. B). La distance entre A et B est définie par :

$$d(A, B) = \sum_{i=0}^{79} |h_A(i) - h_B(i)| + 5 \times \sum_{i=0}^4 |h_A^g(i) - h_B^g(i)| + \sum_{i=0}^{64} |h_A^S(i) - h_B^S(i)|$$

où $h_A^g(i)$ et $h_B^g(i)$ (resp. $h_A^S(i)$ et $h_B^S(i)$) sont les valeurs de l'histogramme global et de l'histogramme semi-global déduit de $h_A(i)$ (resp. $h_B(i)$).

A.3.3 Les descripteurs de forme

L'identification des objets dans une image est parfois liée à leur forme. L'œil humain parvient à reconnaître les caractéristiques d'un objet uniquement à partir de la forme de celui-ci. Les objets les plus courants sont en dimension $3D$ et MPEG-7 propose des descripteurs pour ces types d'objets [Zaharia et Preteux, 2001]. Sur les images par contre, on ne retrouve que les projections $2D$ des objets dont la représentation réelle en $3D$ est parfois inconnue. MPEG-7 propose les types de descripteurs *Contour-based Shape* (**CSD**) et *Region-based Shape* (**RSD**) pour décrire ces objets $2D$. Nous présentons le descripteur **RSD** utilisé dans notre travail.

Le descripteur **RSD** détermine la distribution des pixels dans un objet $2D$ ou dans une région. Il décrit aussi bien les objets simples que les objets complexes composés de plusieurs régions puisqu'il tient à la fois compte des pixels internes et des contours. Il utilise une transformation radiale angulaire (ART) définie sur une unité de disque dans un système de coordonnées polaire [Kim et Kim, 2000]. Le descripteur **RSD** est entièrement défini par un vecteur de 35 coefficients comme le montre la figure A.9, description MPEG-7 de l'image A.2 au format XML.

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<Mpeg7 xmlns = "http://www.mpeg7.org/2001/MPEG-7_Schema" xmlns:xsi =
  "http://www.w3.org/2000/10/XMLSchema-instance">
  <DescriptionUnit xsi:type = "DescriptorCollectionType">
  <Descriptor xsi:type = "RegionShapeType">
  <MagnitudeOfART>11 15 13 15 11 13 14 12 15 9 15 15 11 12
  15 14 13 13 8 12 13 12 7 7 8 6 14 10 10 8 8 5 12
  10 3 </MagnitudeOfART>
  </Descriptor>
  </DescriptionUnit>
</Mpeg7>
```

FIG. A.9 – Exemple de descripteur MPEG-7 de type *Region-based Shape*.

La norme L_1 est utilisée comme mesure de distance entre deux vecteurs de descripteurs de type RSD.

Bibliographie

- [Agrawal *et al.*, 1993] Agrawal, R., Imielinski, T., et Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, District of Columbia, USA.
- [Agrawal et Srikant, 1994] Agrawal, R. et Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of 20th International Conference on Very Large Databases (VLDB'94)*, pages 487–499, Santiago de Chile, Chile.
- [Agrawal et Srikant, 1995] Agrawal, R. et Srikant, R. (1995). Mining Sequential Patterns. In *Proceedings of the 11th International Conference on Data Engineering (ICDE '95)*, pages 3–14, Washington, District of Columbia, USA.
- [Akkus et Güvenir, 1996] Akkus, A. et Güvenir, H. (1996). k -Nearest Neighbor Classification on Feature Projections. In *Proceedings of the 13th International Conference on Machine Learning (ICML'96)*, pages 12–19, Bari, Italy.
- [Amsaleg et Gros, 2001] Amsaleg, L. et Gros, P. (2001). Content-Based Retrieval using Local Descriptors : Problems and Issues from a Database Perspective. *Pattern Analysis and Applications, Special Issue on Image Indexation*, 4, 108–124.
- [Antunes et Oliveira, 2004] Antunes, C. et Oliveira, A. L. (2004). Sequential Pattern Mining Algorithms : Trade-offs between Speed and Memory. In *Workshop on Mining Graphs, Trees and Sequences (MGTS - ECML/PKDD 2004)*, Pisa, Italy.
- [Aouiche *et al.*, 2003] Aouiche, K., Darmont, J., et Gruenwald, L. (2003). Frequent Itemsets Mining for Database Auto-Administration. In *Proceedings of the 7th International Database Engineering and Application Symposium (IDEAS 03)*, pages 98–103, Hong Kong, China.
- [Ashley *et al.*, 1995] Ashley, J., Barber, R., Flickner, M., Hafner, J., Lee, D., Niblack, W., et Petkovic, D. (1995). Automatic and Semi-Automatic Methods for Image Annotation and Retrieval in Query by Image Content (QBIC). In *Proceedings of SPIE Conference 2420 on Storage and Retrieval for Image and Video Databases III*, pages 24–35.
- [Aufaure et Hajji, 2002] Aufaure, M.-A. et Hajji, H. (2002). Semantic Structuration of Image Annotations : A Data Mining Approach. In *Proceedings of International Workshop on Multimedia Information Systems*, Tempe, Arizona, USA.

- [Ayres *et al.*, 2002] Ayres, J., Flannick, J., Gehrke, J., et Yiu, T. (2002). Sequential Pattern Mining using a Bitmap Representation. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pages 429–435, Edmonton, Alberta, Canada.
- [Bajscy, 1973] Bajscy, R. (1973). Computer Identification of Visual Surfaces. *Computer Graphics and Image Processing*, 2, 118–130.
- [Balko *et al.*, 2004] Balko, S., Schmitt, L., et Saake, G. (2004). The Active Vertice Method : a Performant Filtering Approach to High-Dimensional Indexing. *Data and Knowledge Engineering*, 51(3), 369–397.
- [Bastide *et al.*, 2002] Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., et Lakhal, L. (2002). Pascal : un algorithme d'extraction des motifs fréquents. *Technique et science informatiques*, 21(1), 65–95.
- [Bechmann *et al.*, 1990] Bechmann, N., Kriegel, H.-P., Schneider, R., et Seeger, B. (1990). The R*-tree : An Efficient and Robust Access Method for Points and Rectangles. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, pages 322–331, Atlantic City, New Jersey, USA.
- [Bennett *et al.*, 1999] Bennett, K. P., Fayyad, U., et Geiger, D. (1999). Density-Based Indexing for Approximate Nearest-Neighbor Queries. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99)*, pages 233–243, San Diego, California, USA.
- [Bentley, 1979] Bentley, J.-L. (1979). Multidimensional Binary Search in Database Applications. *IEEE Transactions on Software Engineering*, 4(5), 333–340.
- [Berrani, 2004] Berrani, S.-A. (2004). *Recherche approximative de plus proches voisins avec contrôle probabiliste de la précision ; application à la recherche d'images par le contenu*. Thèse de Doctorat, Université de Rennes 1.
- [Berrani *et al.*, 2002] Berrani, S.-A., Amsaleg, L., et Gros, P. (2002). Recherche par similarité dans les bases de données multidimensionnelles : panorama des techniques d'indexation. *RSTI -Ingénierie des systèmes d'information. Bases de données et multimédia*, 7(5-6), 9–44.
- [Berrani *et al.*, 2003] Berrani, S.-A., Amsaleg, L., et Gros, P. (2003). Approximate Searches : k-Neighbors + Precision. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM '03)*, pages 24–31, New Orleans, Louisiana, USA.
- [Berretti *et al.*, 2004] Berretti, S., Bimbo, A. D., et Pala, P. (2004). Merging Results for Distributed Content Based Image Retrieval. *Multimedia Tools and Applications*, 24(3), 215–232.
- [Beyer *et al.*, 1999] Beyer, K. S., Goldstein, J., Ramakrishnan, R., et Shaft, U. (1999). When Is "Nearest Neighbor" Meaningful? In *Proceeding of the 7th International Conference on Database Theory (ICDT '99)*, pages 217–235, London, UK, Springer-Verlag.
- [Bimbo, 1999] Bimbo, A. D. (1999). *Visual Information Retrieval*. Morgan Kaufmann.

- [Bouet et Khenchaf, 2002] Bouet, M. et Khenchaf, A. (2002). Traitement de l'information multimédia : Recherche du média image. *RSTI -Ingénierie des systèmes d'information. Bases de données et multimédia*, 7(5-6), 65–90.
- [Breiman *et al.*, 1984] Breiman, L., Friedman, J. H., Olshen, R. A., et Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth Intl. Group, Belmont, CA.
- [Carpineto et Romano, 1993] Carpineto, C. et Romano, G. (1993). GALOIS : An Order-Theoretic Approach to Conceptual Clustering. In *Proceedings of the 10th International Conference on Machine Learning (ICML'93)*, pages 33–40, Amherst, Massachusetts, USA.
- [Cha *et al.*, 2002] Cha, G., Zhu, X., Petkovic, D., et Chang, C. (2002). An Efficient Indexing Method for Nearest Neighbor Searches in High-Dimensional Image Databases. *IEEE Transactions on Multimedia*, 4(1), 76–87.
- [Chakrabarti et Mehrotra, 2000] Chakrabarti, K. et Mehrotra, S. (2000). Local Dimensionality Reduction : A New Approach to Indexing High Dimensional Spaces. In *Proceedings of 26th International Conference on Very Large Databases (VLDB'00)*, pages 89–100, Cairo, Egypt.
- [Chang *et al.*, 2005] Chang, C.-C., Li, Y.-C., et Lee, J.-S. (2005). An Efficient Algorithm for Incremental Mining of Association Rules. In *Proceedings of 15th International Workshop on Research Issues in Data Engineering : Stream Data Mining and Applications (RIDE-SDMA'05)*, pages 37–10, Tokyo, Japan.
- [Chavez *et al.*, 2001] Chavez, E., Navarro, G., Baeza-Yates, R., et Marroquin, J. L. (2001). Searching in Metric Spaces. *ACM Computing Survey*, 33(3), 273–321.
- [Cheeseman et Stutz, 1996] Cheeseman, P. et Stutz, J. (1996). Bayesian Classification (AutoCclass) : Theory and Results. In *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press.
- [Ciaccia *et al.*, 1997] Ciaccia, P., Patella, M., et Zezula, P. (1997). M-tree : An Efficient Access Method for Similarity Search in Metric Spaces. In *Proceedings of the 23rd International Conference on Very Large Databases (VLDB'97)*, pages 426–435, Athens, Greece.
- [Deng et Manjunath, 1999] Deng, Y. et Manjunath, B. S. (1999). An Efficient Low-Dimensional Color Indexing Scheme for Region Based Image Retrieval. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-99)*, pages 3017–3020, Phoenix, Arizona, USA.
- [Denoeux, 1995] Denoeux, T. (1995). A k -Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory. *IEEE Transactions on Systems, Man and Cybernetics*, 25(5), 804–813.
- [Diday, 1971] Diday, E. (1971). La méthode des nuées dynamiques. *Revue de Statistique appliquée*, 19(2), 19–34.
- [Djeraba, 2003] Djeraba, C. (2003). Association and Content-Based Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 15(1), 118–135.

- [Dy *et al.*, 2003] Dy, J., Brodley, C., Kak, A. C., Broderick, L., et Aisen, A. (2003). Unsupervised Feature Selection Applied to Content-Based Retrieval of Lung Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3), 373–378.
- [Eidenberger, 2004] Eidenberger, H. (2004). Statistical Analysis of Content-Based MPEG-7 Descriptors for Image Retrieval. *Multimedia Systems*, 10(2), 84–97.
- [Fagin *et al.*, 2003] Fagin, R., Kumar, R., et Sivakumar, D. (2003). Efficient Similarity Search and Classification via Rank Aggregation. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 301–312, San Diego, California, USA.
- [Fayyad *et al.*, 1996a] Fayyad, U., Djorgovski, S. G., et Weir, N. (1996a). Automating the Analysis and Cataloguing of Sky Surveys. In *Advances in Knowledge Discovery and Data Mining*, pages 471–493. AAAI/MIT Press.
- [Fayyad *et al.*, 1996b] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., et Uthurusamy, R. (1996b). *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press.
- [Fernandez *et al.*, 2002] Fernandez, G., Meckaouche, A., Peter, P., et Djeraba, C. (2002). Intelligent Image Clustering. *Lecture Notes in Computer Science*, 2490, 406–419.
- [Fisher, 1987] Fisher, D. H. (1987). Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning*, 2(2), 139–172.
- [Forgy, 1965] Forgy, E. W. (1965). Cluster Analysis of Multivariate Data : Efficiency versus Interpretability of Classifications. *Biometrics*, 21, 768–780.
- [Friedman *et al.*, 1977] Friedman, J. H., Bentley, J. L., et Finkel, R. A. (1977). An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software*, 3(3), 209–226.
- [Gibson et Nickerson, 1940] Gibson, K. S. et Nickerson, D. (1940). Analysis of the Munsell Color System. *Journal of the Optical Society of America*, 30, 591–608.
- [Gionis *et al.*, 1999] Gionis, A., Indyk, P., et Motwani, R. (1999). Similarity Search in High Dimensions via Hashing. In *Proceedings of 25th International Conference on Very Large Databases (VLDB'99)*, pages 518–529, Edinburgh, Scotland, UK.
- [Goldstein et Ramakrishnan, 2000] Goldstein, J. et Ramakrishnan, R. (2000). Contrast Plots and P-Sphere Trees : Space vs. Time in Nearest Neighbour Searches. In *Proceedings of the 26th International Conference on Very Large Databases (VLDB'00)*, pages 429–440, Cairo, Egypt.
- [Gounaris *et al.*, 2002] Gounaris, A., Paton, N., Fernandes, A., et Sakellariou, R. (2002). Adaptive Query Processing. In *Proceedings of the 19th British National Conference on Databases*, pages 11–25, Sheffield, UK.
- [Grosky *et al.*, 1992] Grosky, W. I., Neo, P., et Mehrotra, R. (1992). A Pictorial Index Mechanism for Model-Based Matching. *Data and Knowledge Engineering*, 8(4), 309–327.

- [Guttman, 1984] Guttman, A. (1984). R-trees : A Dynamic Index Structure for Spatial Searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, pages 47–57, Boston, Massachusetts, USA.
- [Haddon et Boyce, 1993] Haddon, J. F. et Boyce, J. F. (1993). Co-occurrence Matrices for Image Analysis. *IEEE Electronics and Communication Engineering Journal*, 5(2), 71–83.
- [Haralick *et al.*, 1973] Haralick, R. M., Shanmugam, K., et Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transaction on System, Man, Cybernetics*, 3, 610–621.
- [Henrich, 1998] Henrich, A. (1998). The LSD^h-Tree : An Access Structure for Feature Vectors. In *Proceedings of the 14th International Conference on Data Engineering (ICDE '98)*, pages 362–369, Orlando, Florida, USA.
- [Henrich *et al.*, 1989] Henrich, A., Six, H.-W., et Widmayer, P. (1989). The LSD Tree : Spatial Access to Multidimensional and non-Point Objects. In *Proceedings of the 15th International Conference on Very Large Databases (VLDB'89)*, pages 45–53, Amsterdam, Netherlands.
- [Hjaltason et Samet, 1995] Hjaltason, G. R. et Samet, H. (1995). Ranking in Spatial Databases. In *Proceedings of the 4th International Symposium on Spatial Databases*, pages 83–95, Portland, Maine, USA.
- [Hsu *et al.*, 2003] Hsu, W., Dai, J., et Lee, M. L. (2003). Mining Viewpoint Patterns in Image Databases. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pages 553–558, Washington, District of Columbia, USA.
- [Huang *et al.*, 1998] Huang, J., Kumar, S. R., et Zabih, R. (1998). An Automatic Hierarchical Image Classification Scheme. In *Proceedings of the 6th ACM International Conference on Multimedia (MULTIMEDIA '98)*, pages 219–228, Bristol, UK.
- [Indyk et Motwani, 1998] Indyk, P. et Motwani, R. (1998). Approximate Nearest Neighbors : towards Removing the Curse of Dimensionality. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, pages 604–613, Dallas, Texas, USA.
- [Judd et Wysecki, 1975] Judd, D. B. et Wysecki, G. (1975). *Color in Business, Science and Industry*. John Wiley & Sons.
- [Jähne, 1997] Jähne, B. (1997). *Digital Image Processing - Concepts, Algorithms, and Scientific Applications*. Springer-Verlag.
- [Kantor, 1994] Kantor, P. (1994). Information Retrieval Techniques. *Annual Review of Information Science and Technology*, 29(2), 53–90.
- [Katayama et Satoh, 1997] Katayama, N. et Satoh, S. (1997). The SR-tree : An Index Structure for High-Dimensional Nearest Neighbor Queries. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, pages 360–389, Tucson, Arizona, USA.

- [Kelly *et al.*, 1995] Kelly, P. M., Cannon, T. M., et Hush, D. R. (1995). Query by Image Example : the CANDID Approach. *SPIE Storage and Retrieval for Image and Video Databases III*, 2420, 238–248.
- [Kim et Kim, 2000] Kim, W.-Y. et Kim, Y.-S. (2000). A Region-Based Shape Descriptor using Zernike Moments. *Signal Process : Image Communication*, 16, 95–102.
- [Kiranyaz et Gabbouj, 2004] Kiranyaz, S. et Gabbouj, M. (2004). A Novel Multimedia Retrieval Technique : Progressive Query (why wait ?). In *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services*, Lisbon, Portugal.
- [Kourie et Oosthuizen, 1998] Kourie, D. et Oosthuizen, G. (1998). Lattices in Machine Learning : Complexity Issues. *Acta Informatica*, 35(4), 269–292.
- [Kuhn, 1955] Kuhn, H. W. (1955). The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quart.*, 2, 83–97.
- [Lang et Singh, 2002] Lang, C. et Singh, A. (2002). Accelerating High-Dimensional Nearest Neighbor Queries. In *Proceedings of the 14th International Conference on Scientific and Statistical Database Management (SSDBM 02)*, pages 109–118, Edinburgh, UK.
- [Lebart *et al.*, 1984] Lebart, L., Morineau, A., et Warwick, K. (1984). *Multivariate Descriptive Statistic Analysis : Correspondence Analysis and Related Techniques for Large Matrices*. John Wiley & Sons.
- [Li *et al.*, 2002] Li, C., Chang, E., Garcia-Molina, H., et Wiederhold, G. (2002). Clustering for Approximate Similarity Search in High-Dimensional Spaces. *IEEE Transactions on Knowledge and Data Engineering*, 14(4), 792–808.
- [Lippmann, 1987] Lippmann, R. P. (1987). An Introduction to Computing with Neural Networks. *IEEE Acoustics, Speech and Signal Processing Magazine*, 4(2), 4–22.
- [Lu, 1999] Lu, G. (1999). *Multimedia Database Management Systems*. Artech House.
- [Lu *et al.*, 1995] Lu, H., Setiono, R., et Liu, H. (1995). NeuroRule : a Connectionist Approach to Data Mining. In *Proceedings of the 21st Intl. Conference on Very Large Databases (VLDB'95)*, pages 478–489, Zurich, Switzerland.
- [Manjunath et Ma, 1996] Manjunath, B. S. et Ma, W. Y. (1996). Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8), 837–842.
- [Manjunath *et al.*, 2002] Manjunath, B. S., Salembier, P., et Sikora, T. (2002). *Introduction to MPEG-7*. John Wiley & Sons.
- [Manolescu, 2002] Manolescu, I. (2002). Adaptive and Self-tuning Query Processing. Tutorial at the Extending Database Technology Summer School, Cargèse, France.
- [Minkowsky, 1896] Minkowsky (1896). *Die Geometrie der Zahlen*. Teubner, Liebig.
- [Munsell, 1915] Munsell, A. H. (1915). *An Atlas of the Munsell System*. Malden, MA : Wadsworth-Howland.

- [Nakache et Confais, 2005] Nakache, J.-P. et Confais, J. (2005). *Approche pragmatique de la classification : arbres hiérarchiques, partitionnements*. Éditions Technip.
- [Nakamura *et al.*, 1993] Nakamura, Y., Abe, S., Ohsawa, Y., et Sakauchi, M. (1993). A Balanced Hierarchical Data Structure for Multidimensional Data with Highly Efficient Dynamic Characteristics. *IEEE Transactions on Knowledge and Data Engineering*, 5(4), 682–694.
- [Nastar *et al.*, 1998] Nastar, C., Mitschke, M., Meilhac, C., et Boujemaa, N. (1998). Surfimage : a Flexible Content-Based Image Retrieval System. In *Proceedings of the 6th ACM International Conference on Multimedia (MULTIMEDIA '98)*, pages 339–344, Bristol, UK.
- [Niblack *et al.*, 1993] Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E., Petkovic, D., Yanker, P., Faloutsos, C., et Taubin, G. (1993). The QBIC Project : Querying Images by Content using Color, Texture, and Shape. In *Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases*, San Jose, California, USA.
- [Nievergelt *et al.*, 1984] Nievergelt, J., Hinterberger, H., et Sevcik, K.-C. (1984). The GridFile : An Adaptable, Symmetric Multikey File Structure. *ACM Transactions on Database Systems*, 9(1), 38–71.
- [Obeid *et al.*, 2001] Obeid, M., Jedynak, B., et Daoudi, M. (2001). Image Indexing & Retrieval using Intermediate Features. In *Proceedings of the 9th ACM International Conference on Multimedia*, pages 531–533, Ottawa, Canada.
- [Oosthuizen, 1991] Oosthuizen, D. (1991). Lattice-Based Knowledge Discovery. In *Proceedings of AAAI'91 Knowledge Discovery in Databases Workshop*, pages 221–235, Anaheim, California, USA.
- [Ordóñez et Omiecinski, 1999] Ordóñez, C. et Omiecinski, E. (1999). Discovering Association Rules Based on Image Content. In *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries (ADL '99)*, pages 38–49, Baltimore, Maryland, USA.
- [Pasquier *et al.*, 1999] Pasquier, N., Bastide, Y., Taouil, R., et Lakhal, L. (1999). Efficient Mining of Association Rules using Closed Itemset Lattices. *Information Systems*, 24(1), 25–46.
- [Pei *et al.*, 2001] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., et Hsu, M. C. (2001). PrefixSpan : Mining Sequential Patterns Efficiently by Prefix Projected Pattern Growth. In *Proceedings of the 17th International Conference on Data Engineering (ICDE'01)*, pages 215–224, Heidelberg, Germany.
- [Pentland *et al.*, 1996] Pentland, A., Picard, R. W., et Sclaroff, S. (1996). Photobook : Content-Based Manipulation of Image Databases. *International Journal of Computer Vision*, 18(3), 233–254.
- [Quinlan, 1986] Quinlan, J. (1986). Induction of Decisions Trees. *Machine Learning*, 1, 81–106.

- [Quinlan, 1993] Quinlan, J. (1993). *C4.5 : Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc., Los Altos, California, USA.
- [Rijsbergen, 1979] Rijsbergen, C. V. (1979). *Information Retrieval*. Butterworth.
- [Robinson, 1981] Robinson, J. T. (1981). The K-D-B-tree : a Search Structure for Large Multidimensional Dynamic Indexes. In *Proceedings of the 1981 ACM SIGMOD International Conference on Management of Data*, pages 10–18, Ann Arbor, Michigan, USA.
- [Roussopoulos *et al.*, 1995] Roussopoulos, N., Kelley, S., et Vincent, F. (1995). Nearest Neighbor Queries. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 71–79, San Jose, California, USA.
- [Rui *et al.*, 1996] Rui, Y., She, A. C., et Huang, T. S. (1996). Modified Fourier Descriptors for Shape Representation - A Practical Approach. In *Proceedings of the 1st International Workshop on Image Databases and Multi Media Search*, Amsterdam, Netherlands.
- [Samet, 1984] Samet, H. (1984). The Quadtree and Related Hierarchical Data Structures. *ACM Computing Surveys*, 16(2), 187–260.
- [Santini et Jain, 1995] Santini, S. et Jain, R. (1995). Similarity Matching. In *Proceedings of the 2nd Asian Conference on Computer Vision (ACCV '95)*, pages 571–580, Singapore.
- [Sarle, 1983] Sarle, W. S. (1983). Cubic Clustering Criterion. SAS Technical Report P-256, NC, SAS Institute.
- [Scott et Thompson, 1983] Scott, D. et Thompson, J. (1983). Probability Density Estimation in Higher Dimensions. In *Proceedings of the 15th Symposium on the Interface*, pages 173–179, North Holland-Elsevier, Amsterdam, New York, Oxford.
- [Seeger et Kriegel, 1990] Seeger, B. et Kriegel, H.-P. (1990). The Buddy-Tree : An Efficient and Robust Access Method for Spatial Data Base Systems. In *Proceedings of the 16th International Conference on Very Large Databases (VLDB'90)*, pages 590–601, Brisbane, Queensland, Australia.
- [Sellis *et al.*, 1987] Sellis, T. K., Roussopoulos, N., et Faloutsos, C. (1987). The R⁺-Tree : A Dynamic Index for Multi-Dimensional Objects. In *Proceedings of the 13th International Conference on Very Large Databases (VLDB'87)*, pages 507–518, Brighton, England.
- [Shusterman et Feder, 1994] Shusterman, E. et Feder, M. (1994). Image Compression via Improved Quadtree Decomposition Algorithms. *IEEE Transactions on Image Processing*, 3(2), 207–215.
- [Shyu *et al.*, 1999] Shyu, C.-R., Brodley, C. E., Kak, A. C., Kosaka, A., Aisen, A., et Broderick, L. (1999). Assert : A Physician-in-the-Loop Content-Based Retrieval System for HRCT Image Databases. *Computer Vision and Image Understanding*, 75(1/2), 111–132.
- [Smeulders *et al.*, 2000] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., et Jain, R. (2000). Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), 1349–1380.

- [Smith et Chang, 1997] Smith, J. R. et Chang, S.-F. (1997). Querying by Color Regions using the VisualSEEK Content-Based Visual Query System. In *Intelligent Multimedia Information Retrieval*. MIT Press.
- [Srikant et Agrawal, 1996] Srikant, R. et Agrawal, R. (1996). Mining Sequential Patterns : Generalizations and Performance Improvements. In *Proceedings of the 5th International Conference on Extending Database Technology (EDBT'96)*, volume 1057, pages 3–17. Springer-Verlag.
- [Stehling et al., 2003] Stehling, R. O., Nascimento, M. A., et Falcão, A. X. (2003). Techniques for Color-Based Image Retrieval. In *Multimedia Mining - a High Way to Intelligent Multimedia Document*, chapter 4, pages 61–80. Kluwer Academic.
- [Stricker et Orengo, 1995] Stricker, M. et Orengo, M. (1995). Similarity of Color Images. *SPIE - Storage and Retrieval for Image and Video Databases III*, 2420, 381–392.
- [Su et al., 2001] Su, Z., Li, S., et Zhang, H. (2001). Extraction of Feature Subspaces for Content-Based Retrieval using Relevance Feedback. In *Proceedings of the 9th ACM International Conference on Multimedia (MULTIMEDIA '01)*, pages 98–106, Ottawa, Canada. ACM Press.
- [Tao et Grosky, 1999] Tao, Y. et Grosky, W. (1999). Object-Based Image Retrieval using Point Feature Maps. In *Proceedings of the International Conference on Database Semantics (DS-8)*, pages 59–73.
- [Torgeson, 1965] Torgeson, W. S. (1965). Multidimensional Scaling of Similarity. *Psychometrika*, 30, 379–393.
- [Tversky, 1977] Tversky, A. (1977). Feature of Similarity. *Psychological Review*, 84(4), 327–352.
- [Wang et Li, 2002] Wang, J. Z. et Li, J. (2002). Learning-Based Linguistic Indexing of Pictures with 2-d MHMMs. In *Proceedings of the 10th ACM International Conference on Multimedia (MULTIMEDIA '02)*, pages 436–445, Juan-les-Pins, France.
- [Wang, 2001] Wang, S. (2001). A Robust CBIR Approach using Local Color Histograms.
- [Weber et Böhm, 2000] Weber, R. et Böhm, K. (2000). Trading Quality for Time with Nearest Neighbor Search. In *Proceedings of the 7th International Conference on Extending Database Technology (EDBT '00)*, pages 21–35, Konstanz, Germany.
- [Weber et al., 1998] Weber, R., Schek, H., et Blott, S. (1998). A Quantitative Analysis of Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proceedings of the 24th International Conference on Very Large Databases (VLDB'98)*, pages 194–205, New-York City, New-York, USA.
- [White et Jain, 1996] White, D. A. et Jain, R. (1996). Similarity Indexing with the SS-tree. In *Proceedings of the 12th International Conference on Data Engineering (ICDE'96)*, pages 512–523, New Orleans, Louisiana, USA.
- [Wu et Bretschneider, 2004] Wu, L. et Bretschneider, T. (2004). Comparative Analysis of the Efficiency of R-Tree Based Indexing Strategies for Information Retrieval. In

- Proceedings of the International Conference on Information and Knowledge Engineering (IKE 2004)*, pages 180–184, Las Vegas, Nevada, USA.
- [Wu et Manjunath, 2001] Wu, P. et Manjunath, B. S. (2001). Adaptive Nearest Neighbor Search for Relevance Feedback in Large Image Databases. In *Proceedings of the 9th ACM International Conference on Multimedia (MULTIMEDIA '01)*, pages 89–97, Ottawa, Canada.
- [Youille *et al.*, 1991] Youille, A. L., Honda, K., et Peterson, C. (1991). Particle Tracking by Deformable Templates. In *Proceedings of the International Joint Conference on Neural Networks*, pages 7–11, Seattle, Washington, USA.
- [Zaharia et Preteux, 2001] Zaharia, T. et Preteux, F. (2001). 3D Shape-Based Retrieval within the MPEG-7 Framework. In *Proceedings of SPIE Conference 4304 on Nonlinear Image Processing and Pattern Analysis XII*, pages 133η–145, San Jose, California, USA.
- [Zaki, 1998] Zaki, M. J. (1998). Efficient Enumeration of Frequent Sequences. In *Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM '98)*, pages 68–75, Bethesda, Maryland, USA.
- [Zařane *et al.*, 2000] Zařane, O., Han, J., et Zhu, H. (2000). Mining Recurrent Items in Multimedia with Progressive Resolution Refinement. In *Proceedings of the 16th IEEE International Conference on Data Engineering (ICDE'00)*, pages 461–476, San Diego, California, USA.
- [Zhang *et al.*, 2001] Zhang, J., Hsu, W., et Lee, M. L. (2001). Image Mining : Issues, Frameworks and Techniques. In *Proceedings of the 2nd International Workshop on Multimedia Data Mining (MDM/KDD 2001)*, pages 13–20, San Francisco, California, USA.
- [Zhang *et al.*, 1996] Zhang, T., Ramakrishnan, R., et Livny, M. (1996). BIRCH : an Efficient Data Clustering Method for Very Large Databases. *ACM SIGMOD Record*, 25(2), 103–114.
- [Zhang *et al.*, 1998] Zhang, Y. J., Liu, Z. W., et He, Y. (1998). Comparison and Improvement of Color-Based Image Retrieval Techniques. *SPIE - Storage and Retrieval for Image and Video Databases IV*, 3312, 371–382.

Liste des notations

rap	Rappel	15
$prec$	Précision	15
$supp$	Support d'une règle d'association	41
$conf$	Confiance d'une règle d'association	41
$minsupp$	Seuil de support d'une règle d'association	41
$minconf$	Seuil de confiance d'une règle d'association	41
r	Règle d'association	50
B	Base d'images fixes	58
D	Ensemble des types de descripteurs globaux d'images fixes	58
m	Nombre de types de descripteurs	58
d_i	Type de descripteur	58
$d_i(I)$	Vecteur de descripteur de type d_i pour l'image I	58
C_{ij}	Cluster j du type de descripteur i	58
$n_c(d_i)$	Nombre total de clusters du type de descripteur d_i	58
$card(X)$	Nombre d'éléments d'un ensemble quelconque X	58
I_q	Image-requête par l'exemple	58
n_q	Nombre de points dans l'image-requête I_q	58
O_q	Ensemble de points de l'image-requête I_q	58
W_q	Ensemble de poids affectés aux points de O_q	58
D_q	Fonction de distance entre deux points	58
S	Espace multidimensionnel	58
d_S	Dimension de l'espace multidimensionnel S	58
$\mu_q^{(j)}$	Poids associé à la j^e dimension de l'espace S	58
$D_q(I_q, O)$	Distance entre l'image-requête I_q et un point quelconque O de S ..	59
C_S	Ensemble des clusters les plus proches de I_q	59
R	Ensemble des règles d'association calculées hors-ligne	59
C_r	Ensemble des clusters dans l'expression de la règle d'association r ..	59
T_0	Instant de lancement d'une requête progressive	61
T_i	Instant intermédiaire d'exécution d'une requête progressive	61
T_f	Instant final d'exécution d'une requête progressive	61
$\vec{Q}(I_q, T_0, T_f)$	Requête progressive exécutée entre les instants T_0 et T_f	61
$RI(I_q, T_i)$	Résultat intermédiaire d'une requête progressive à l'instant T_i ...	61
T_{stop}	Instant d'arrêt de l'exécution d'une requête progressive	63
P	Plan de requête	63

$P_i(I_q, d_i)$	Sous-plan de traitement des <i>clusters</i> du type de descripteur d_i ...	63
\circ	Opérateur de fusion ou d'agrégation des résultats	63
$Dep_{(k)}(I_q, C_{ij})$	k^e caractéristique de C_{ij} dépendante de I_q	65
n_{dep}	Nombre total de caractéristiques dépendantes de I_q	65
$Indep_{(k)}(C_{ij})$	k^e caractéristique de C_{ij} indépendante de I_q	65
n_{indep}	Nombre total de caractéristiques indépendantes de I_q	65
$NormX$	Valeur normalisée de la caractéristique X	65
$eligibility(I_q, C_{ij})$	Éligibilité du <i>cluster</i> C_{ij} pour la requête I_q	65
$ExtrNormX$	Valeur Extrémale de $NormX$	65
α_k	Poids affecté à la caractéristique k	65
$TopC(M, j)$	Ensemble des M <i>clusters</i> de meilleure éligibilité à la phase j	66
R_{ij}	Rayon du <i>cluster</i> C_{ij}	67
$d_{min}(I_q, C_{ij})$	Distance minimum entre une image-requête I_q et un <i>cluster</i> C_{ij} ..	67
$Dist(I_q, C_{ij})$	Distance entre le centroïde de C_{ij} et la requête I_q	67
$Rule_i$	Règle de filtrage numéro i	67
$d(I_q, ppv_k)$	Distance entre I_q et le $k^{ième}$ plus proche voisin courant	67
l_i	Liste ordonnée d'images, résultats de la recherche suivant d_i	68
$S_f(I)$	Score de fusion ou fonction d'agrégation	68
λ_i	Poids affecté au type de descripteur d_i	68
$dim(d_i)$	Dimension d'un vecteur du type de descripteur d_i	81
$Cost(C_{ij})$	Coût de parcours du <i>cluster</i> C_{ij}	81

Liste des définitions

1.1	Rappel et précision	15
3.1	Descripteur visuel	57
3.2	Base d'images fixes	58
3.3	Requête par l'exemple	58
3.4	Sous-requête	59
3.5	Requête progressive	61
3.6	Résultat intermédiaire	61
3.7	Plan de requête	63
3.8	Opérateur de fusion	64
3.9	Phase élémentaire de recherche	64

Liste des figures

1	Étapes du processus d'ECBD	7
2.1	Une hiérarchie binaire indicée	45
2.2	Processus d'organisation des images fixes	52
3.1	Scénario de recherche progressive par le contenu	57
3.2	Sélection des règles d'association	60
3.3	Recherche progressive par le contenu	62
4.1	Variation du nombre de règles en fonction du nombre de <i>clusters</i> sur B_1	74
4.2	Diagramme à bâtons du nombre de règles en fonction du nombre de <i>clusters</i> sur B_1	74
4.3	Règles découvertes avec <i>Apriori</i> sur la base B_1	74
4.4	Choix du nombre et du support des règles sur B_2	75
4.5	Estimation topologique des relations entre les types de descripteurs . . .	77
4.6	Diagramme d'utilisation des règles sur la base de 7727 images (base B_1)	78
4.7	Variation de la qualité relative des résultats en fonction du nombre de <i>clusters</i> parcourus sur B_2	80
4.8	Variation du temps de recherche en fonction du nombre de <i>clusters</i> parcourus sur B_2	80
4.9	Étude de la qualité des résultats intermédiaires en fonction de l'éligibilité des <i>clusters</i> sur B_3	83
4.10	Étude des temps intermédiaires de recherche en fonction de l'éligibilité des <i>clusters</i> sur B_3	83
4.11	Étude des résultats intermédiaires en fonction de l'éligibilité des <i>clusters</i> pour 100 <i>clusters</i> par type de descripteur sur B_3	86
4.12	Comparaison des règles de filtrage sur la base du nombre moyen de plus proches voisins trouvés (base B_3)	88
4.13	Comparaison des règles de filtrage sur la base du temps moyen de recherche (base B_3)	88
4.14	Comparaison des règles de filtrage : utilisation du <i>M-Tree</i> pour le calcul des <i>clusters</i>	90
4.15	Influence de la taille de la base d'images sur la qualité et le temps de la recherche : utilisation du <i>k-means</i> pour le calcul des <i>clusters</i> sur B_3 . . .	91

4.16	Influence de la taille de la base d'images sur la qualité et le temps de la recherche : utilisation du <i>M-Tree</i> pour le calcul des <i>clusters</i> sur B_3 . . .	91
A.1	Description MPEG-7 d'un document multimédia	100
A.2	Exemple d'image à décrire avec <i>MPEG-7</i>	101
A.3	Exemple de descripteur <i>MPEG-7</i> de type <i>Color Layout</i>	102
A.4	Exemple de descripteur <i>MPEG-7</i> de type <i>Scalable Color</i>	103
A.5	Canaux utilisés dans l'extraction du descripteur HTD	104
A.6	Exemple de descripteur <i>MPEG-7</i> de type <i>Homogeneous Texture</i>	105
A.7	Sémantique de l'histogramme de partition de EHD	105
A.8	Exemple de descripteur <i>MPEG-7</i> de type <i>Edge Histogram</i>	106
A.9	Exemple de descripteur <i>MPEG-7</i> de type <i>Region-based Shape</i>	107

Liste des tableaux

3.1	Rappel des notations utilisées	60
4.1	Types de descripteurs utilisés	72
4.2	Paramètres déterminés expérimentalement	76
4.3	Paramètres à l'organisation des images et règles découvertes	76
4.4	Règles produites par l'ACM	77
4.5	Nombre moyen de <i>clusters</i> parcourus pour avoir le résultat séquentiel final : cas de 10 <i>clusters</i> par type de descripteur (base B_3)	84
4.6	Nombre moyen de <i>clusters</i> parcourus pour avoir le résultat séquentiel final : cas de 50 <i>clusters</i> par type de descripteur (base B_3)	84
4.7	Nombre moyen de <i>clusters</i> parcourus pour avoir le résultat séquentiel final : cas de 100 <i>clusters</i> par type de descripteur (base B_3)	84
4.8	Nombre de <i>clusters</i> parcourus par type de descripteur pour la règle ($Rule_1$) sur B_3	87
4.9	Nombre de <i>clusters</i> parcourus par type de descripteur pour la règle ($Rule_2$) sur B_3	87
4.10	Nombre moyen de <i>clusters</i> parcourus par type de descripteur en séquentiel, puis avec chacune des règles ($Rule_1$) et ($Rule_2$) sur B_3	89
4.11	F - <i>measure</i> par type de descripteur	93
4.12	F - <i>measure</i> du système avec combinaison de plusieurs types de descripteurs	93

Références de l'auteur

Revues nationales avec comité de lecture

Kouomou Choupo A., Berti-Équille L., Morin A. Indexation et recherche par le contenu dans une base d'images fixes : l'intérêt des règles d'association. *Revue RNTI-E-4 Fouille de Données Complexes, CÉPADUÈS*, 2005.

Kouomou Choupo A., Morin A., Berti-Équille L. Recherche dans de grandes bases d'images fixes : une nouvelle approche guidée par les règles d'association. *Revue RNTI-E-2 (Actes EGC'2004), CÉPADUÈS*, pages 65-70, 2004.

Conférences internationales avec comité de lecture et édition d'actes

Morin A., Kouomou Choupo A., Chauchat J.-H. Dimension Reduction and Clustering for Query-by-Example in Huge Image Databases. *Proceedings of the 3rd World Conference on Computational Statistics and Data Analysis*, Limassol, Chypre. Octobre 2005.

Kouomou Choupo A., Berti-Équille L., Morin A. Multimedia Indexing and Retrieval with Features Association Rules Mining. *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME'04)*, Taipei, Taiwan. Juin 2004.

Workshops et symposium internationaux avec comité de lecture et édition d'actes

Kouomou Choupo A., Berti-Équille L., Morin A. Optimizing Progressive Query-by-Example over Pre-Clustered Large Image Databases. *Proceedings of the 2nd ACM SIGMOD International Workshop on Computer Vision meets DataBases (CVDB'05)*, Baltimore, Maryland, USA. Juin 2005.

Kouomou Choupo A., Berti-Équille L. Visual Feature Mining for Adapting Query-by-Example over Large Image Databases. *Proceedings of the International Workshop on Multidisciplinary, Video, and Audio Retrieval and Mining*, Sherbrooke, Canada. Octobre 2004.

Berti-Équille L., Kouomou Choupo A., Morin A. Feature Mining for Multimedia Indexing and Retrieval. *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'04)*, Lisbon, Portugal. Avril 2004.

Morin A., Kouomou Choupo A., Berti-Équille L. Research in Image Databases : How Statistical Data Analysis Methods can Enrich the Association Rules. *Proceedings of the 16th Symposium of International Association for Statistical Computing (COMPSTAT'04)*, Prague, République Tchèque. Août 2004.

Conférences nationales avec comité de lecture et édition d'actes

Kouomou Choupo A., Berti-Équille L., Morin A. Optimizing Progressive Query-by-Example over Pre-Clustered Large Image Databases. *Actes des 21^e journées Bases de Données Avancées (BDA'05)*, Saint-Malo, France. Octobre 2005.

Kouomou Choupo A. Sélection des critères de recherche par le contenu dans une base d'images : l'apport des règles d'associations entre les descripteurs visuels. *Actes du 22^e congrès francophone INFORSID (INFORSID'04)*, Biarritz, France. Mai 2004.

Résumé

Les images fixes peuvent, entre autre, être décrites au niveau pixel par des descripteurs visuels globaux de couleur, de texture ou de forme. La recherche par le contenu exploite et combine alors ces descripteurs dont le coût de calcul est d'autant plus important que la taille de la base d'images est grande. Les résultats de la recherche sont ensuite classés en fonction de leur similarité à la requête soumise et présentés à l'utilisateur sous forme de liste ordonnée. Un sous-ensemble de descripteurs pourrait cependant suffire à répondre à une recherche par similarité beaucoup plus rapidement, tout en gardant une qualité acceptable des résultats de recherche. Nous proposons pour cela une méthode de sélection automatique des descripteurs visuels qui exploite les règles d'association pour élaborer des stratégies d'exécution réduisant le temps de la recherche par le contenu dans de grandes bases d'images fixes. Dans cette thèse, nous présentons également comment une recherche par le contenu peut être adaptée pour proposer des résultats intermédiaires qui sont fusionnés de façon progressive avec l'avantage pour l'utilisateur, d'une part, de ne pas attendre que toute la base ait été parcourue avant de fournir un résultat et, d'autre part, de lui permettre de stopper la requête en cours d'exécution. Les expérimentations conduites sur des bases d'images réelles montrent que notre méthode améliore notablement les temps de réponse. Elles confirment aussi l'intérêt de la combinaison des descripteurs globaux pour la recherche d'images par le contenu.

Mots-clés : règles d'association, recherche progressive d'images par le contenu, combinaison des descripteurs globaux.

Abstract

Still images can be retrieved by similarity searching on global visual features such as color, texture, or shape at the pixel level. Content-based retrieval systems then use and combine all the available low-level features whose computing cost can be prohibitive and they rank the images according to how well they match the submitted query-by-example. Finally, they return the best few matches to the user in a ranked result list. But, a subset of features could be sufficient enough to answer very quickly while offering an acceptable quality of results. Moreover, the administration of large collections of images accentuates the classical problems of indexing and efficiently querying information. Our work focuses on the elaboration of fully automatic and generic strategies of visual global features usage which exploit association rules to speed up the content-based retrieval on large still image databases. In this thesis, we also present how query-by-example processing is adapted to propose intermediate results that are progressively merged together with the advantage, for the users, on one hand, not to wait until the whole database has been processed and, on the other hand, to allow them to stop the current execution of the query without losing the first partial results. Experiments performed on real image sets show that our method improves retrieval times. They also confirm the interest of global features combination for content-based image retrieval.

Keywords : association rules, progressive content-based image retrieval, combination of global features.