

**MINISTRE DE L'EDUCATION ET DE LA FORMATION
INSTITUT POLYTECHNIQUE DE HANOI**

LÊ THỊ LAN

**INDEXATION ET RECHERCHE D'IMAGES
PAR LE CONTENU**

Spécialité : Traitement d'Information et Communication

MÉMOIRE DE MASTER

SOUS LA DIRECTION DE

Dr. Alain BOUCHER

HANOI 2005

Remerciements

Je tiens tout d'abord à remercier Alain Boucher, professeur à l'Institut de la Francophonie pour l'Informatique (IFI), responsable du groupe de recherche en traitement d'images à MICA, qui m'a donné ce sujet et m'a encadré pendant mon Master. Il a toujours été à mon écoute et son point de vue complémentaire est souvent été très utile.

Je souhaite également exprimer ma gratitude à Nguyen Trong Giang, directeur du Centre MICA, directeur du centre doctoral de l'Institut Polytechnique de Hanoi, Pham Thi Ngoc Yen, directeur adjoint au Centre MICA, Eric Castelli, directeur adjoint du Centre MICA qui m'ont accepté dans leur centre de recherche et m'ont donné beaucoup d'aides.

Je tiens à remercier mes professeurs dans la faculté de la Technologie de l'Information de l'Institut Polytechnique de Hanoi. Ils m'ont donné des connaissances fondamentales qui sont très utiles pour poursuivre mes études.

Merci également à tous mes collègues du Centre MICA et mes amis à l'IFI et dans ma classe de Master, Dang Thanh Ha pour sa coopération pendant 3 mois, Do Thi Thanh Hien pour les démarches administratives, et les autres pour leur soutien.

Je voudrais remercier le projet Orion de l'INRIA pour avoir permis l'utilisation de la base de Pollen et pour les discussions sur l'ontologie.

Merci également aux chercheurs, aux professeurs dans le monde entier qui ont donné leurs articles, leurs bases d'images disponibles sur l'Internet librement. Enfin je remercie mes proches, mes parents, mes sœurs, pour leur soutien et leur confiance tout au long de cette épreuve.

Résumé

Le travail présenté dans ce rapport est mon travail pendant mon Master au Centre MICA.

L'objectif de mon travail est d'étudier l'état de l'art des méthodes et des systèmes déjà existants à l'heure actuelle et de construire un système d'indexation et de recherche d'images par le contenu à partir de ces connaissances.

A MICA, nous construisons un système d'indexation et de recherche d'images par le contenu symbolique en héritant les techniques déjà existantes et en rajoutant nos idées. Nous orientons notre recherche vers la recherche d'images par le contenu sémantique. Donc nous étudions des voies de recherche d'images par le contenu sémantique, testons notre système avec le retour de pertinence. Avec le retour de pertinence, nous développons une interface traditionnelle et proposons une interface graphique 2D. Nous pensons que l'interface graphique 2D donne une façon flexible pour afficher des résultats et pour interagir avec l'utilisateur. Avec cette interface, une méthode de retour de pertinence avec des images positives a été appliquée.

Mots clés : indexation et recherche d'images par le contenu, retour de pertinence, interprétation sémantique.

Table des matières

Chapitre 1: Introduction	10
1.1. Contexte du travail.....	10
1.2. Objectifs.....	11
1.3. Contenu du rapport.....	11
Chapitre 2: Recherche d'images par le contenu symbolique	12
2.1. Etat de l'art	12
2.1.1. Architecture du système d'indexation et recherche d'images.....	13
2.1.2. Les requêtes	15
2.1.3. Caractéristiques symboliques	20
2.1.4. Mesures pour évaluer un système.....	28
2.2. Notre système de recherche par le contenu symbolique.....	34
2.2.1. Des caractéristiques choisises.....	34
2.2.2. Les bases d'images utilisées	40
2.2.3. Une image requête et plusieurs images requêtes	44
2.2.4. Nos résultats.....	47
Chapitre 3: Recherche d'images par le contenu sémantique	67
3.1. Etat de l'art	67
3.1.1. Retour de pertinence	68
3.1.2. Ontologie	72
3.2. Notre système de recherche d'images par le contenu sémantique	77
3.2.1. Le retour de pertinence avec l'interface traditionnelle.....	77

3.2.2. Le retour de pertinence avec notre interface	82
3.2.3. Nos résultats.....	87
Chapitre 4: Conclusion et perspectives.....	94
4.1. Conclusions	94
4.2. Perspectives	95
4.2.1. Améliorer notre système d'indexation et recherche par le contenu	95
4.2.2. La base d'images MICA	96
4.2.3. Caractérisation des bases d'images pour l'évaluation.....	99
4.2.4. Le projet IRIS	102
Annexe1 – Transformation d'une image RVB en une image TSV	105
Annexe2- Normaliser la distance – Gaussien	106
Publications.....	107
Bibliographie.....	108

Table des figures

Fig. 2-1: L'architecture d'un système d'indexation et recherche d'images par le contenu	14
Fig. 2-2 : Trois façons pour donner une requête dans CBIR (Deselaers, 2003)	15
Fig. 2-3 : Un exemple de recherche d'images dans Google	16
Fig. 2-4 : Une requête par esquisse dans QBIC (QBIC).....	17
Fig. 2-5 : Une requête par image dans Blobworld (Blobworld)	17
Fig. 2-6 : Une requête par régions.....	18
Fig. 2-7 : Un exemple quand l'utilisateur essaie de rechercher des images de voitures. La colonne de gauche affiche les résultats quand il n'y a qu'un exemple. La colonne de droite affiche les résultats quand il y a deux exemples (Nakazato, 2002)	20
Fig. 2-8 : Deux découpages différents de l'image. (a) Découpage en grille de l'image. (b) Découpage par la segmentation division-fusion de l'image.	25
Fig. 2-9 : Le rappel et la précision pour une requête (Yates, 1999)	30
Fig. 2-10 : Une courbe de rappel et précision	31
Fig. 2-11 : L'histogramme et l'histobin	35
Fig. 2-12 : 10 classes de la base de Wang (Deselaers, 2003)	41
Fig. 2-13 : Les objets utilisés dans COIL-100(Deselaers, 2003).....	42
Fig. 2-14 : Les objets utilisés dans COIL-20 (Deselaers, 2003).....	42
Fig. 2-15 : Quelques images exemples de la base de Pollens.....	43
Fig. 2-16 : Quelques images exemples dans la base de CUReT(CUReT)	44
Fig. 2-17 : Quelques images exemples dans la base de Fei-Fei	44
Fig. 2-18 : Notre interface développée pour le système d'indexation et recherche d'images par le contenu	48

Fig. 2-19 : Quelques résultats avec la base de Wang.....	49
Fig. 2-20 : Les courbes de rappel et précision avec la base de Wang avec les différentes caractéristiques.....	51
Fig. 2-21 : Les courbes de rappel et précision de chaque classe de la base de Wang avec les différentes caractéristiques	53
Fig. 2-22 : Quelques résultats avec la base Coil, des images en rouge sont des images non-pertinentes.....	54
Fig. 2-23 : Les courbes de rappel et précision avec la base de Coil 100 avec les différentes caractéristiques	56
Fig. 2-24 : Les courbes de rappel et précision pour la base de Pollens avec différentes caractéristiques	57
Fig. 2-25 : Quelques résultats avec la base de FeiFei, les images en rouge sont des images non-pertinentes	59
Fig. 2-26 : Les courbes de rappel et précision avec la base de Fei-Fei avec les différentes caractéristiques	61
Fig. 2-27 : Résultat avec une seule image requête, qui est grande et au milieu de la figure.....	62
Fig. 2-28 : Le résultat avec la méthode de combinaison par distance minimale en rajoutant une image avec les images requêtes indiquées par un point vert au milieu et montrées dans le coin inférieur gauche.....	63
Fig. 2-29 : Le résultat avec la méthode de combinaison linéaire en rajoutant une image, avec les images requêtes indiquées par un point vert au milieu et montrées dans le coin inférieur gauche	64
Fig. 2-30 : Le résultat avec la méthode de combinaison par distance minimale en rajoutant deux images, avec les images requêtes indiquées par un point vert au milieu et montrées dans le coin inférieur gauche.....	65

Fig. 2-31 : Le résultat avec la méthode de combinaison linéaire en rajoutant deux autres images, avec les images requêtes indiquées par un point vert au milieu et montrées dans le coin inférieur gauche.....	66
Fig. 3-1 : Un exemple de retour de pertinence dans le CBIR, on a répété le processus de retour de pertinence deux fois et après chaque fois, le nombre d'images retrouvées pertinentes a augmenté.	69
Fig. 3-2 : Trois réponses possibles de la question « What is a pipe ? » [Gandon02].....	74
Fig. 3-3 : L'interface traditionnelle qui fournit le retour de pertinence avec 3 niveaux : pertinente, neutre ou non-pertinente.....	79
Fig. 3-4 : Notre première interface graphique en 2D, l'image requête est au coin inférieur gauche, en rouge, les autres images ont des positions relatives à l'image requête en fonction des distances entre elles et l'image requête.	84
Fig. 3-5 : Un exemple de notre nouvelle interface, les images sont affichées en un plan avec deux axes : l'intersection d'histogrammes RVB et l'intersection d'histogrammes TSV.	85
Fig. 3-6 : L'utilisateur veut trouver des images qui contiennent un bus, l'image de gauche montre les seize premiers résultats, les images en rouge sont les images non-pertinentes, le reste est pertinent. L'image de droite montre les seize premiers résultats quand on fait une fois de retour de pertinence.....	88
Fig. 3-7 : L'utilisateur veut trouver des images qui contiennent un (des) cheval (chevaux). Avec le retour de pertinence, les résultats sont très bons. Parmi les seize images résultats, il y a seulement deux images non-pertinentes qui contiennent aussi des animaux (des éléphants).	89

Fig. 3-8 : Les courbes de rappel et précision de la base de Wang sans le retour de pertinence, avec un seul retour de pertinence et avec deux retours de pertinence	90
Fig. 3-9 : Les courbes de rappel et précision de la classe 3 de la base de Wang sans retour de pertinence, avec un seul retour de pertinence et avec deux retours de pertinence	91
Fig. 3-10 : L'image à gauche montre les résultats de recherche d'images sans retour de pertinence où la région en rouge indique l'image requête. L'image à droite montre les résultats de recherche d'images avec le retour de pertinence où la région en bleue indique les images positives que nous avons choisies.....	93
Fig. 4-1 : Quelques exemples dans notre base d'images	98
Fig. 4-2 : Interface de notre programme pour annoter une image, un arbre de mots clés (à gauche), image à annoter (à droite).....	98
Fig. 4-3 : Quelques images annotées avec ses mots clés choisis.....	99

Liste des tableaux

Tab 2-1 : Région de 15 pixels	37
Tab 2-2 : Matrice de cooccurrence avec $d = 1$ et $\alpha = 0$ (un pixel vers le bas) ..	38
Tab 2-3 : Les valeurs de rappel et précision de la base de Wang, avec un nombre d'images retrouvées est de 16.	50
Tab 2-4 : Les valeurs de rappel et précision de la base de Coil où le nombre d'images retrouvées est de 16.	55
Tab 2-5 : Les valeurs de rappel et précision de la base de FeiFei, où le nombre d'images retrouvées est de 16.	60

Chapitre 1: Introduction

1.1. Contexte du travail

Le domaine de l'image numérique est un domaine en pleine expansion. Depuis quelques années, avec l'explosion d'Internet et aussi le développement à grande échelle de la photographie numérique, il n'est pas rare d'avoir des bases d'images numériques contenant plusieurs milliers et même plusieurs dizaines de milliers d'images, que ce soit des bases ciblées pour un domaine d'activité professionnelle (journalisme, tourisme, éducation, musées, ...) ou tout simplement pour les particuliers qui accumulent d'immenses bases de photographies numériques (souvenirs, voyages, famille, événements, ...).

Pour gérer et utiliser efficacement ces bases d'images, un système d'indexation et de recherche d'images est nécessaire. C'est pourquoi le sujet de la recherche d'images devient un sujet très actif dans la communauté internationale depuis plus d'une dizaine d'années.

Au Vietnam ainsi qu'à MICA, il n'existe pas de système d'indexation et recherche d'images par le contenu. Les quelques systèmes de recherche d'images existants sont des systèmes de recherche d'images soit par le nom des images soit utilisant des annotations manuelles. Donc ils demandent beaucoup d'efforts et dans beaucoup de cas, ils ne satisfont pas des utilisateurs.

1.2. Objectifs

A partir de besoins d'un système d'indexation et recherche d'images au Vietnam, nous avons choisi le sujet : l'indexation et recherche d'images par le contenu.

Avec ce sujet, nos objectifs sont :

- d'étudier l'état de l'art des méthodes et des systèmes déjà existants à l'heure actuelle.
- de construire un système d'indexation et recherche d'images par le contenu. Le but est de trouver des caractéristiques permettant d'indexer l'image et de retrouver des images similaires lors d'une requête en fonction du contenu des images. En développant ce système, nous avons aussi développé une interface qui permet à l'utilisateur de faire la requête et de prendre les résultats ainsi que les mesures qui permettent d'évaluer notre système.
- d'aller plus loin, et d'étudier les idées pour un système d'indexation et recherche d'images par le contenu sémantique.

1.3. Contenu du rapport

Ce rapport est organisé comme suit. Le chapitre 2 introduit l'état de l'art du système d'indexation et recherche d'images par le contenu, notre système et quelques résultats. Le chapitre 3 se consacre au système d'indexation et recherche d'images par le contenu sémantique avec quelques résultats préliminaires. Nous terminons ce rapport en donnant quelques conclusions et perspectives de travail dans le chapitre 4.

Chapitre 2: Recherche d'images par le contenu symbolique

2.1. Etat de l'art

Nous avons déjà mentionné l'intérêt actuel porté sur les bases d'images, les techniques d'indexation et les techniques de recherche d'images dans de telles bases. Cependant, dans les applications pratiques de ces bases d'images que l'on retrouve aujourd'hui, que ce soit pour les chaînes de télévision, les journaux, les musées et même pour les moteurs de recherche sur Internet qui proposent des solutions de recherche d'images, l'indexation et la recherche de ces images se font en se basant majoritairement sur des annotations textuelles ou des éléments de texte qu'on peut rattacher aux images.

Dans plusieurs cas (chaînes de télévision, journaux,...), l'archivage des images et des séquences vidéos ne se fait qu'au prix d'une étape d'annotation manuelle à l'aide de mots-clés. Cette indexation représente une tâche longue et répétitive pour l'humain, surtout avec les bases d'images qui deviennent aujourd'hui de plus en plus grandes. De plus, cette tâche est très subjective à la culture, à la connaissance et aux sentiments de chaque personne.

Le besoin de méthodes d'indexation et de recherche directement basée sur le contenu de l'image n'est donc plus à démontrer. Le premier prototype de système a été proposé en 1970 et ce système a attiré l'attention de beaucoup de chercheurs. Quelques systèmes deviennent des systèmes commerciaux tels que QBIC (Query By Image Content), CIRES (Content Based Image REtrieval System), ...

Les systèmes d'indexation et recherche d'images par le contenu permettent de rechercher les images d'une base d'images en fonction de leurs caractéristiques visuelles. Ces caractéristiques, encore appelées

caractéristiques de bas-niveau sont la couleur, la texture, la forme et tout autre caractère de l'image qu'on peut imaginer. Dans ce document, nous appelons ces caractéristiques des caractéristiques symboliques (dans le domaine de recherche d'images, parfois des auteurs appellent ces caractéristiques des caractéristiques symboliques, parfois des auteurs appellent ces caractéristiques des caractéristiques symboliques).

Dans cette section, nous présentons l'architecture du système de recherche d'images par le contenu. Nous présentons aussi la façon que ce système utilise pour prendre la (les) requête(s), calculer la distance et lancer des résultats. Ensuite nous rappelons des concepts de base sur des caractéristiques visuelles : caractéristiques globales ou caractéristiques locales. Une image contient beaucoup de caractéristiques. Donc comment peut-on choisir une (des) caractéristiques adéquates et comment peut-on combiner ces caractéristiques ? A la fin de la section, nous allons discuter ces aspects.

2.1.1. Architecture du système d'indexation et recherche d'images

La figure 2-1 présente l'architecture du système de recherche d'images par le contenu.

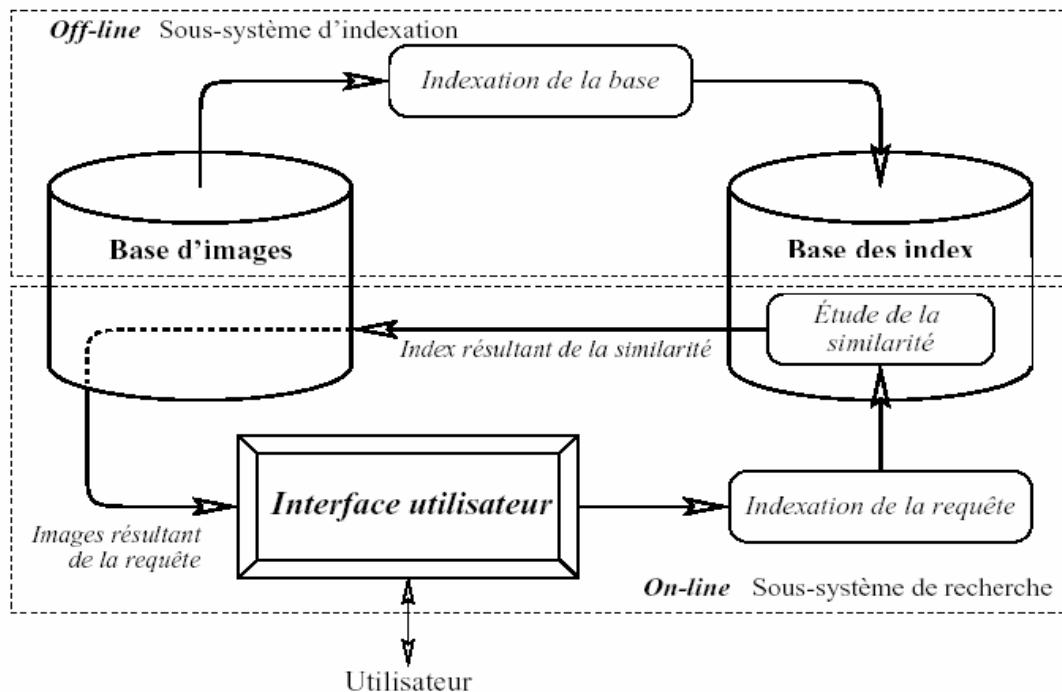


Fig. 2-1: *L'architecture d'un système d'indexation et recherche d'images par le contenu*

Ce système s'exécute en deux étapes : l'étape d'indexation et l'étape de recherche.

Dans l'étape d'indexation, des caractéristiques sont automatiquement extraites à partir de l'image et stockées dans un vecteur numérique appelé descripteur visuel. Grâce aux techniques de la base de données, on peut stocker ces caractéristiques et les récupérer rapidement et efficacement.

Dans l'étape de recherche, le système prend une ou des requêtes à l'utilisateur et lui donne le résultat correspond à une liste d'images ordonnées en fonction de la similarité entre leur descripteur visuel et celui de l'image requête en utilisant une mesure de distance. Néanmoins, nous nous concentrerons sur l'étape de recherche.

2.1.2. Les requêtes

Il existe 3 façons de faire une requête dans un système d'indexation et recherche des images : soit une requête par mots clés, soit une requête par esquisse, soit une requête par exemple. La figure 2-2 donne une démonstration pour les trois façons.

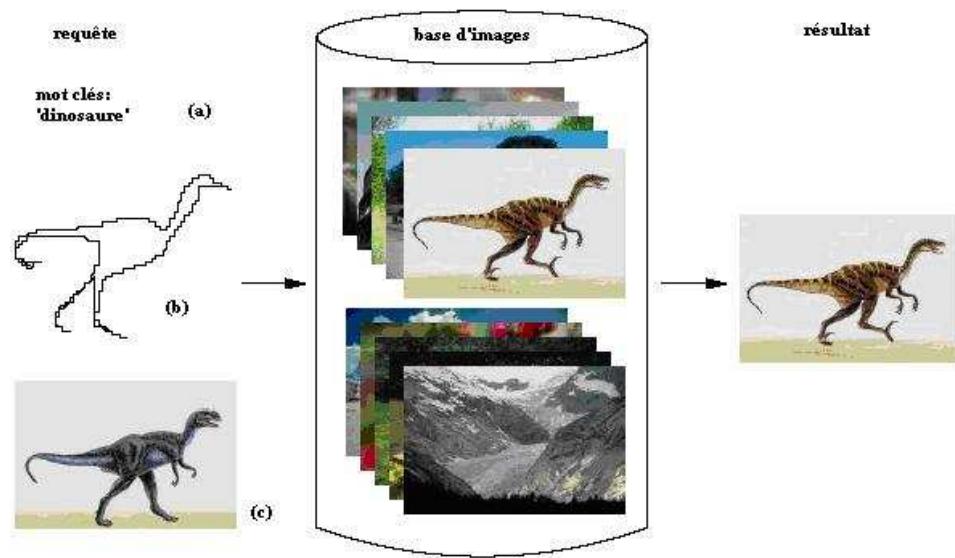


Fig. 2-2 : *Trois façons pour donner une requête dans CBIR (Deselaers, 2003)*

Pour travailler avec des requêtes par mots clés, le système doit extraire des mots clés à partir d'une image. Afin de faire ce travail le système se base sur l'annotation textuelle manuelle d'images. Cette façon est aujourd'hui la plus employée. Beaucoup de moteurs de recherche d'images tels que Google, Yahoo...utilisent cette façon. Mais cette façon n'est pas parfaite parce que quelques mots n'expriment pas le sens d'une image. La figure 2-3 donne un exemple, l'utilisateur veut trouver des images contiennent une (des) voiture(s) avec le ciel cependant les premières images résultats ne sont pas pertinentes car ils n'ont ni les voitures ni le ciel. De plus l'indexation de ces images

représente une tâche longue et répétitive pour l'humain, surtout avec les bases d'images qui deviennent aujourd'hui de plus en plus grandes. Elle est subjective à la culture, à la connaissance et aux sentiments de chaque personne.

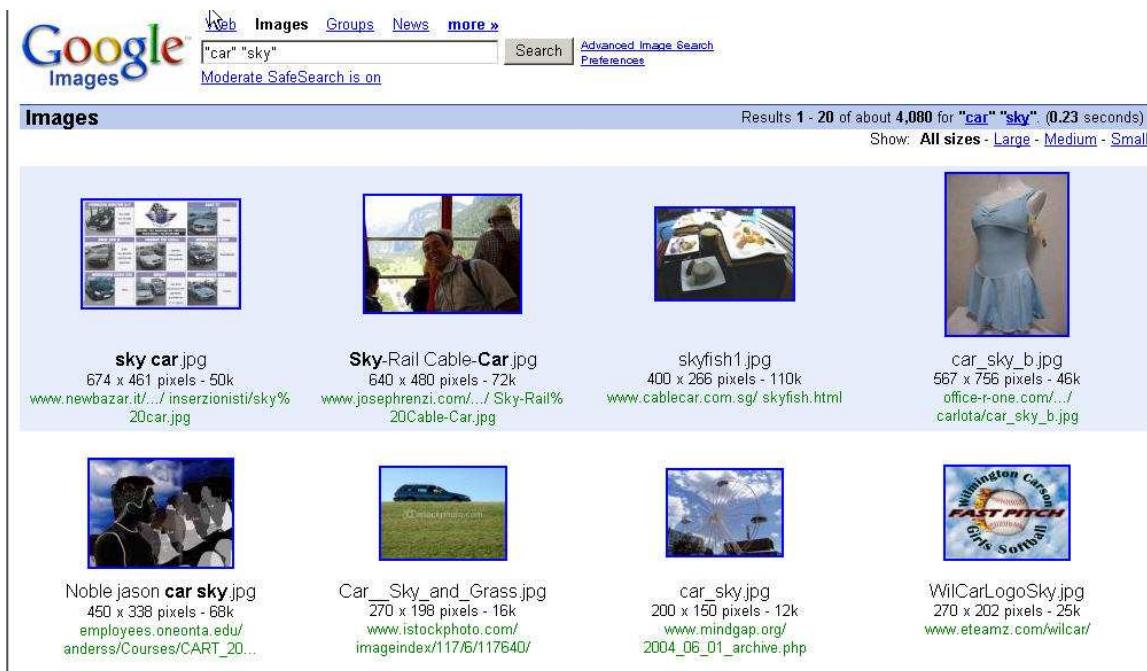


Fig. 2-3 : *Un exemple de recherche d'images dans Google*

Puisque la requête par mots clés a des désavantages, les systèmes de recherche d'images par le contenu utilisent souvent les deux autres façons : requête par exemple et requête par esquisse.

Dans la première approche, le système donne des outils à l'utilisateur. Ces outils lui permettent de dessiner des images requêtes. La figure 2-4 montre un exemple dans le système QBIC (QBIC). Ce système permet de créer une requête avec les formes et les couleurs des objets.

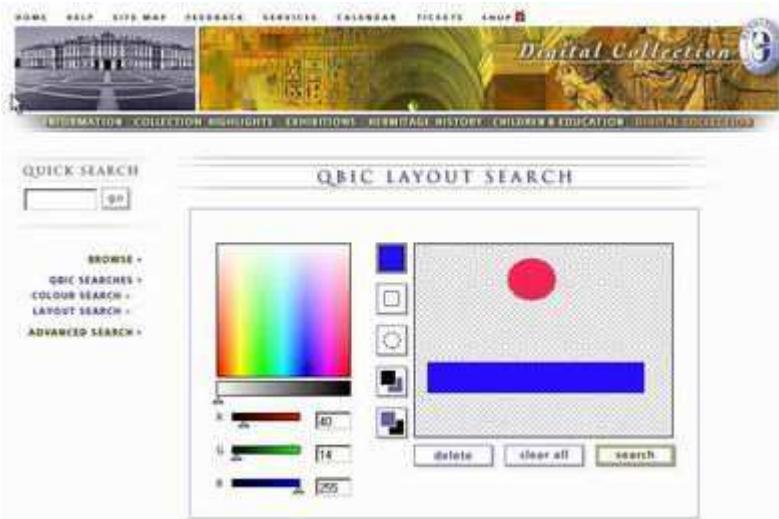


Fig. 2-4 : *Une requête par esquisse dans QBIC (QBIC)*

Dans la deuxième approche, soit le système choisit quelques images au hasard dans la base d’images et les montre à l’utilisateur, soit l’utilisateur parcourt la base d’images et choisit une requête, soit l’utilisateur donne son image requête.



Fig. 2-5 : *Une requête par image dans Blobworld (Blobworld)*

Par exemple, dans le système Blobworld, pour choisir une requête, le système affiche quelques classes d’images tels que ‘Animal’, ‘Ocean

Scenes', 'Outdoor Scenes', 'People', 'Manmade Objects'. L'utilisateur clique sur une classe pour voir quelques images dans cette classe et choisit une de ces images comme image requête. Dans la figure 2-5 l'utilisateur clique sur la classe 'Flower' et quelques images de fleurs sont affichées.

Dans quelques cas, la requête par esquisse devient difficile. Il est parfois difficile de faire l'esquisse et il est difficile de détecter les différences avec les caractéristiques de l'image. Donc en général, le système utilise seulement le paradigme requête par image ou utilise les deux approches ensembles.

Quelquefois l'utilisateur veut donner la requête tel que « Retrouvez toutes les images de la base ayant certaines régions semblables à celles de l'image requête» (Malki, 1999). Quelques travaux ont été effectués, soit le système segmente une image, soit on utilise une structure spatiale. Dans le premier cas, après avoir segmenté une image, l'utilisateur choisit quelques régions que l'on veut à partir de l'image segmentée. Dans le deuxième cas, on utilise des structures spatiales. Une structure souvent utilisée est l'arbre quartenaire (Malki, 1999). Cette structure permet de stocker les caractéristiques visuelles des différentes régions d'image et de filtrer les images en augmentant au fur et à mesure le niveau de détails.

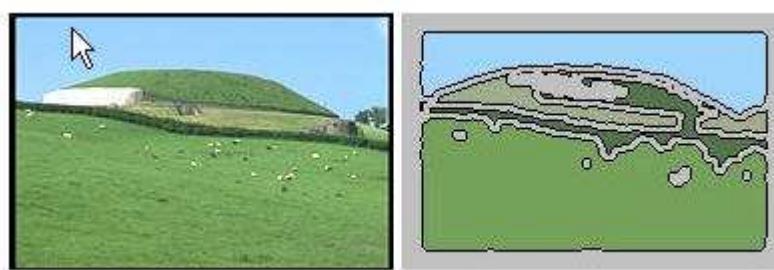


Fig. 2-6 : *Une requête par régions*

Dans la figure 2-6, le système fait la segmentation d'une image et l'utilisateur choisit quelques régions (correspondantes aux objets) pour faire la requête.

D'autre part, l'utilisateur veut trouver des images qui contiennent des régions à partir de plusieurs images requêtes. Dans (Zhu, 2000), les auteurs ont proposé des façons différentes pour combiner les résultats.

Dans tous les cas, après avoir pris des requêtes (images ou régions) le système va extraire un vecteur caractéristique de ces requêtes et va l'utiliser pour comparer avec ceux des images dans la base d'images.

Dans les cas précédents, nous supposons que l'étape pour saisir la requête et l'étape de recherche d'images sont séparées. Mais les deux étapes peuvent avoir lieu en même temps. On appelle cette approche Naviguer. Nous allons décrire cette approche dans la section sur le retour de pertinence.

L'approche pour donner des requêtes par exemples est très visuelle et très souvent utilisée. Mais cette approche a un inconvénient. Les systèmes supposent que plus il y a d'exemples de questions disponibles, meilleurs sont les résultats. Par conséquent, les utilisateurs pensent qu'ils peuvent avoir plus de résultats pertinents s'ils ajoutent de nouveaux exemples au résultat de la requête précédente. Cependant, cette supposition n'est pas toujours vraie. Les exemples additionnels peuvent contenir des caractéristiques inappropriées et faire baisser la performance du système.

La figure 2-7 donne un exemple. Dans cet exemple, l'utilisateur essaie de rechercher des images de voitures. La colonne de gauche affiche les résultats quand il n'y a qu'un exemple. La colonne de droite affiche les résultats quand il y a deux exemples.



Fig. 2-7 : *Un exemple quand l'utilisateur essaie de rechercher des images de voitures. La colonne de gauche affiche les résultats quand il n'y a qu'un exemple. La colonne de droite affiche les résultats quand il y a deux exemples (Nakazato, 2002)*

2.1.3. Caractéristiques symboliques

Une image contient plusieurs caractéristiques symboliques. Dans cette section, nous allons présenter des caractéristiques que l'on utilise souvent dans les systèmes de recherches des images telles que la couleur, la texture, la forme...

La couleur

La couleur est très souvent le premier descripteur qui est employé pour la recherche d'images. Plusieurs travaux ont déjà prouvé qu'il s'agit d'un descripteur efficace (Smeulders, 2000). Il existe de nombreuses possibilités d'attributs pour caractériser la couleur tels que l'histogramme, les moments

couleur...Et on trouve qu'avec la même méthode, si on change l'espace de couleur, il peut émerger des informations différentes de l'image. Cette section se consacre aux méthodes de couleur les plus utilisées dans un système d'indexation et recherche d'images ainsi que sur les espaces de couleur.

L'histogramme

Une technique très utilisée pour la couleur est l'intersection d'histogrammes (Swain, 1991). Les histogrammes sont faciles et rapides à calculer, et robustes à la rotation et à la translation. Cependant l'utilisation d'histogrammes pour l'indexation et la recherche d'images pose quatre problèmes (Gong, 1998). Premièrement, ils sont de grandes tailles, donc par conséquent il est difficile de créer une indexation rapide et efficace en les utilisant tels quels. Deuxièmement, ils ne possèdent pas d'informations spatiales sur les positions des couleurs. Troisièmement, ils sont sensibles à de petits changements de luminosité, ce qui est problématique pour comparer des images similaires, mais acquises dans des conditions différentes. Et quatrièmement, ils sont inutilisables pour la comparaison partielle des images (objet particulier dans une image), puisque calculés globalement sur toute l'image.

Plusieurs travaux récents ont été effectués pour améliorer ces problèmes. Il existe deux approches suivies par les chercheurs : l'une ajoute des informations spatiales à l'histogramme, l'autre utilise d'autres espaces de codage des couleurs qui se rapprochent plus de la perception humaine.

La première approche ajoute des informations spatiales aux histogrammes. Dans (Stricker, 2004), les auteurs ont divisé une image en cinq blocs fixés superposés et ils ont extrait les trois premiers moments d'inertie de chaque bloc pour créer un vecteur de descripteurs. (Pass, 1996) ont ajouté de la cohérence spatiale dans des histogrammes. Un pixel est cohérent s'il

appartient à une région validée par la segmentation et incohérent autrement (soit les pixels situés hors des régions segmentées). Les valeurs d'un histogramme sont divisées en deux classes en fonction des pixels correspondants à ces valeurs : classe cohérente et classe incohérente. La comparaison entre deux histogrammes devient la comparaison entre les valeurs d'histogrammes dans les classes correspondantes. Huang et al. (Huang, 1997) ont proposé le corrélogramme et l'auto-corrélogramme.

Les espaces de couleur

La deuxième approche recherche d'autres espaces de couleurs qui se basent sur la perception de couleur de l'humain. L'espace RVB est un espace de couleur utilisé couramment, par facilité, dans tous les systèmes de vision automatique, mais il n'est pas forcément le mieux adapté. En effet, les trois composantes RVB sont très dépendantes les unes les autres. Un simple changement d'éclairage dans la scène modifie les trois composantes, alors que les objets de la scène n'ont pas changé de couleur, mais se sont simplement assombris ou éclairés.

L'espace TSV (Teinte Saturation Valeur - en anglais HSV) est le plus utile pour la segmentation et la reconnaissance et il a été prouvé un espace très fort dans le système de recherche des images. Parce qu'il est une représentation plus physique de la couleur. Dans cet espace, on peut séparer pour un pixel : l'intensité du pixel (valeur) et la couleur du pixel (teinte et saturation). On n'a pas cette séparation dans le codage RVB. En RVB les trois informations ne sont pas entièrement décorrélées.

Non seulement les espaces RVB, TSV ont été étudiés, mais aussi d'autres espaces. Dans (Smeulders, 2000) les auteurs ont présenté quelques expérimentations intéressantes pour utiliser d'autres espaces en indexation,

tandis que Park et al. (Park, 1999) ont proposé l'espace CIE LUV et Gong et al. (Gong, 1998) ont employé l'espace HVC.

Un panorama des travaux existants permet de trouver plusieurs espaces couleur différents comme solution au lieu d'un espace unique. La solution se trouve probablement dans l'utilisation de plusieurs espaces complémentaires (Vandenbroucke, 2000).

Les moments statistiques

La méthode d'histogramme utilise la distribution complète de la couleur. On doit stocker de nombreuses données. Au lieu de calculer la distribution complète, dans les systèmes de recherche d'images, on calcule seulement des dominantes caractéristiques de couleur tels que l'espérance, la variance et d'autres moments. Dans (Stricker, 1995) les auteurs ont prouvé que les méthodes utilisées des moments statistiques marchent plus vite et donnent des résultats meilleurs que les méthodes d'histogrammes. L'espérance, la variance, les moments d'ordre 3 peuvent également être calculés sur chaque composante couleur par la formule suivante :

$$E_i = \frac{1}{N} \sum_{j=1}^N p_{ij} \quad (2.1)$$

$$\delta_i = \left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^2 \right)^{1/2} \quad (2.2)$$

$$s_i = \left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^3 \right)^{1/3} \quad (2.3)$$

Où E_i est l'espérance, δ_i est la variance, s_i est le troisième moment qui correspond à composante couleur i (i est R, V, B de l'espace RVB ou T, S, V de l'espace TSV).

La texture

La texture, autre primitive visuelle, est étudiée depuis une vingtaine d'années et plusieurs techniques ont été développées pour l'analyser. Certaines approches récentes semblent donner de bons résultats. Une méthode très connue pour analyser la texture est la matrice de co-occurrences de Haralick. Notre système de recherche des images par le contenu utilise ces matrices de co-occurrences. Donc nous allons les détailler dans la section 2.2.1. Il existe aussi d'autres méthodes pour analyser les textures dont celles basées sur la transformée en ondelettes de Gabor, sur les caractéristiques de Tamura, Word decomposition, Fractal...

Après avoir appliqué la transformation de Gabor sur une image, une région de texture est caractérisée par la moyenne et la variance des coefficients de transformation. Un vecteur de caractéristiques est construit en utilisant ces caractéristiques comme composants. Les caractères symboliques calculés à partir de la transformée en ondelettes de Gabor sont notamment très utilisés à l'heure actuelle pour caractériser les textures. L'intérêt principal des coefficients de Gabor est leur invariance aux rotations et mises à l'échelle. Pour plus détails sur la méthode basée sur l'ondelettes, le lecteur pourra se reporter à l'article (Wang, 1997).

Caractéristiques globales vs locales

A la base, les caractéristiques présentées précédemment sont calculées de façon globale sur l'image. Cependant, un système basé uniquement sur des caractéristiques globales ne peut pas donner les résultats désirés. Soit une image composée de plusieurs objets ayant des caractéristiques, couleurs et textures, très différentes, le vecteur de caractéristiques globales extrait à partir de l'image entière perd les informations locales (les objets) et ne produit qu'une moyenne grossière du contenu de cette image. Par contre, l'analyse

uniquement basée sur des caractéristiques locales risque de perdre le sens global de l'image, en submergeant celui-ci dans un flot de petits détails inutiles. Par conséquent, un compromis doit être trouvé, différent selon les applications et selon les requêtes individuelles, entre caractéristiques globales et caractéristiques locales.

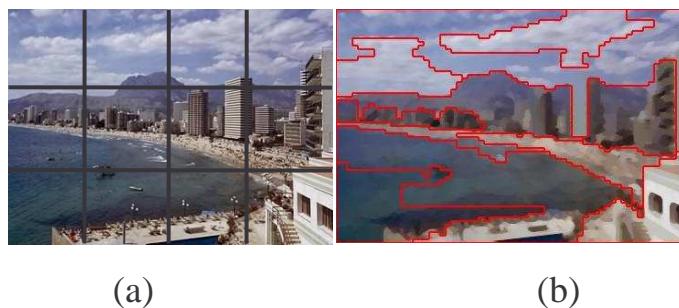


Fig. 2-8 : ***Deux découpages différents de l'image. (a) Découpage en grille de l'image. (b) Découpage par la segmentation division-fusion de l'image.***

Deux approches différentes peuvent être employées pour calculer les caractéristiques locales. La première approche consiste à diviser une image en utilisant une grille et les caractéristiques de chaque bloc sont calculées pour chaque case de cette grille. Par exemple dans la figure 2-8a, l'image est divisée en 12 zones de même taille et les caractéristiques sont calculées pour chacune de ces zones.

La deuxième approche consiste à segmenter l'image pour la diviser en zones locales plus proches des objets constituant l'image et ensuite calculer les caractéristiques pour chacune des régions extraites. La segmentation est une étape qu'on souhaiterait éviter, en raison de tous les problèmes pour choisir une "bonne" méthode de segmentation valide pour toutes les images de la base d'images. Cependant, la division d'une image en régions ou objets d'intérêt est souvent nécessaire pour pouvoir s'attaquer ensuite à l'extraction d'informations sémantiques à partir de l'image. Plutôt que de viser une segmentation exacte (en terme d'interprétation de l'image), on préfère

souvent parler de « groupement de pixels » (Forsyth, 2003) ou encore de « segmentation faible » (Medioni, 2005) selon les auteurs. Dans les deux cas, il s'agit simplement d'oublier le découpage précis en terme d'objets, mais plutôt de découper l'image en régions similaires du point de vue caractéristiques mesurées sur l'image (voir figure 2-8b). Cette dernière idée apparaît comme plus logique, car l'idée même d'objet est une vue sémantique de l'image, or il est facile de conclure à la lumière des innombrables travaux en segmentation des dernières décennies que la segmentation bas niveau (basée uniquement sur les caractéristiques issues de l'image) et le découpage sémantique de l'image sont deux idées complètement différentes. Un objet tel que nous le percevons dans l'image n'est pas forcément homogène en termes de caractéristiques, tandis qu'une zone homogène de l'image peut très bien contenir plusieurs objets différents.

Autres caractéristiques de l'image

Il existe plusieurs autres caractéristiques ou descripteurs, de l'image qui peuvent être utilisées pour mesurer la similarité d'une image, telles que la forme, les points d'intérêt, les invariants, et plus encore. La forme est une caractéristique visuelle importante et elle est une des caractéristiques de base pour décrire le contenu d'une image. Cependant, la description des formes est une tâche difficile, surtout dans un contexte de « segmentation faible » et dans un contexte où on recherche la similitude et non l'exactitude. Ainsi dans les systèmes de recherche récents, peu emploient la forme comme critère de recherche.

D'autres caractéristiques telles que des invariants, les points d'intérêts sont recherchés pour améliorer les performances des systèmes de recherche d'images (Alfarez, 2001 ; Sun, 2004 ; Wang, 1998). Iqbal et Aggarwal (Iqbal, 2002) ont ajouté la structure de l'image à la couleur et la texture pour le

système CIRES. Dans (Sciascio, 1997), les auteurs ont présenté un système de recherche d'images basé sur la couleur, l'orientation d'objet et la position relative. Chacune de ces combinaisons a des avantages et des inconvénients et chacune peut être bien adaptée pour certaines bases d'images, mais ne fonctionnera pas correctement avec d'autres.

Comme déjà vu précédemment, à partir une image, nous pouvons trouver beaucoup de caractéristiques. Leurs rôles vont changer dans les applications différentes. Les problèmes du bon choix de caractéristiques pour décrire l'image et de la pondération de ces caractéristiques ont été posés, mais ils ne sont pas encore résolus. Alors qu'un consensus minimal sur l'utilisation de la couleur et de la texture semble se dégager des travaux présentés ces dernières années, certains systèmes ajoutent d'autres caractéristiques, variables selon les systèmes et surtout selon les applications visées. Quant à la pondération des caractéristiques dans la fonction de mesure de la similarité entre les images, elle varie aussi fortement selon les systèmes, les applications et même selon les requêtes. La solution semble se trouver dans l'adaptation de la fonction et de ses poids, mais selon quels critères ? Ces critères ne sont pas forcément d'ordre symbolique, mais plutôt d'ordre sémantique. Deux approches proposées pour résoudre ce problème : l'une demande à l'utilisateur de donner des poids pour chaque caractéristique (par exemple 35% de couleur, 50% de texture...), l'autre apprend les poids à partir des requêtes pour faire un retour de pertinence. La première approche est très difficile à effectuer car l'utilisateur n'est pas capable de donner de bons chiffres. La deuxième approche est utilisé de plus en plus dans les systèmes d'indexation et recherche des images à l'heure actuelle. Il donne des résultats très prometteurs. Nous allons détailler cette approche dans la section 3.3.1.

2.1.4. Mesures pour évaluer un système

Avant l'exécution d'un système de recherche d'informations, une évaluation qui permet de mesurer la performance de ce système est nécessaire. Les mesures les plus courantes pour évaluer un système sont le temps de réponse et l'espace utilisé. Plus le temps de réponse est court, plus l'espace utilisé est petit, et plus le système est considéré bon. Mais avec des systèmes qui ont été faits pour la recherche d'informations, en plus de ces deux mesures, on s'intéresse à d'autres mesures. Dans le système de recherche d'informations, l'utilisateur s'intéresse aux réponses pertinentes du système. Donc les systèmes de recherche d'informations exigent l'évaluation de la précision de la réponse. Ce type d'évaluation est considéré comme l'évaluation des performances de recherche. Le système d'indexation et de recherche d'images est un système de recherche d'informations. Dans les systèmes de recherche d'images, les auteurs ont souvent utilisé les mesures d'évaluation que l'on a utilisé pour évaluer des systèmes de recherche d'informations.

Dans cette section, nous allons décrire les deux mesures les plus courantes: le rappel et la précision. Ces mesures sont reliées entre elles. Donc on décrit souvent cette relation par une courbe de rappel et précision. Ensuite nous présentons d'autres mesures que l'on utilise aussi pour évaluer des systèmes de recherche d'informations.

Rappel et précision (en anglais : Recall and Precision)

Le rappel:

Le rappel est le rapport entre le nombre d'images pertinentes dans l'ensemble des images trouvées et le nombre d'images pertinentes dans la base d'images.

$$Rappel = \frac{|Ra|}{|R|} \quad (2.4)$$

La précision :

La précision est le rapport entre le nombre d'images pertinentes dans l'ensemble des images trouvées et le nombre d'images trouvées.

$$\text{Précision} = \frac{|Ra|}{|A|} \quad (2.5)$$

Où :

- I : une image requête
- R : l'ensemble d'images pertinentes dans la base d'images utilisée pour évaluer.
- |R| : le nombre d'images pertinentes dans la base d'images.
- A : l'ensemble des réponses.
- |A| : le nombre d'images dans l'ensemble des réponses.
- |Ra| : le nombre d'images pertinentes dans l'ensemble des réponses.

Des définitions sont montrées dans la figure 2-9 :

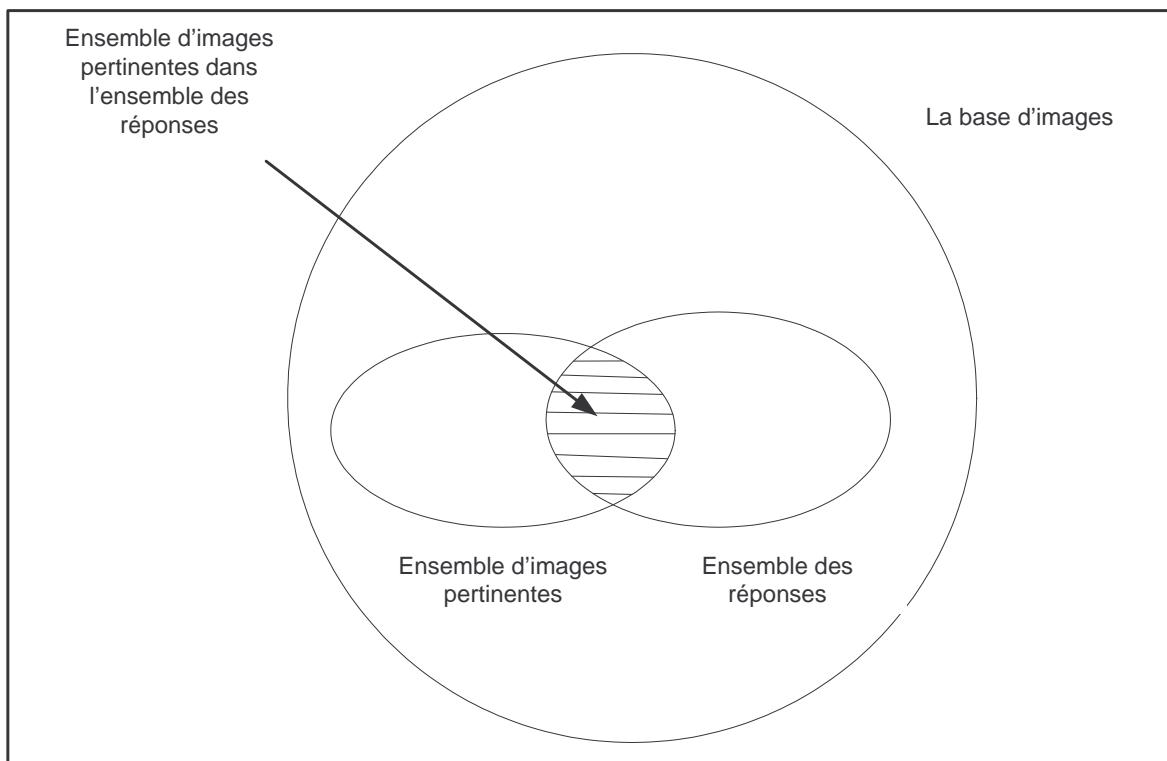


Fig. 2-9 : *Le rappel et la précision pour une requête (Yates, 1999)*

Dans les systèmes de recherche d’informations, afin de définir si une information est pertinente ou non, on a besoin d’experts dans le domaine. Dans les systèmes de recherche d’images, une image est pertinente pour une requête si les deux images sont dans la même classe. C’est pourquoi dans l’étape de préparation de la base d’images pour évaluer, on doit faire des annotations. L’annotation est un processus qui permet aux utilisateurs de choisir des mots clés correspondants à chaque image. Après l’annotation, on va classifier les images en classes appropriées. Si des images ne contiennent pas beaucoup d’objets, c’est facile de les classifier dans ces classes. Mais si les images contiennent beaucoup d’objets, la tâche de classification devient de plus en plus difficile. Dans ce cas là, chaque image appartient à plusieurs classes.

La courbe de rappel et précision

Le rappel et la précision sont les mesures importantes, mais si on voit seulement une paire de valeurs de rappel et précision, cette paire de valeurs ne peut pas indiquer la performance du système. C'est pourquoi on donne souvent une distribution de rappel et précision sous forme de courbe.

La figure 2-10 donne un exemple de courbe de rappel et précision. Pour dessiner cette courbe, on doit calculer plusieurs paires de rappel et précision et les interpoler.

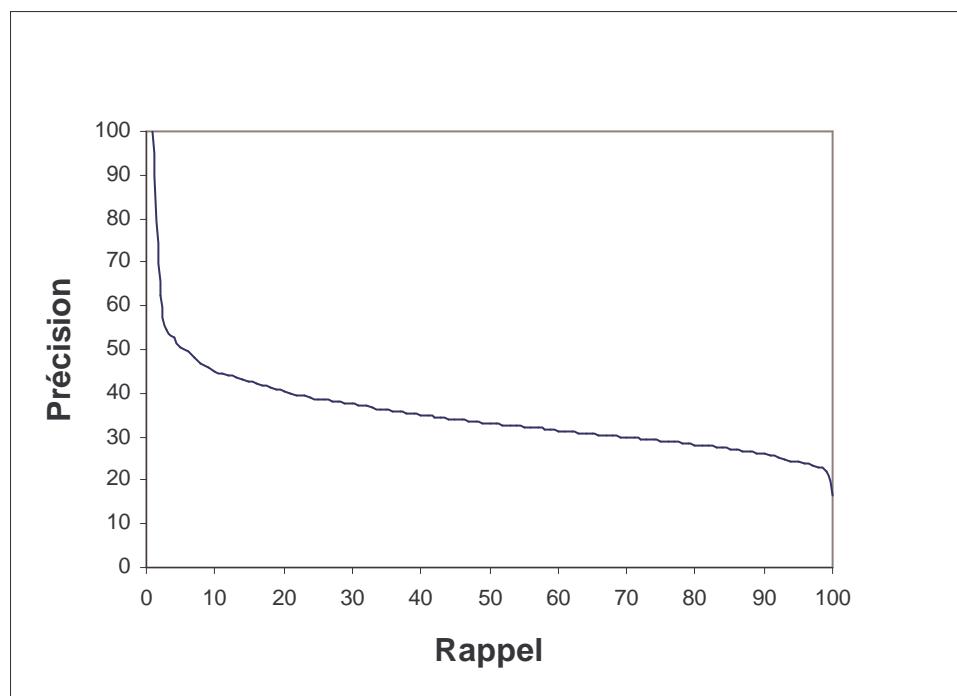


Fig. 2-10 : *Une courbe de rappel et précision*

En pratique, on utilise plusieurs requêtes. Dans ces cas pour évaluer un système, on calcule la précision moyenne pour toutes les requêtes correspondant à chaque niveau de rappel.

$$\overline{P(r)} = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q} \quad (2.6)$$

où :

- N_q est le nombre de requêtes

- $P_i(r)$ est la précision pour le rappel r avec la requête i

A côté de la courbe de rappel et précision, d'autres mesures ont été utilisées pour évaluer un système d'indexation et recherche d'images telles que :

- $P(10)$, $P(30)$, $P(|R|)$: la précision quand 10, 30, $|R|$ images retrouvées.
- La moyenne de la précision
- Le rappel quand la précision est 0.5
- $R(1000)$: le rappel pour 1000 images retrouvées.
- ...

Pour plus de détails sur d'autres mesures, le lecteur pourra se reporter aux articles (Muller, 1999) et (Smith, 1998).

Le rappel et la précision sont très utiles parce qu'ils nous permettent d'évaluer quantitativement la qualité de la réponse globale et la largeur de l'algorithme de recherche. Mais ils ont 4 désavantages (Yates, 1999). Premièrement, l'estimation de la valeur maximum de rappel exige de savoir toutes les connaissances de la base d'images. Quand la base d'images devient de plus en plus grande, ces connaissances ne sont pas disponibles, ce qui veut dire que l'évaluation n'est pas bien estimée. Deuxièmement, le rappel et la précision sont reliés. Donc dans quelques cas, les deux mesures ne sont pas suffisantes. Dans ces cas, l'utilisation d'autres mesures qui combinent le rappel et précision pourrait être plus appropriée. Troisièmement, ces mesures travaillent bien sur un ensemble de requêtes par lots (non-interactif). Cependant, les systèmes modernes ne travaillent pas dans ce mode. Ils permettent de communiquer avec l'utilisateur donc dans ces systèmes d'autres mesures qui sont plus appropriées doivent être utilisées. Quatrièmement, le rappel et la précision sont faciles à définir quand l'ordre des images est

linéaire. Ces mesures ne sont pas appropriées pour les systèmes qui ont un ordre faible.

A cause de ces désavantages, dans les systèmes de recherche d'informations on essaie d'utiliser d'autres mesures telles que la moyenne harmonique , la mesure E,... que nous allons décrire et commenter dans la section suivante.

D'autres mesures

La moyenne harmonique

La moyenne harmonique de rappel et précision $F(j)$ a été définie par la formule suivante :

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}} \quad (2.7)$$

Où :

- $r(j)$: le rappel quand on donne j images en réponse.
- $P(j)$: la précision quand on donne j images en réponse.

La valeur de cette mesure est entre 0 et 1. Si la valeur est 0 ça veut dire qu'aucune image pertinente a été retrouvée. Si la valeur est 1, toutes les images pertinentes ont été retrouvées. De plus, cette mesure a une valeur élevée quand le rappel et la précision sont élevés.

La mesure E

La mesure E de rappel et précision $E(j)$ a été définie par la formule suivante :

$$E(j) = \frac{1+b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}} \quad (2.8)$$

Où :

- $r(j)$: le rappel quand on donne j images en réponse.
- $P(j)$: la précision quand on donne j images en réponse.
- b est le paramètre qu'on peut choisir. Ce paramètre reflète l'importance relative du rappel et de la précision. Si b est égal 1, $E(j)$ devient $F(j)$. Si b est plus grand que 1, ça veut dire que l'utilisateur s'intéresse plus à la précision qu'au rappel. Et si b est plus petit que 1, ça veut dire l'utilisateur s'intéresse plus au rappel qu'à la précision.

L'évaluation d'un système de recherche d'images est un problème difficile. Il manque une base d'images standard (qui doit être assez large, générique, et où on est capable d'avoir des ensembles d'images pertinentes, ...). De plus, il n'a pas de mesure assez efficace.

Dans cette section, nous avons présenté les mesures souvent utilisées dans les systèmes de recherche d'images actuels : le rappel, la précision et la courbe de rappel et précision. Nous avons aussi discuté leurs avantages et leurs inconvénients. En fin, nous avons introduit d'autres mesures d'évaluation.

Dans notre système d'indexation et recherche d'images, nous utilisons la courbe de rappel et précision moyenne pour chaque classe dans la base d'images et pour toute la base d'images. Nos résultats avec des courbes de rappel et précision seront présentés dans la section suivante.

2.2. Notre système de recherche par le contenu symbolique

2.2.1. Des caractéristiques choisies

Cette section se consacre aux caractéristiques utilisées dans notre système d'indexation et recherche des images. Ce sont l'intersection d'histogrammes, les moments de couleurs et la texture. Avec deux premières caractéristiques nous utilisons deux espaces couleurs : RVB et TSV. Nous

allons détailler précisément ces caractéristiques. La distance entre deux images devient la distance entre leurs caractéristiques.

Couleur – L'intersection d'histogramme

Une technique très utilisée pour la couleur est l'intersection d'histogrammes. Avec cette méthode, tout d'abord on doit calculer l'histogramme d'image (l'histogramme d'une image peut être présenté par un vecteur dont chaque composant est un nombre de pixels de couleur correspondant à son indice). Ensuite un *histobin* va être créé à partir de l'histogramme de la manière suivante : chaque *bin* (trou) dans l'*histobin* est la somme de quelques éléments voisins de l'histogramme. Le nombre de voisins est déterminé par le nombre de *bin* de l'*histobin*. Soit N est le nombre de *bin* pour chaque composant, l'*histobin* a $3N$ *bin* au total. Le nombre de voisins est $256/N$. On peut voir que l'*histobin* est plus compact que l'histogramme Dans notre système nous calculons l'*histobin* pour chaque composant couleur (R, V, B dans l'espace RVB et T, S, V dans l'espace TSV).

La figure 2-11 donne un exemple. L'image originale est à gauche, l'histogramme est au milieu et l'*histobin* de l'histogramme est à droite.



Fig. 2-11 : *L'histogramme et l'histobin*

Après avoir eu l'*histobin* d'image, la distance entre deux images devient la distance entre deux *histobin*. On peut calculer cette distance par la formule suivante (Swain, 1991) :

$$d(H, I) = \frac{\sum_i |H_{Histo\ i} - I_{Histo\ i}|}{\sum_i H_{Histo\ i}} \quad (2.9)$$

où

- H,I : deux images
- H_{Histo} : l'histobin de H
- I_{Histo} : l'histobin de I
- $H_{Histo\ i}$: le bin i d'histobin de H.
- $I_{Histo\ i}$: le bin i d'histobin de I
- $D_{Histo}(H, I)$: la distance entre l'image H et l'image I en fonction l'intersection d'histogrammes.

Nous notons $D_{HistoRVB}(H, I)$ le calcul fait avec l'espace RVB et $D_{HistoTSV}(H, I)$ le calcul fait avec l'espace TSV.

Couleur – Moments des couleurs

Notre système utilise les moments des couleurs : l'espérance (ou moyenne), la variance, le moment de troisième ordre (défini par (2.1), (2.2) et (2.3)) pour chaque composante couleur dans deux espaces couleur RVB et TSV. Au total, nous avons 9 chiffres pour RVB (3 pour chaque composante), 9 chiffres pour TSV (3 pour chaque composante).

H est une image, représentée par un ensemble de 9 chiffres par espace couleur i (i : RVB ou TSV) $\{E_i, \sigma_i, s_i\}$. I est autre image, avec un ensemble de 9 chiffres par espace couleur j(RVB ou TSV) $\{F_i, \zeta_i, t_i\}$.

La distance entre l'image H et l'image I:

$$D_{Moment}(H, I) = \sum w_{i1} |E_i - F_i| + w_{i2} |\sigma_i - \zeta_i| + w_{i3} |s_i - t_i| \quad (2.10)$$

où r est le nombre de composantes dans l'espace j et w_{ij} avec $j=\{1,2,3\}$ sont les poids correspondants à l'espérance, la variance et le moment de troisième ordre.

Les espaces couleurs utilisés

Dans notre système, nous utilisons deux espaces couleurs : RVB et TSV (en anglais HSV). Nous avons choisi ces espaces parce que RVB est l'espace couramment utilisé et TSV est un espace très fort dans le système de recherche des images et plus indépendant face aux variations d'éclairage. Pour transformer une image RVB en une image TSV, nous avons utilisé une formule qui est décrite dans l'annexe 1.

En ce moment, nous utilisons seulement deux espaces fondamentaux. D'autres espaces peuvent aisément se rajouter dans notre système. Nous pensons que chaque espace joue un rôle différent pour des bases d'images différentes. Donc pour chaque application, on doit tester plusieurs espaces couleurs et va choisir l'espace le plus efficace.

Texture- Les matrices de co-occurrences

En 1973, Haralick (Haralick, 1973) a proposé une méthode en se basant sur les matrices de co-occurrences de niveaux de gris. Elle est probablement une des méthodes les plus célèbres pour analyser la texture. La matrice $P_{Har, d, \alpha}(g, g')$ compte les nombres de paires de pixel (m, n) et (m', n') dans une image qui ont une valeur d'intensité g et g' avec une distance d dans une direction α .

Dans (Lan, 1999) un exemple a été montré comme suit :

Une région de 15 pixels quantifiés sur 8 niveaux de gris.

1	2	1	3	4
2	3	1	2	4
3	3	2	1	1

Tab 2-1 : *Région de 15 pixels*

Matrice de cooccurrence avec $d = 1$ et $\alpha = 0$ (un pixel vers le bas)

a	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	0	0	0	0	0
2	0	1	0	2	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	1	0	0	1	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Tab 2-2 : *Matrice de cooccurrence avec $d = 1$ et $\alpha = 0$ (un pixel vers le bas)*

Afin d'estimer la similarité entre les matrices de cooccurrences, Haralick a proposé 14 caractéristiques statistiques extraites à partir de cette matrice. Actuellement, seulement les quatre caractéristiques les plus appropriées sont largement utilisées : l'énergie, l'entropie, le contraste et le moment inverse de différence.

- L'énergie (T_1) :
$$\sum_i \sum_j P_d^2(i, j) \quad (2.11)$$

- L'entropie (T_2):
$$-\sum_i \sum_j P_d(i, j) \log P_d(i, j) \quad (2.12)$$

- Le contraste (T_3):
$$\sum_i \sum_j (i - j)^2 P_d(i, j) \quad (2.13)$$

- Le moment inverse de différence (T_4):
$$\sum_i \sum_j \frac{P_d(i, j)}{|i - j|^2}, i \neq j \quad (2.14)$$

La distance entre deux images H, I se calcule par la formule suivante :

$$D_{Texture}(H, I) = \sqrt{(T_{H1} - T_{I1})^2 + (T_{H2} - T_{I2})^2 + (T_{H3} - T_{I3})^2 + (T_{H4} - T_{I4})^2} \quad (2.15)$$

Où

- T_{H1} : l'énergie de H
- T_{I1} : l'énergie de I
- T_{H2} : l'entropie de H
- T_{I2} : l'entropie de I
- T_{H3} : le contraste de H
- T_{I3} : le contraste de I
- T_{H4} : le moment inverse de différence de H
- T_{I4} : le moment inverse de différence de I

La distance d choisie dans notre système dans toutes les directions est toujours 1.

Combinaison des caractéristiques

Puisque chaque distance entre caractéristiques a été calculée avec une norme différente, avant de combiner ces caractéristiques, nous devons normaliser les distances à l'aide d'une Gaussienne (Iqbal, 2002). Cette méthode assure que la distance est normalisée entre 0 et 1. Après avoir normalisé les distances, nous avons utilisé une combinaison linéaire des distances.

$$\begin{aligned} D(H, I) = & w_{HistoRVB} * D_{HistoRVB}(H, I) + w_{HistoTSV} * D_{HistoTSV}(H, I) + \\ & w_{MomentRVB} * D_{MomentRVB}(H, I) + w_{MomentTSV} * D_{MomentTSV}(H, I) + \\ & w_{Texture} * D_{Texture}(H, I). \end{aligned} \quad (2.16)$$

où $w_{HistoRVB}$, $w_{HistoTSV}$, $w_{MomentRVB}$, $w_{momentTSV}$, $w_{Texture}$ sont les poids correspondants à l'intersection d'histogrammes RVB, l'intersection

d'histogrammes TSV, la distance entre les moments RVB, la distance entre les moments TSV, la distance entre les textures. $D_{\text{HistoRVB}}(H, I)$, $D_{\text{HistoTSV}}(H, I)$, $D_{\text{MomentRVB}}(H, I)$, $D_{\text{MomentTSV}}(H, I)$, $D_{\text{Texture}}(H, I)$ sont les distances entre les intersections d'histogrammes RVB, les intersections d'histogrammes TSV, les moments RVB, les moments TSV, les textures de l'image H et de l'image I.

Pour faciliter le calcul, nous choisissons $1/N$ où N est le nombre de caractéristiques (dans ce cas $N=5$) pour chaque poids. Ces poids seront changés dans le processus de retour de pertinence.

Dans notre système, nous avons testé avec chaque caractéristique ainsi que pour les combinaisons entre ces caractéristiques. Les résultats obtenus sur plusieurs bases d'images seront présentés dans la section 2.2.4.

2.2.2. *Les bases d'images utilisées*

Pour évaluer et valider notre système, nous avons utilisé cinq bases d'images. Ces bases d'images sont disponibles sur Internet librement sauf la base Pollens. Ces bases d'images possèdent déjà des classes définies où chaque image n'appartient qu'à une seule classe.

La base de Wang

La base d'images de Wang est un sous-ensemble de la base d'images Corel. Cette base d'images contient 1000 images naturelles en couleurs. Ces images ont été divisées en 10 classes, chaque classe contient 100 images. L'avantage de cette base est de pouvoir évaluer les résultats. Cette base d'images a été utilisée pour faire des expériences de classification. Un exemple de chaque classe peut être vu sur la figure 2-12(Deselaers, 2003). Cette base d'images a été créée par le groupe du professeur Wang de l'université Pennsylvania State

et est disponible à l'adresse : <http://wang.ist.psu.edu/>. Chaque image dans cette base d'images a une taille de 384×256 pixels ou 256×384 pixels.



Fig. 2-12 : *10 classes de la base de Wang (Deselaers, 2003)*

Coil (Columbia Object Image Library)

Cette base d'images est très connue pour la reconnaissance des objets. Il y a deux bases d'images COIL : COIL-20 qui contient des images en niveaux de gris prises à partir de 20 objets différents et COIL-100 qui contient des images en couleurs prises à partir de 100 objets différents. Les deux bases d'images consistent en des images prises à partir des objets 3D avec des positions différentes. La base COIL-100 a 7200 images en couleurs (100 objets x 72 images/objet). Chaque image a une taille de 128×128 pixels. Tous les objets utilisés dans COIL-100 sont montrés dans la figure 2-13(Deselaers, 2003). La base COIL-20 a 1440 images en niveaux de gris (20 objets x 72 images/objet). Chaque image a taille 128×128 pixels. Tous les objets utilisés dans COIL-20 sont dépeints sur la figure 2-14(Deselaers, 2003). Ces bases d'images sont disponibles à l'adresse :

<http://www1.cs.columbia.edu/CAVE/research/softlib/>



Fig. 2-13 : *Les objets utilisés dans COIL-100 (Deselaers, 2003)*



Fig. 2-14 : *Les objets utilisés dans COIL-20 (Deselaers, 2003)*

Pollen

La base de 347 images de grains de pollen (31 classes au total) de l'INRIA Sophia-Antipolis et provenant du projet Européen ASTHMA1 (Bonton, 2002). Quelques exemples de cette base d'images sont montrés dans

la figure 2-15. Nous remercions le projet Orion de l'INRIA pour avoir permis l'utilisation ici de cette base d'images.

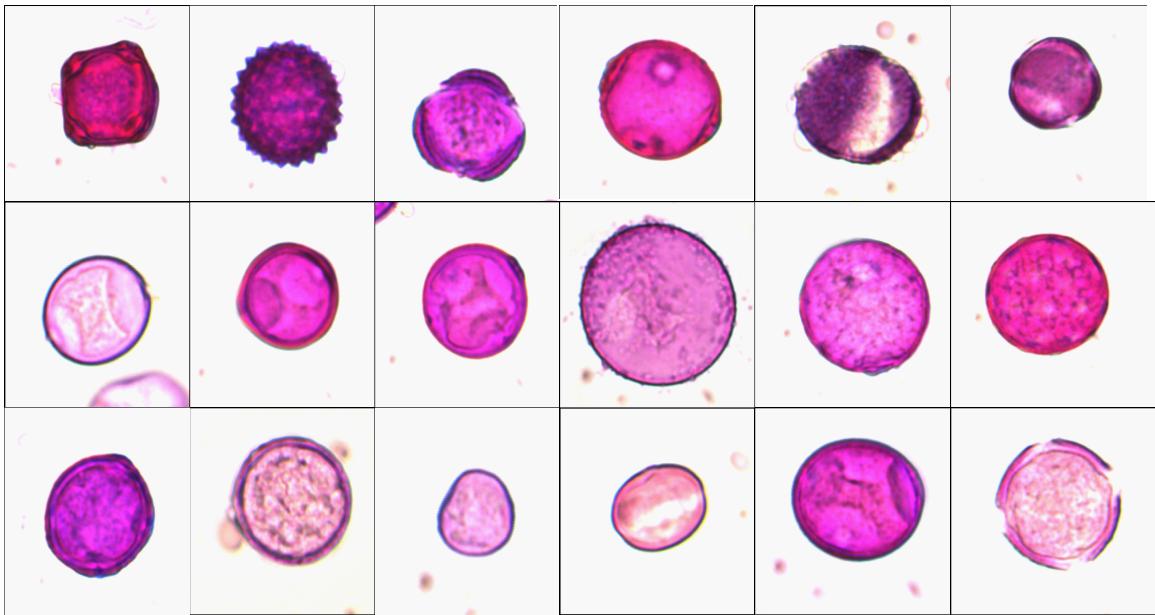


Fig. 2-15 : *Quelques images exemples de la base de Pollens*

CUReT(Columbia Utrecht Reflectance and Texture Database)

Cette base a plus de 14000 images de textures (70 textures x ~200 images/texture) des Universités Columbia et d'Utrecht (CUReT). Ces images sont disponibles à l'adresse : <http://www1.cs.columbia.edu/CAVE/curet>. Quelques exemples de cette base d'images sont montrés dans la figure 2-16.

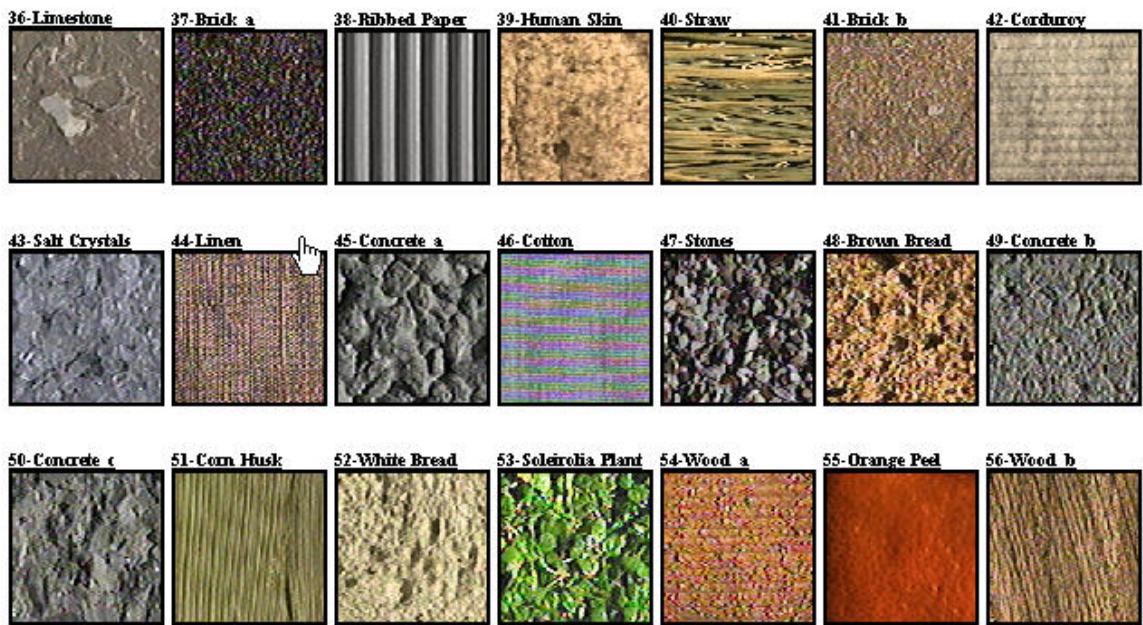


Fig. 2-16 : *Quelques images exemples dans la base de CUReT(CUReT)*

La base de FeiFei

Cette base contient des images de 101 objets collectées par Fei-Fei Li, Marco Adreetto et Marc Aurolio Ranzato. Avec chaque objet, de 40 à 800 images ont été prises. Chaque image a une taille de 300×200 pixels. Dans notre système, nous utilisons seulement une sous-partie de cette base (8296 images couleur de 100 objets). Ces images sont disponibles à l'adresse : <http://www.vision.caltech.edu/feifeili/Datasets.htm>. La figure 2-17 montre quelques exemples de cette base :



Fig. 2-17 : *Quelques images exemples dans la base de Fei-Fei*

2.2.3. Une image requête et plusieurs images requêtes

Les besoins de l'utilisateur sont très variés. Donc dans plusieurs cas, une seule requête n'est pas suffisante parce qu'elle n'est pas capable de

satisfaire complètement son désir. A partir de cette observation, notre système supporte les deux types de requête par l'exemple : une seule requête ou plusieurs requêtes simultanées.

Bien que le problème de multi-requêtes puisse être résolu en le divisant en plusieurs requêtes simples, il reste à savoir comment fusionne-t-on les résultats de chaque requête pour trouver les images pertinentes, ce qui est un problème important ?

Dans notre système, nous supportons 2 méthodes pour combiner les résultats : la méthode de combinaison par distance minimale et la méthode par combinaison linéaire.

Avant de les présenter, nous donnons quelques notations que nous employons.

La base d'images D contient m images. La requête Q avec k images requêtes peut être définie comme :

$$Q = \{ \langle q_1, w_1 \rangle, \dots, \langle q_t, w_t \rangle, \dots, \langle q_k, w_k \rangle \}$$

où w_i est entre 0 et 1. Ce paramètre est employé pour indiquer l'importance de l'image q_l dans la requête Q.

Avec chaque image requête, nous prenons les n images les plus semblables.

$$R = \{ \langle i_1, dist_1 \rangle, \dots, \langle i_j, dist_j \rangle, \dots, \langle i_n, dist_n \rangle \}$$

Nous voyons que la distance doit satisfaire : $dist_1 \leq \dots \leq dist_j \dots \leq dist_n$

Pour résoudre le problème de plusieurs images requêtes, nous divisons la requête Q en k sous-requêtes Q_i , chacune contenant une seule image requête.

$$Q_i = \{ \langle q_l, w_l \rangle \} \text{ où } 1 \leq l \leq k$$

Avec une sous-requête Q_i , nous avons le résultat R_i :

$$R_i = \{ \langle i_{1l}, dist_{1l} \rangle, \dots, \langle i_{nl}, dist_{nl} \rangle \}$$

où $n_l \leq n$ et $dist_{l1}$ est la distance entre l'image i_1 dans la base D avec l'image requête q_l .

Maintenant, nous allons fusionner k sous-résultats R_l pour avoir le résultat final R.

Méthode de combinaison par distance minimale

Dans cette méthode, la distance entre une image I dans la base d'images et la requête Q est la plus petite distance dans l'ensemble de distances entre I et Q_l .

$$dist'j = \min_{l=1}^k w_l * dist_{lj} \quad (2.17)$$

Le résultat qu'on trouve finalement :

$$R' = \{ \langle i_1, dist'_1 \rangle, \dots, \langle i_j, dist'_j \rangle, \dots, \langle i_n, dist'_n \rangle \}$$

Méthode par combinaison linéaire

Dans cette méthode, la distance entre une image I dans la base d'images avec la requête Q est la combinaison linéaire des distances entre I et Q_l .

$$dist'_j = \sum_{l=1}^k w_l * dist_{ij} \quad (2.18)$$

Le résultat qu'on trouve finalement :

$$R' = \{ \langle i_1, dist'_1 \rangle, \dots, \langle i_j, dist'_j \rangle, \dots, \langle i_n, dist'_n \rangle \}$$

Remarque : avant d'appliquer ces méthodes, les distances doivent être normalisées.

D'après nos observations, nous trouvons que la méthode de distance minimale marche mieux que la méthode par combinaison linéaire si l'utilisateur rajoute le même type d'images. Par exemple, dans la première requête, il donne une requête qui contient une fleur et ensuite il va rajouter une autre image requête qui contient aussi une fleur. Par contre la méthode linéaire marche mieux si le contenu des images requêtes est variable.

Il existe d'autres méthodes pour combiner les résultats telles que la combinaison non-linéaire des distances, la combinaison des rangs... Mais notre système utilise seulement ces deux méthodes parce que la méthode de combinaison par minimum est très efficace dans quelques cas et la méthode de combinaison linéaire a été prouvée assez bien dans la plupart des cas [Zhu00]. Nos résultats seront présentés dans la section 2.2.4.

Comme nous avons déjà dit dans la section 2.1.1, dans plusieurs cas la requête avec plusieurs images donne de meilleurs résultats que ceux avec une seule image requête. Cependant, ceci n'est pas toujours vrai, puisque nous travaillons avec des caractéristiques symboliques et globales. Un exemple a été montré dans la section 2.1.1. Nous pouvons les améliorer en utilisant des caractéristiques locales ou en rajoutant les valeurs sémantiques.

2.2.4. Nos résultats

Dans cette section, nous allons présenter quelques résultats avec l'interface et allons donner des courbes de rappel et précision pour chaque base d'images avec chaque descripteur. Avec la base de Wang, le nombre de classes n'est pas très grand, ce qui nous permet aussi de donner les courbes de rappel et précision pour chaque classe avec chaque descripteur. Nous allons terminer cette section en donnant quelques conclusions.

Pour présenter les résultats, nous avons développé un système de recherche d'images par le contenu symbolique. La figure 2-18 décrit l'interface de notre système. Cette interface se divise en trois zones : la zone de requête, la zone de résultats et la zone de descripteurs. Dans la zone de requête (en rouge), une image requête a été montrée avec son nom. Dans la zone de descripteurs, nous donnons 6 options : l'intersection d'histogrammes RVB, l'intersection d'histogrammes TSV, les moments RVB, les moments

TSV, la texture et la combinaison de tous ces descripteurs. L'utilisateur clique sur le bouton «Load Image» pour prendre une image requête et choisit une option de descripteur. Quand l'utilisateur clique sur le bouton «Search match», le système va comparer toutes les images dans la base d'images avec l'image requête et va montrer les seize premières images qui sont les plus semblables à l'image requête avec leurs rangs, leurs noms et leurs distances avec l'image requête.

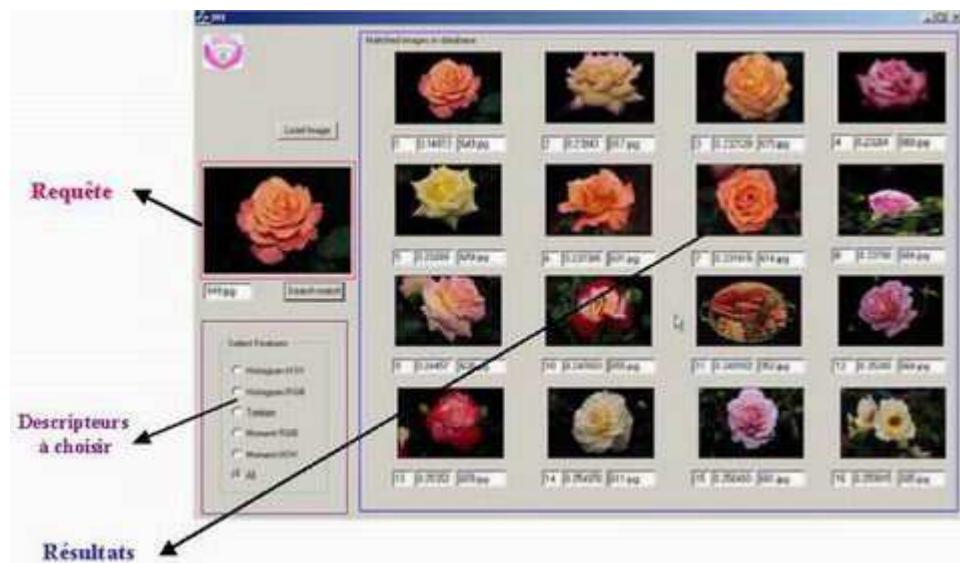


Fig. 2-18 : *Notre interface développée pour le système d'indexation et recherche d'images par le contenu*

Nous poursuivons cette section par des résultats avec cette interface pour quelques bases d'images et des courbes de rappel et précision.

La base de Wang

La figure 2-19 montre six résultats d'une image dans la base de Wang avec des options de caractéristiques différentes. Les images en rouge sont des images non-pertinentes.



L'intersection d'histogrammes
RVB 24 bin



L'intersection d'histogrammes
RVB 24 bin



Les moments RVB



Les moments TSV



La texture



La combinaison

Fig. 2-19 : *Quelques résultats avec la base de Wang*

A partir de ces résultats, nous pouvons calculer la table de rappel et précision. Dans la base de Wang, chaque classe contient 100 images donc $|R|=100$. Nous présentons seulement les 16 premiers résultats, alors le nombre

d'images dans l'ensemble de réponse est égal 16 ($|A|=16$). Nous supposons qu'une image résultat est pertinente si elle appartient à la même classe que l'image requête.

Descripteur	Nombre d'images pertinentes dans 16 images	Précision ($ Ra / A $) $ A =16$	Rappel ($ Ra / R $) $ R =100$
L'intersection d'histogrammes RVB	15	93.75%	15%
L'intersection d'histogrammes TSV	16	100%	16%
Les moments RVB	16	100%	16%
Les moments TSV	9	56.25%	9%
La texture	4	25%	4%
La combinaison	16	100%	16%

Avec $|R|=100$ et $|A|=16$, la précision maximale est de **100%**, le rappel maximal est de **16%**.

Tab 2-3 : *Les valeurs de rappel et précision de la base de Wang, avec un nombre d'images retrouvées est de 16.*

Nous trouvons qu'avec cette image requête, l'intersection d'histogrammes TSV, les moments RVB et la combinaison sont les meilleurs descripteurs. L'intersection d'histogrammes RVB et les moments TSV sont

des descripteurs assez efficaces. La texture n'est pas un descripteur très fort dans ce cas. Par contre, dans le cas général si l'utilisateur veut trouver des images contenant des animaux, la texture peut donner de bons résultats (avec des images contenant des chevaux et des éléphants par exemple) parce que la texture de la peau des différents animaux sont très semblables.

La figure 2-20 donne une vue globale de la performance de notre système avec la base de Wang. Nous pouvons trier les caractéristiques en ordre décroissant de leurs efficacités comme suit : la combinaison de toutes les caractéristiques, l'intersection d'histogrammes TSV, l'intersection d'histogrammes RVB, les moments TSV, les moments RVB, la texture.

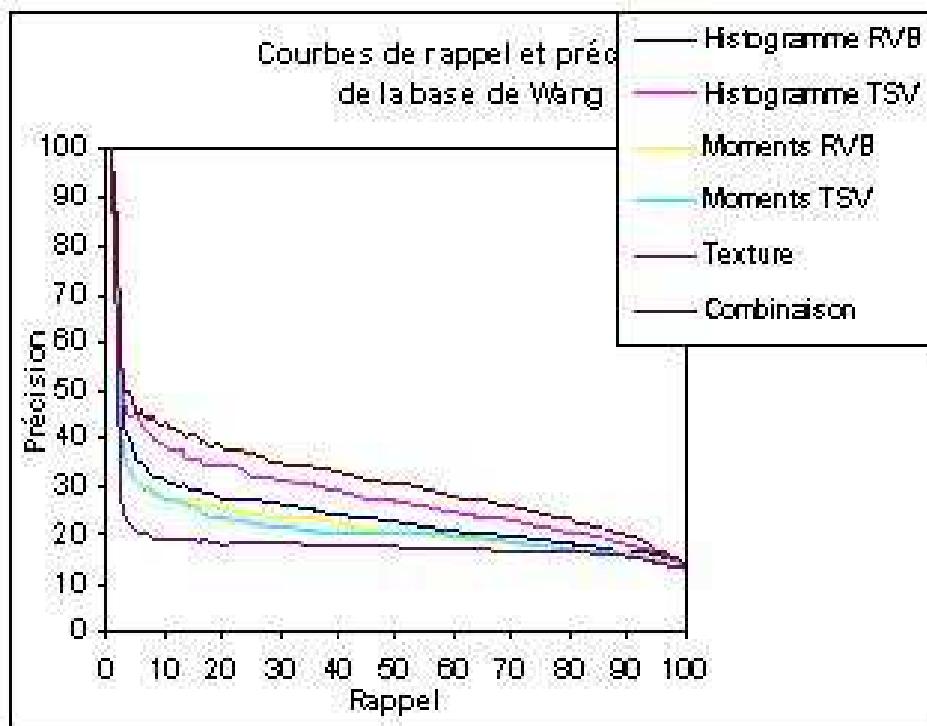


Fig. 2-20 : *Les courbes de rappel et précision avec la base de Wang avec les différentes caractéristiques*

La figure 2-21 permet de voir plus en détails le rôle de chaque caractéristique avec chaque classe de la base de Wang. En voyant cette figure, on trouve que la performance de chaque caractéristique avec chaque classe change. Parmi

les classes, les classes 4, 6 et 7(contenant des dessins de dinosaures, des fleurs en gros plans, des chevaux) donnent toujours des résultats parfaits, mais les classes 1, 2 et 8(contenant des paysages de mer, urbains ou de montages) donnent des mauvais résultats. Le reste donne des résultats assez bons. A partir de ces courbes, nous donnons quelques remarques comme suit :

- La combinaison donne les meilleurs résultats pour toutes les classes. La performance du système augmente avec le nombre de caractéristiques utilisées.
- Nous pouvons donner des poids différents pour chaque caractéristique correspondant à leurs performances pour chaque classe. Par exemple avec la classe 7, l'intersection d'histogrammes TSV et les moments RVB donnent les meilleurs résultats qu'avec l'intersection RVB, les moments TSV et la texture alors que nous choisissons des poids élevés pour l'intersection d'histogrammes TSV et les moments RVB et des poids faibles pour l'intersection d'histogrammes RVB, les moments TSV et la texture.

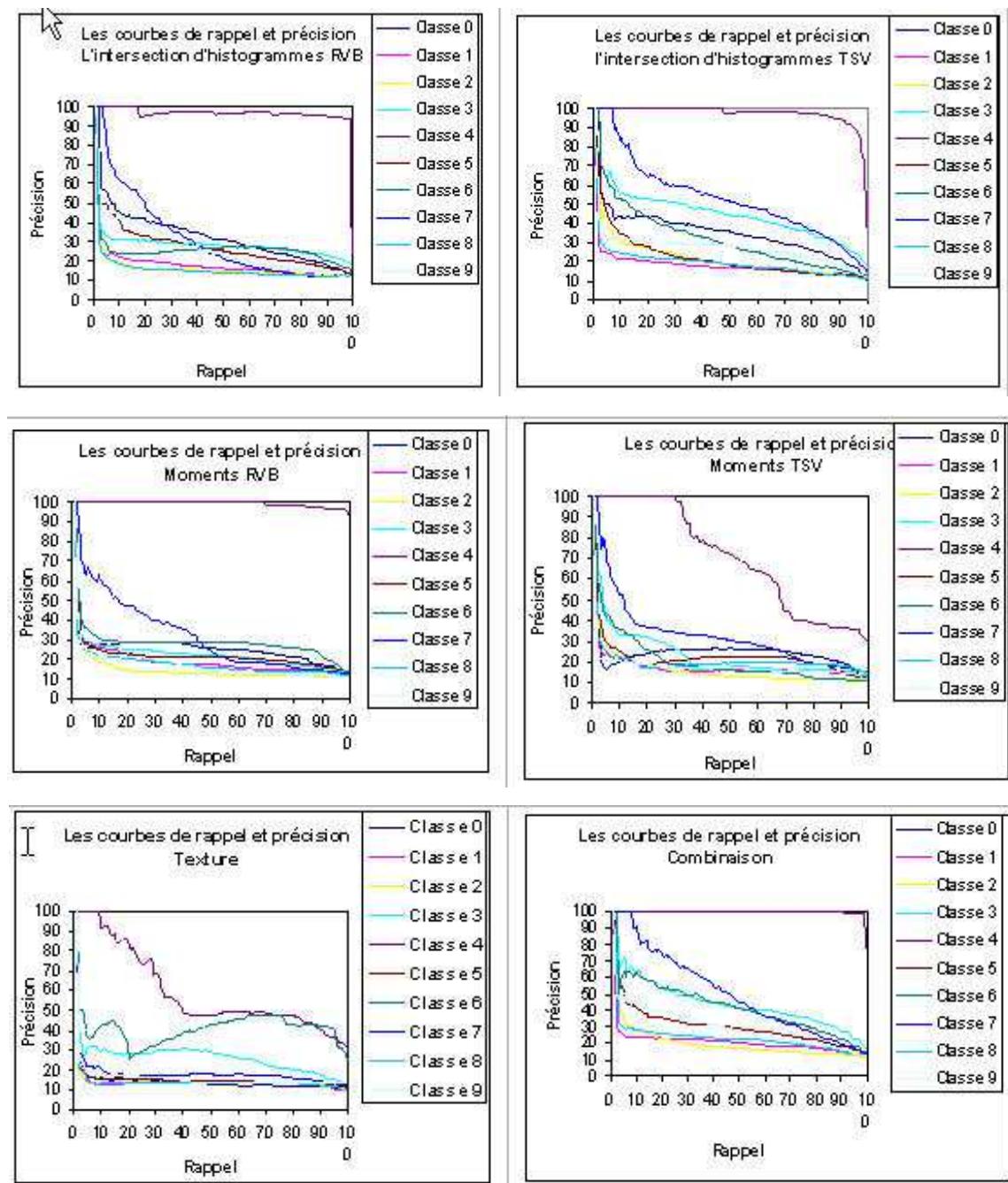
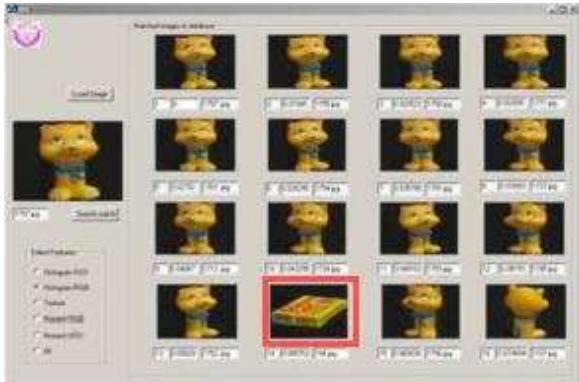


Fig. 2-21 : *Les courbes de rappel et précision de chaque classe de la base de Wang avec les différentes caractéristiques*

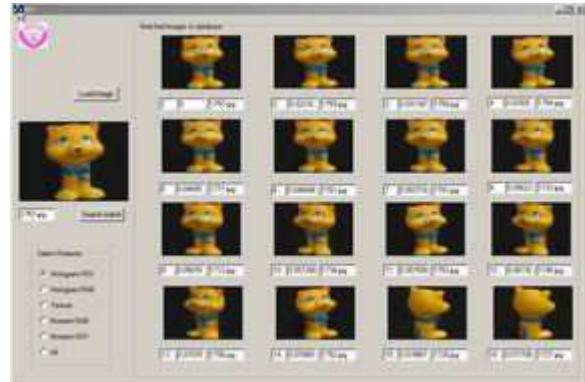
La base Coil-100

La figure 2-22 montre quelques résultats avec la base Coil et le tableau 2-2 contient les valeurs de rappel et précision calculées à partir de ces résultats. Avec cet exemple, il semble que toutes les caractéristiques sont très

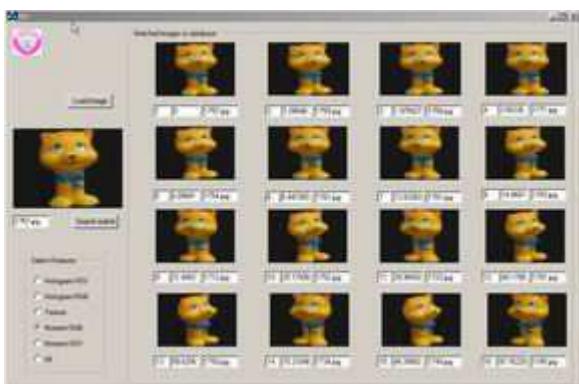
fortes (sauf la texture). La figure 2-22 donne les courbes totales pour toute la base.



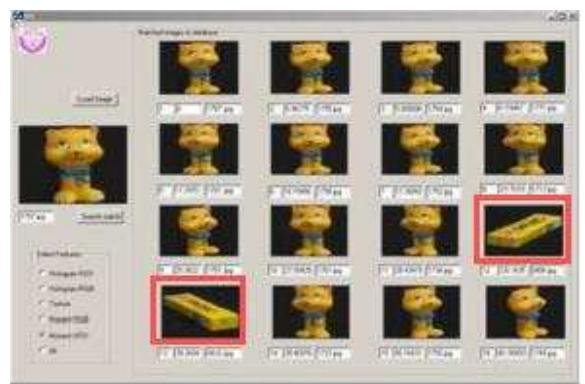
L'intersection d'histogrammes
RVB 24 bin



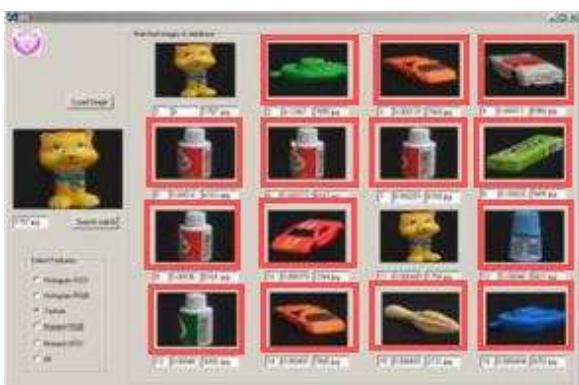
L'intersection d'histogrammes
TSV 24 bin



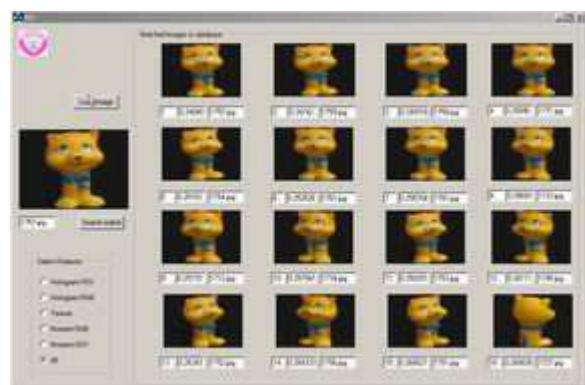
Les moments RVB



Les moments TSV



La texture



La combinaison

Fig. 2-22 : Quelques résultats avec la base Coil, des images en rouge sont des images non-pertinentes.

Descripteur	Nombre d'images pertinentes dans 16 images	Précision ($ Ra / A $) $ A = 16$	Rappel ($ Ra / R $) $ R = 72$
L'intersection d'histogrammes RVB	15	93.75%	20.83%
L'intersection d'histogrammes TSV	16	100%	22.22%
Les moments RVB	16	100%	22.22%
Les moments TSV	14	87.5%	19.44%
La texture	2	12.5%	2.77%
La combinaison	16	100%	22.22%

Avec $|R|=72$ et $|A|=16$, la précision maximale est de **100%**, le rappel maximal est de **22.22%**.

Tab 2-4 : *Les valeurs de rappel et précision de la base de Coil où le nombre d'images retrouvées est de 16.*

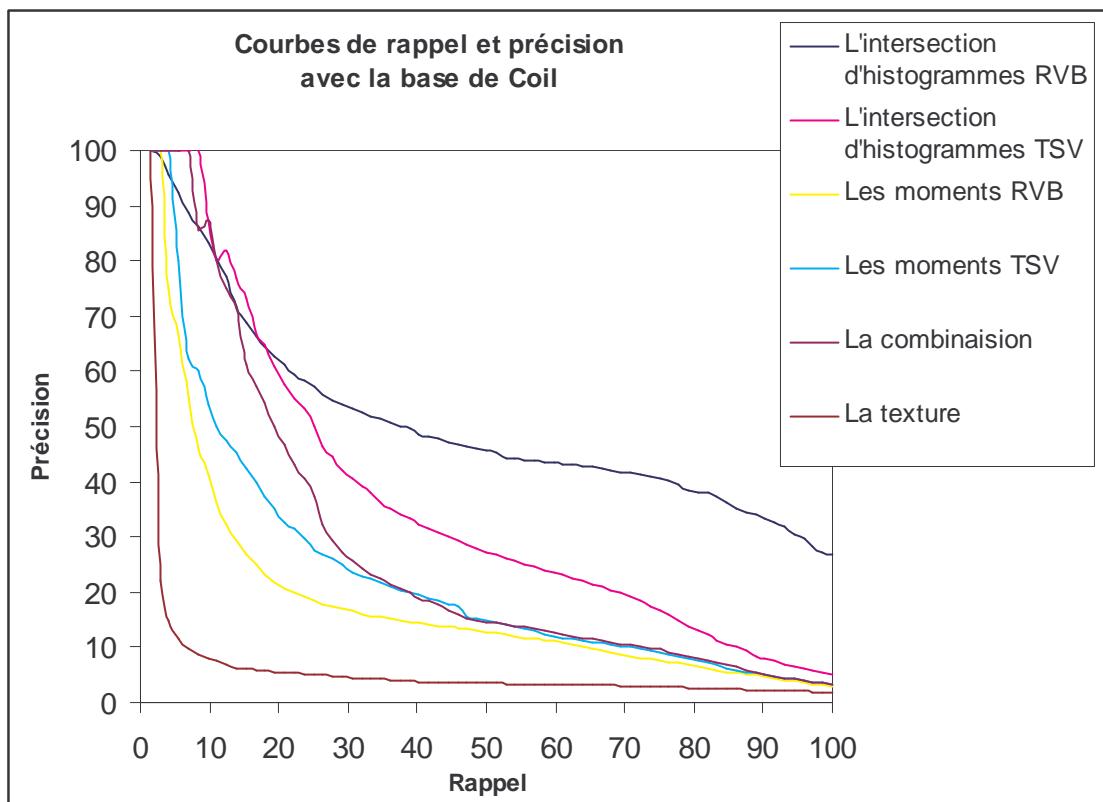


Fig. 2-23 : *Les courbes de rappel et précision avec la base de Coil 100 avec les différentes caractéristiques*

En comparant avec les résultats de la base de Wang, nous trouvons que les résultats avec la base Coil sont meilleurs que ceux avec la base de Wang. Les meilleurs résultats avec la base Coil sont obtenus avec, triées en ordre décroissant, l'intersection d'histogrammes RVB, l'intersection d'histogrammes TSV, la combinaison de toutes les caractéristiques, les moments RVB, les moments TSV et la texture. Cependant avec la base de Wang, les meilleurs résultats sont obtenus avec, en ordre décroissant la combinaison de toutes les caractéristiques, l'intersection d'histogrammes TSV, l'intersection d'histogrammes RVB, les moments TSV, les moments RVB, la texture. Les décalages de résultats entre les caractéristiques avec la base de Coil sont plus grands que ceux avec la base de Wang. Nous pouvons expliquer ces observations par le fait que les images dans la base Coil sont

plus simples que celles dans la base de Wang. Dans la base Coil, les images contiennent un seul objet en avant avec le fond derrière. Ces images ne contiennent pas de scènes compliquées comme par exemple la plage ou la montagne.

La base de Pollens

Cette base ne contient que 347 images de pollens divisées en 33 classes. Le nombre d'images dans chaque classe est différent. Donc la forme des courbes de rappel et précision est différente que celle d'autres bases d'images. Les résultats avec cette base sont très bons. La combinaison de toutes les caractéristiques est la plus forte. Les rôles de l'intersection d'histogrammes TSV, l'intersection d'histogrammes RVB, les moments RGB sont presque égaux.

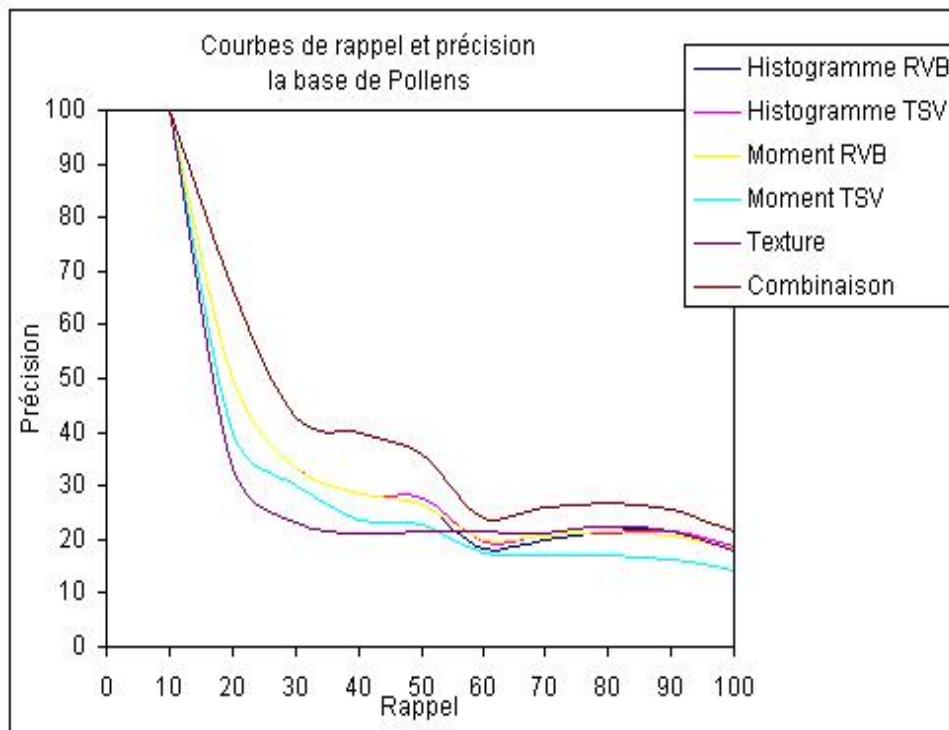
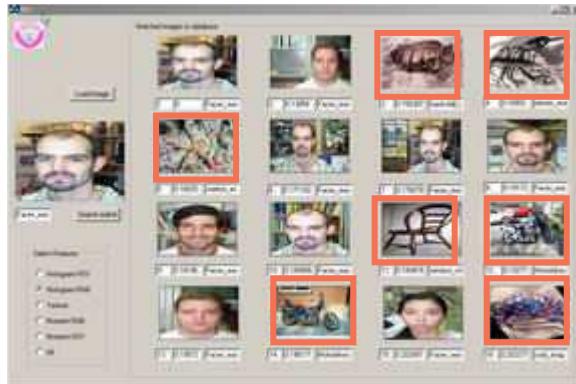


Fig. 2-24 : *Les courbes de rappel et précision pour la base de Pollens avec différentes caractéristiques*

La base de Fei-Fei

La figure 2-25 donne un exemple avec la base de Fei-Fei. L'image grande et à gauche est l'image requête. Parmi les seize images résultats, les images en rouge sont des images non-pertinentes. Dans cette base, l'auteur divise les images contenant le visage en deux classes : la classe des visages faciles (contenant des grands visages) et la classe des visages difficiles (contenant des petits visages). Cette division est utile pour la reconnaissance des formes. Avec la recherche d'images, nous pensons que des images appartenant aux deux classes sont des images pertinentes puisque si on veut trouver une image contenant un visage, on ne distingue pas un grand visage d'un petit visage.



L'intersection d'histogrammes RVB

24 bin

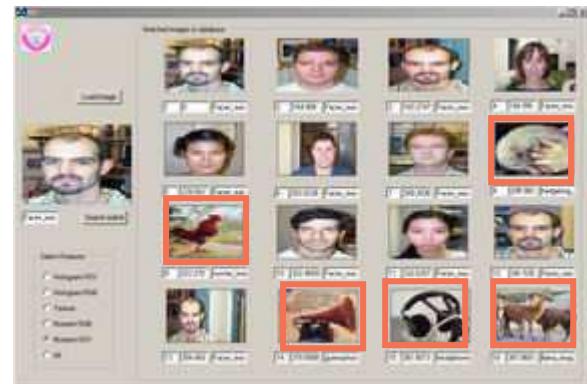


L'intersection d'histogrammes TSV

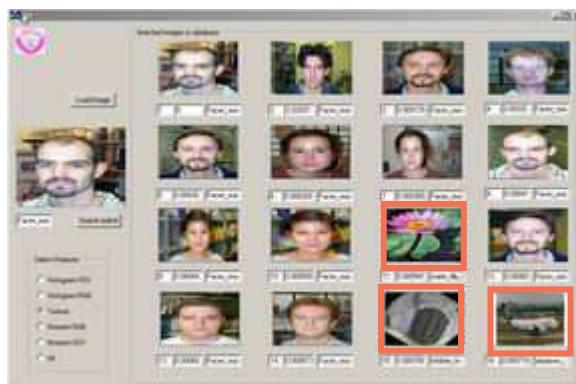
24 bin



Les moments RVB



Les moments TSV



La texture



La combinaison

Fig. 2-25 : *Quelques résultats avec la base de FeiFei, les images en rouge sont des images non-pertinentes*

A partir ces résultats, nous calculons des valeurs de rappel et précision

Descripteur	Nombre d'images pertinentes dans 16 images	Précision ($ Ra / A $) $ A =16$	Rappel ($ Ra / R $) $ R =870$
L'intersection d'histogrammes RVB	9	56.25%	1.03%
L'intersection d'histogrammes TSV	13	81.25%	1.49%
Les moments RVB	14	87.5%	1.609%
Les moments TSV	11	68.75%	1.264%
La texture	13	81.25%	1.49%
La combinaison	14	87.5%	1.609%

Avec $|A|=16$ et $|R|=870$, la précision maximale est de 100%, le rappel maximal est de 1.839%.

Tab 2-5 : *Les valeurs de rappel et précision de la base de FeiFei, où le nombre d'images retrouvées est de 16.*

La figure 2-26 donne les courbes de rappel et précision pour la classe de Moto. Cette classe contient 791 images. Nous trouvons que avec cette classe, l'intersection d'histogrammes RVB, l'intersection d'histogrammes TSV et la combinaison sont les meilleures caractéristiques.

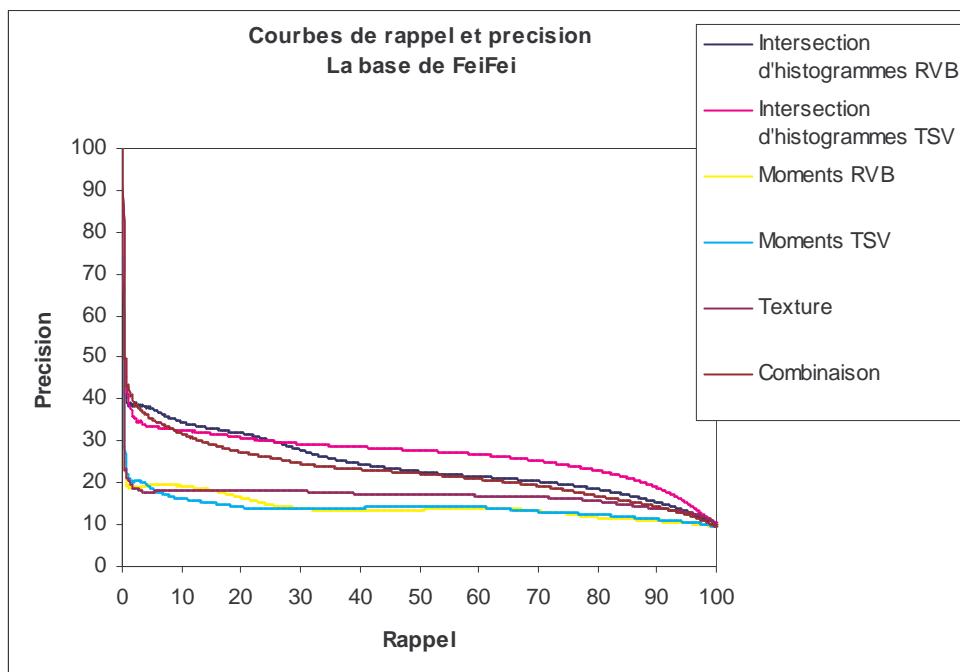


Fig. 2-26 : *Les courbes de rappel et précision avec la base de Fei-Fei avec les différentes caractéristiques*

A partir de ces résultats avec quelques bases d'images différentes, nous donnons quelques remarques comme suit :

- Avec la plupart des bases d'images, la combinaison de toutes les caractéristiques est la meilleure caractéristique.
- L'intersection d'histogrammes donne de légèrement meilleurs résultats que les moments. Mais avec les moments, on peut les calculer et comparer des images plus rapidement.
- La texture avec les nombres des matrices de co-occurrences n'est pas très forte pour les bases d'images testées à cause de deux raisons. Premièrement, la comparaison entre deux images par les mesures de texture est difficile. Deuxièmement, quatre chiffres des matrices de co-occurrences ne sont pas suffisants pour identifier la texture. Il manque d'autres caractéristiques telles que les coefficients de Gabor, Tamura...

Les résultats avec plusieurs images requêtes

Nous appliquons les techniques que nous avons présentées dans la section 2.2.3 à notre système. Pour faciliter le calcul, nous choisissons $n_l=n$ et $w_l=1$ (l'importance de chaque image requête est égale). Nous avons testé avec la base de Wang. La figure 2-27 montre nos résultats quand nous utilisons une seule image requête (qui contient une fleur jaune, grande et au milieu de la figure). A chaque itération nous allons rajouter une autre image requête.

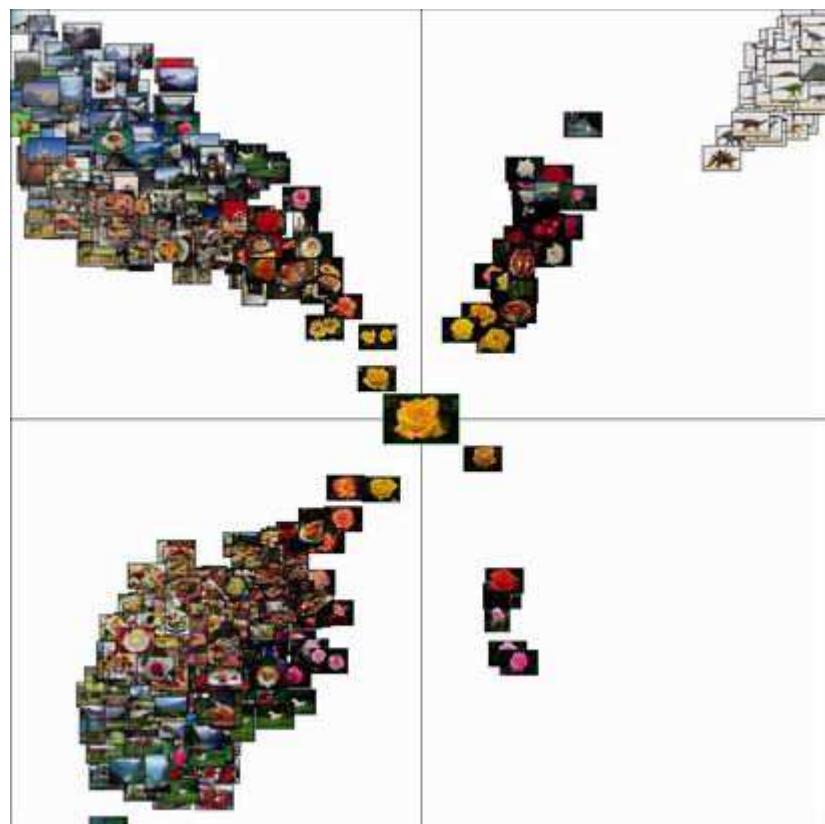


Fig. 2-27 : *Résultat avec une seule image requête, qui est grande et au milieu de la figure.*

Les figures 2-28 et 2-29 montrent les résultats quand nous rajoutons une autre image requête (contenant une fleur rouge). Nous trouvons qu'avec cette requête la méthode de combinaison par distance minimale semble mieux que la méthode de combinaison linéaire parce que ses résultats contiennent plus d'images pertinentes (une image est pertinente si elle contient une ou plusieurs fleurs jaunes ou rouges). Dans ces résultats, les images requêtes ont

été indiquées par un point vert au milieu et elles sont aussi montrées dans le coin inférieur gauche

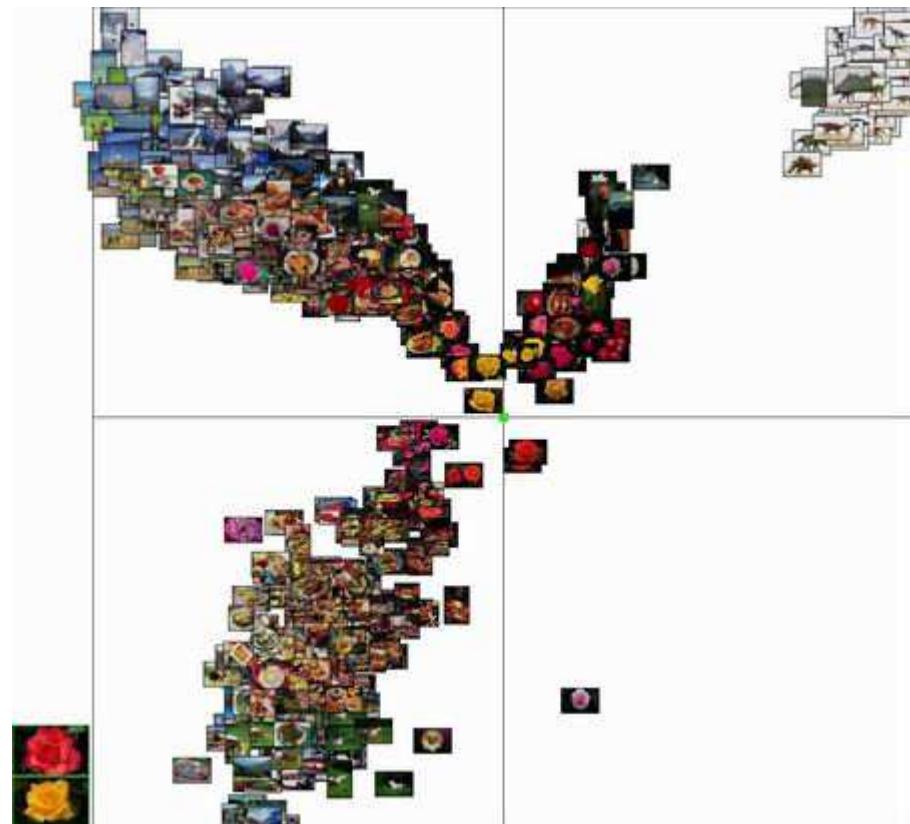


Fig. 2-28 : *Le résultat avec la méthode de combinaison par distance minimale en rajoutant une image avec les images requêtes indiquées par un point vert au milieu et montrées dans le coin inférieur gauche*

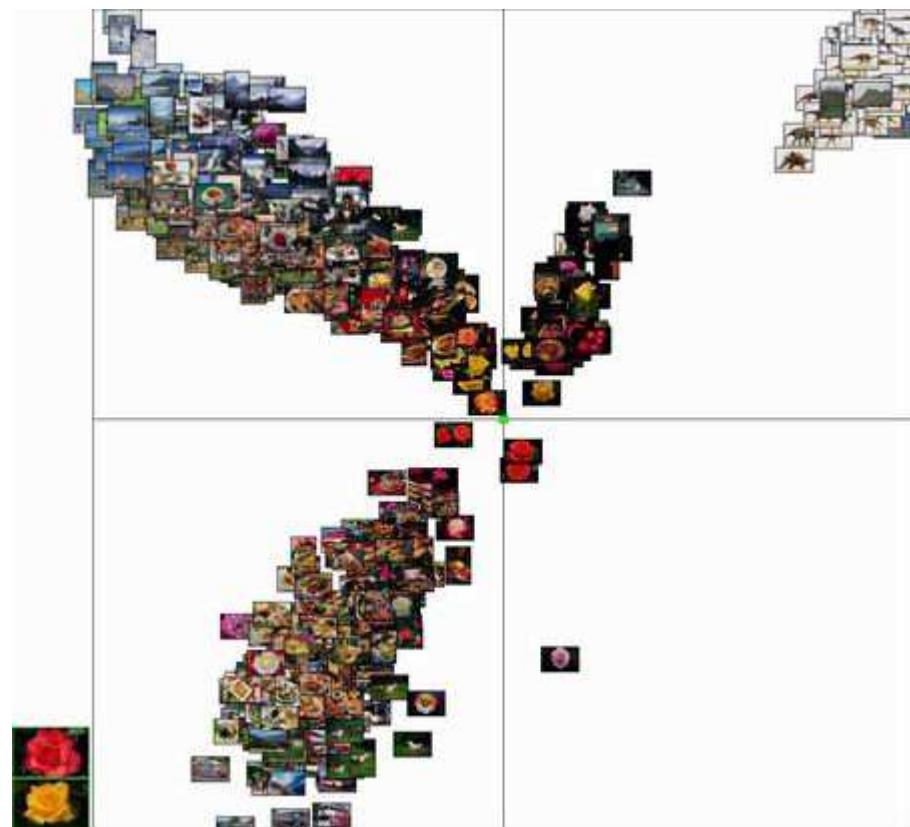


Fig. 2-29 : *Le résultat avec la méthode de combinaison linéaire en rajoutant une image, avec les images requêtes indiquées par un point vert au milieu et montrées dans le coin inférieur gauche*

Les figures 2-30 et 2-31 montrent les résultats quand nous rajoutons deux autres images requêtes (l'une contient une fleur rouge et l'autre contient une fleur jaune). Avec ces requêtes la méthode de combinaison linéaire semble mieux que la méthode de combinaison par distance minimale parce que dans ces résultats beaucoup d'images qui contiennent des fleurs rouges ou des fleurs jaunes se trouvent proche des images requêtes.

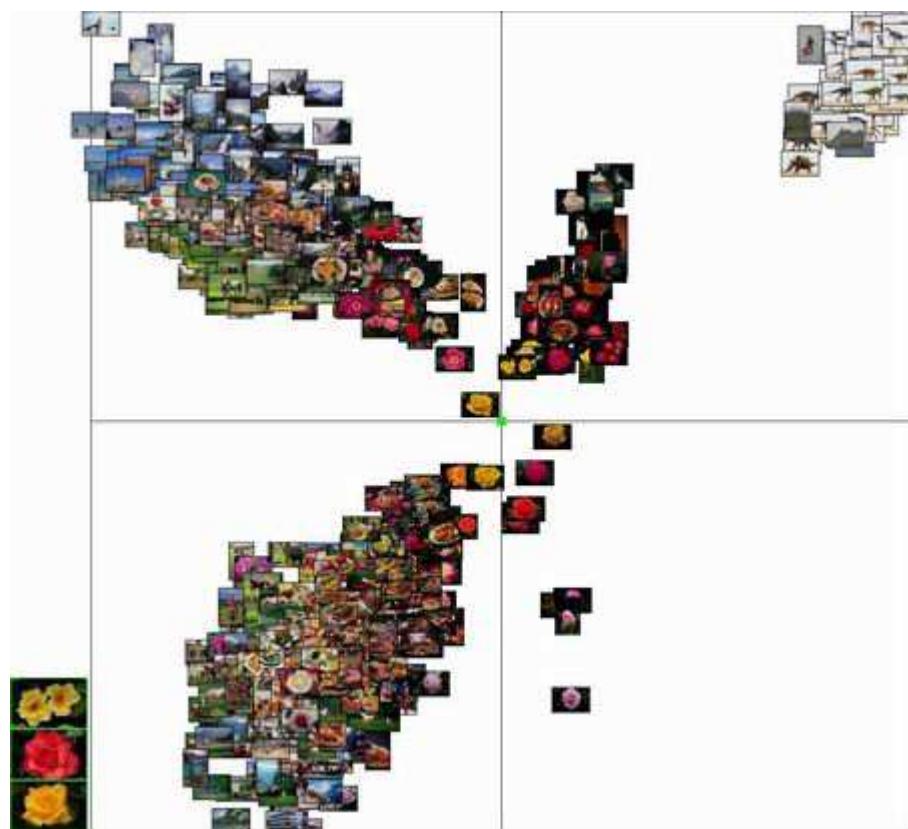


Fig. 2-30 : *Le résultat avec la méthode de combinaison par distance minimale en rajoutant deux images, avec les images requêtes indiquées par un point vert au milieu et montrées dans le coin inférieur gauche*

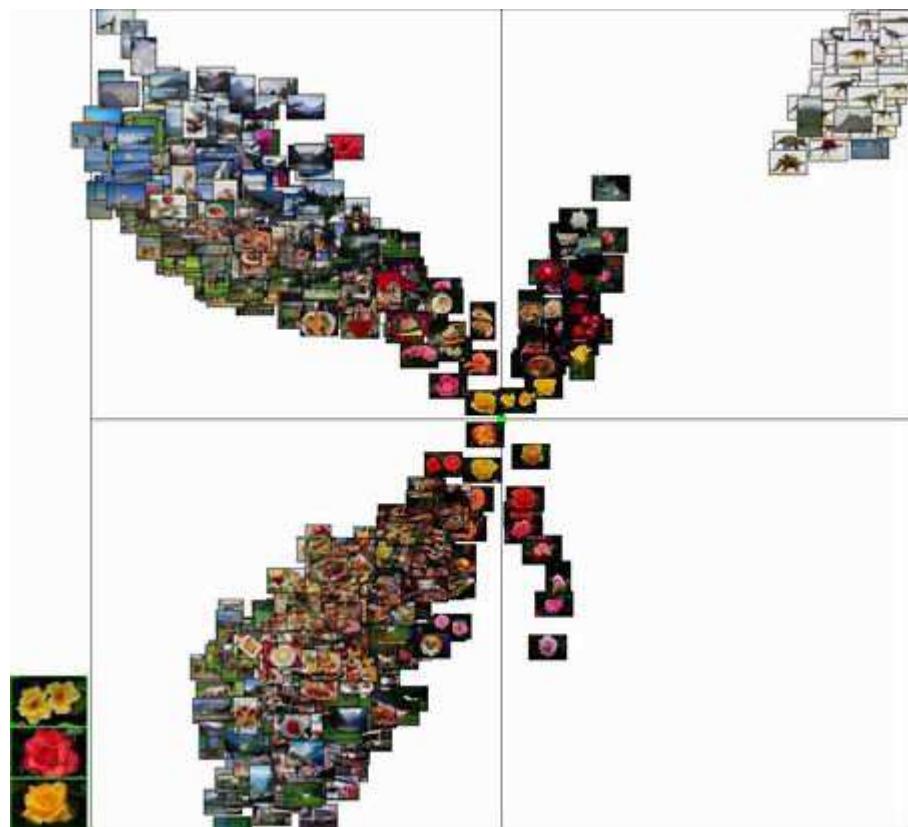


Fig. 2-31 : *Le résultat avec la méthode de combinaison linéaire en rajoutant deux autres images, avec les images requêtes indiquées par un point vert au milieu et montrées dans le coin inférieur gauche*

Avec seulement quelques résultats, nous ne pouvons pas décider quelle méthode est meilleure. Chacune a des avantages ainsi que des inconvénients. Par conséquent, avec chaque base d'images, on doit tester les deux méthodes et utiliser des mesures pour évaluer chaque méthode. Par manque de temps pour tout calculer, nous ne donnons pas les résultats finaux (courbes de rappel et précision) dans ce rapport. Cela sera fait ultérieurement.

Chapitre 3: Recherche d'images par le contenu sémantique

3.1. Etat de l'art

Après avoir présenté les aspects d'un système de recherche d'images par le contenu symbolique, ainsi que notre système utilisé pour la recherche, nous présentons maintenant les extensions possibles d'un tel système pour traiter les images au niveau sémantique, c'est-à-dire au niveau du jugement que l'humain porte en regardant les images. C'est à cette condition que la recherche d'informations sur les images pourra déboucher sur de réelles avancées répondant à des besoins et applications diverses, tant personnelles que professionnelles.

Raisonner au niveau sémantique signifie que l'analyse de l'image se fait en termes d'objets, de contenu et de structure, et non pas seulement en termes de statistiques sur les couleurs, les textures ou autres caractéristiques de base de l'image. Ce qui demande un complément d'information certain à la méthode, car par définition, seules les caractéristiques de base sont disponibles immédiatement dans l'image. La sémantique elle-même n'est pas inscrite dans l'image, mais se trouve ailleurs. Il faut donc rechercher ces sources extérieures nous donnant accès aux clés de décodage sémantique de l'image.

Nous soutenons que la sémantique exprimée dans une image dépend de deux éléments (Jiang, 2004) :

- du niveau de connaissances et de la perception qu'a l'observateur de cette image ;
- de l'objectif poursuivi par l'utilisateur de cette image lorsqu'il la regarde.

Et c'est pourquoi cette sémantique doit être retrouvée selon deux approches complémentaires et indissociables : (1) l'une recherche des méthodes pour comprendre l'objectif de l'utilisateur, le sens de sa requête, (2) l'autre recherche des moyens pour « connecter » (ou lier) la connaissance sémantique humaine et l'apparence de l'image (les caractéristiques extraites de celle-ci). Et c'est avec la combinaison des deux approches que l'on pourra faire émerger la sémantique de l'image

Nous poursuivons cette présentation par l'étude de ces deux approches : (1) l'interaction avec l'utilisateur (le retour de pertinence) et (2) la définition des concepts de l'image (nous nous concentrerons sur l'ontologie et son application pour la recherche des images). Nos recherches sur ce sujet n'étant pas encore complétées, nous présentons dans la suite certains éléments seulement de notre prototype de recherche d'images par le contenu sémantique basé uniquement, pour l'instant, sur le retour de pertinence. Dans la section 3.2, nous allons montrer nos résultats préliminaires.

3.1.1. Retour de pertinence

Une voie de plus en plus utilisée pour pallier le manque de connaissances dans un système de vision par ordinateur est de concevoir des systèmes semi-automatiques. Cette voie existant déjà pour d'autres domaines, mais devient encore plus évidente dans le domaine de l'indexation et recherche d'images. Ici, on suppose que l'humain maîtrise la sémantique et qu'avec l'aide de quelques interactions simples et ciblées (par exemple, l'identification d'exemples positifs et négatifs à partir d'un premier retour du programme), le système peut prendre cette sémantique. Il est connu qu'une même image requête présentée par l'utilisateur peut avoir différentes significations selon les besoins immédiats de cet utilisateur (objet avant-plan,

arrière-plan). La théorie prédominante est celle du retour de pertinence (relevance feedback en anglais) déjà utilisée dans le domaine de la recherche d'informations. Dans ce cas-ci, la sémantique n'est pas dans le système informatique, mais elle est extraite implicitement à chaque coup via l'interaction et l'analyse des retours.

Le processus de retour de pertinence peut se décrire comme suit : quand une requête est faite et qu'une réponse est donnée par le système, l'utilisateur donne plus d'indications au système comme des images positives et des images négatives à partir des premiers résultats. Le système recalcule et donne de nouveaux les deuxièmes résultats. L'utilisateur peut répéter ce processus plusieurs fois jusqu'à ce qu'il soit satisfait. Les images positives sont les images que l'utilisateur considère comme les images qui sont semblables à la requête et inversement pour les images négatives.

La figure 3-1 donne un exemple de retour de pertinence dans le système d'indexation et recherche d'images. Dans cet exemple, on a répété deux fois le processus de retour de pertinence et on a vu que parmi les images retrouvées, le nombre d'images pertinentes a augmenté.

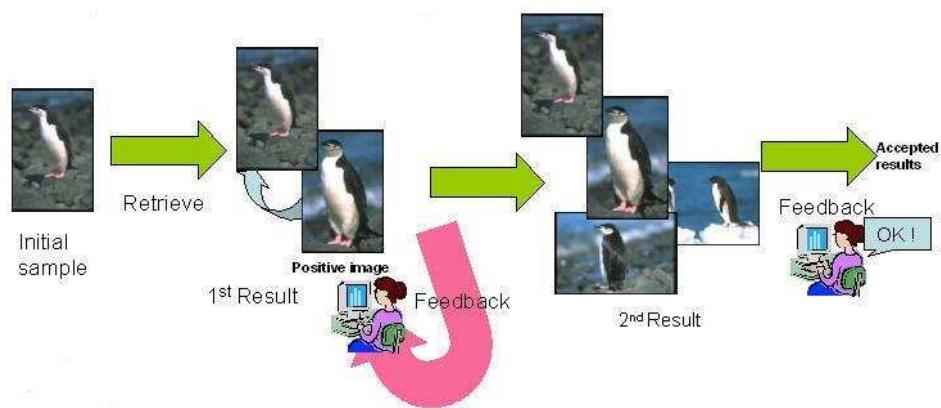


Fig. 3-1 : *Un exemple de retour de pertinence dans le CBIR, on a répété le processus de retour de pertinence deux fois et après chaque fois, le nombre d'images retrouvées pertinentes a augmenté.*

Il y a beaucoup de mécanismes proposés pour le retour de pertinence dans un système d'indexation et recherche d'images. Les premiers mécanismes viennent du système de recherche de textes. Dans ces mécanismes on rajoute des poids aux différentes caractéristiques et aux composants de chaque caractéristique pour exprimer leurs importances correspondantes. Dans (Rui, 1998), les auteurs les ont appliqués dans le système d'indexation et recherche d'images. Pour commencer la première fois, les poids ont été initialisés par même chiffre. Avec chaque image retrouvée, l'utilisateur peut la marquer comme une image très pertinente, pertinente, sans opinion, non pertinente et non pertinente pas du tout. Le système va recalculer les poids (inter-poids – les poids pour les caractéristiques et intra-poids – les poids pour les composants dans chaque caractéristique) en basant sur ces remarques.

Il y a d'autres mécanismes. Dans (Iqbal, 2003), les auteurs supposent que les images positives appartiennent à une classe, cependant les images négatives peuvent appartenir à plusieurs classes. Donc ils proposent deux mécanismes : le premier fonctionne une marche seulement avec des images positives et le deuxième avec les images positives ainsi que les images négatives. Le premier mécanisme est considéré comme le mécanisme appliqué pour le cas de plusieurs images requêtes avec la méthode de combinaison de la distance minimale que nous avons présentée dans la section 2.2.3 (ce mécanisme est appelé *Cluster Feedback*). Le deuxième mécanisme s'appelle *Multi-classe Feedback*. Ce mécanisme a été fréquemment utilisé dans le domaine de reconnaissance des formes. Mais il marche seulement dans le cas où les images pertinentes appartiennent à une classe et les images non pertinentes appartiennent à plusieurs classes. Dans

(Nakazato, 2002) et (Zhou, 2001), les auteurs ont aussi proposé d'autres mécanismes pour le cas où les images négatives appartiennent à une ou plusieurs classes, basés sur l'Analyse discriminante de Fisher (*FDA - Fisher's Discriminant Analysis*) et l'Analyse discriminante du biais (*BDA - Biased Discriminant Analysis*). Parfois les images positives appartiennent à plusieurs classes. Par exemple, quand l'utilisateur recherche des images dans son esprit il a des concepts hauts niveaux tel que : « des belles fleurs». Ce concept ne peut pas être représenté par une classe. Cependant des fleurs blanches et des fleurs rouges ont des caractéristiques communes, elles ont aussi des caractéristiques différentes tel que la couleur. Si le système essaie de les considérer comme une seule classe, les caractéristiques de couleur auront disparues bien que la couleur soit une caractéristique importante (et dominante) dans la recherche de ces fleurs. Dans (Nakazato, 2002), avec une interface originale (Image Grouper), les auteurs ont proposé une variation de BDA pour le cas où les images positives appartiennent à plusieurs classes.

On sait que le système d'indexation et recherche d'images au niveau région a les propriétés que les systèmes d'indexation et recherche d'images au niveau image n'ont pas. Donc on veut aussi développer les mécanismes de retour de pertinence pour ce système. Dans (Jing, 2001), les auteurs ont essayé d'appliquer les mécanismes de retour de pertinence pour les systèmes d'indexation et recherche d'images au niveau image. Mais ces mécanismes ne sont pas vraiment des mécanismes de retour de pertinence au niveau image. Tout d'abord, ils ont appliqué les mécanismes de retour de pertinence au niveau image (les images positives ainsi que les images négatives) et ont estimé l'importance des régions. Les régions sont importantes si elles sont présentes dans beaucoup d'images positives. Ensuite, ils ont comparé les images en comparant les régions. Dans ce système, l'utilisateur ne peut pas

indiquer une région positive ou une région négative. A l'heure actuelle, il n'y a pas beaucoup de recherche sur les régions négatives et régions positives. Cette voie promet d'avoir de bons résultats mais il reste des problèmes à résoudre.

Le retour de pertinence a des avantages, mais avec l'interface traditionnelle, cela n'est pas confortable pour l'utilisateur, parce que l'utilisateur doit cliquer plusieurs fois pour choisir des images positives et négatives. C'est pourquoi dans notre système, nous utilisons les mécanismes déjà existants et essayons de résoudre leurs inconvénients en donnant une autre interface qui peut faciliter et diminuer les opérations de l'utilisateur.

Cette approche attire beaucoup d'intention, beaucoup de mécanismes ont été proposés mais il manque un mécanisme robuste et nous croyons qu'avec seulement les mécanismes de retour de pertinence, l'utilisateur ne peut pas être satisfait. On doit les intégrer avec d'autres mécanismes.

3.1.2. *Ontologie*

L'ontologie vient du domaine de la philosophie et attire de plus en plus l'attention des personnes dans le domaine informatique. Dans cette section, tout d'abord, nous présentons la notation d'ontologie, le rôle d'une ontologie et le processus de création d'une ontologie. Ensuite, nous nous concentrerons sur son application dans un système d'indexation et recherche d'images.

Selon (Mezaris, 2003) une ontologie peut être définie comme des spécifications d'un vocabulaire de représentation pour un domaine partagé du discours qui peut inclure des définitions de classes, des relations, des fonctions et d'autres objets.

Pourquoi veut-on développer une ontologie? À quoi sert-elle, l'ontologie? Il y a quelques raisons (Chandrasekaran, 1999). Premièrement,

l'ontologie permet de partager la connaissance commune d'un domaine entre des personnes ou des agents logiciel. C'est l'un des buts les plus communs dans le développement des ontologies. Par exemple, supposons que quelques sites web différents contiennent des informations médicales. Si ces sites Web partagent et publient avec les mêmes termes de l'ontologie, alors l'agent logiciel est capable d'extraire des informations à partir des différents sites. Ces informations peuvent être utilisées pour répondre à des requêtes de l'utilisateur. Deuxièmement, l'ontologie permet de réutiliser la connaissance d'un domaine. Par exemple, plusieurs domaines ont besoin de représenter la notion du temps. Il s'agit par exemple des notions intervalles du temps, des points du temps, des mesures relative du temps. Si un groupe de chercheurs développe une telle ontologie, d'autres peuvent simplement la réutiliser pour d'autres applications. Troisièmement, l'ontologie permet de faire explicitement des hypothèses sur un domaine. Des hypothèses codées dans un langage de programmation sont difficiles à trouver, à comprendre et aussi à changer en particulier pour des utilisateurs sans compétence de programmer. Avec l'ontologie, en jouant avec des hypothèses, on peut comprendre mieux le domaine. Quatrièmement, l'ontologie permet de séparer la connaissance d'un domaine avec la connaissance de fonctionnement. Par exemple, nous pouvons utiliser une ontologie pour décrire une tâche d'une configuration d'un produit. Cette configuration est indépendante de produit et de ses composants. Nous pouvons développer une ontologie pour la configuration d'un ordinateur, et la réutiliser pour la configuration d'un élévateur. Enfin, l'ontologie permet d'analyser la connaissance d'un domaine. Grâce à l'ontologie, une analyse formelle des termes est possible. Cette analyse est extrêmement importante pour la réutilisation et l'extension d'ontologies existantes.

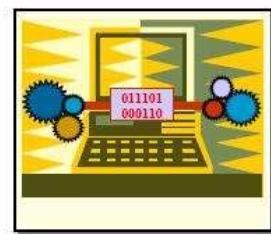
Dans (Gandon, 2002), l'exemple donné par Gomez-Pérez à la conférence ECAI98 est représentatif du problème d'ambiguïté sur un terme lors d'une communication dans un groupe d'acteurs. Que faut-il répondre à la question : » What is a pipe ? ». Il existe plusieurs possibilités de réponse tel qu'illustré à la figure 3-2:



(a)



(b)



(c)

Fig. 3-2 : *Trois réponses possibles de la question « What is a pipe ? » [Gandon02]*

- (a) « *a short narrow tube with a small container at one end, used for smoking eg. tobacco.* »
- (b) « *a long tube made of metal or plastic that is used to carry water or oil or gas.* »
- (c) » *a temporary section of computer memory that can link two different computer processes.* »

Dans cet exemple, nous avons un terme avec trois concepts. Donc cela provoque une ambiguïté. Par contre, dans le cas de synonymes, un concept peut être derrière quelques termes : par exemple voiture, automobile, auto... Ce sont des problèmes qui sont posés quand nous voulons informatiser un système. L'engagement ontologique des acteurs permet d'éviter ce genre de confusion.

Nous venons de présenter la notation de l'ontologie et ses avantages dans le domaine informatique. Nous allons poursuivre cette section par le

processus de création d'une ontologie, les critères utilisés pour évaluer une ontologie et ses applications.

Le cycle de vie d'une ontologie contient trois types d'activités :

- Activités de construction : Ce sont les quatre étapes de construction d'une ontologie.
- Activités de support : Ce sont l'acquisition des connaissances, la documentation, l'évaluation et l'intégration de l'ontologie.
- Activités de gestion : Ce sont la planification des tâches, le contrôle et l'assurance de la qualité.

Les activités de construction d'une ontologie jouent un des rôles les plus importants dans le cycle de vie d'une ontologie. Ces activités se décomposent en quatre phases : la phase de spécification, la phase de conceptualisation, la phase de formalisation et celle d'implémentation.

La phase de spécification :

Selon (Noy, 2001), cette étape doit répondre aux questions suivantes :

- Quel est le domaine que l'ontologie couvrira ?
- Quels sont les buts de l'utilisation de l'ontologie ?
- Quels sont les types de questions que l'ontologie peut répondre ?
- Qui va utiliser et maintenir l'ontologie ?

La phase de conceptualisation :

La phase conceptualisation a pour but la structuration des connaissances d'un domaine. Cette phase se compose des étapes suivantes :

- Identifier les concepts, les attributs, et les valeurs dans un *Dictionnaire Glossaire*.
- Classifier les groupes de concepts dans des *Arbres de Concepts*.
- Construire une *Table des Relations*.

- Définir les constantes dans une *Table des Constantes*
- Construire une *Table de Formules*.

Les phases de formalisation et d'implémentation

La phase de formalisation va mettre le modèle conceptuel structuré sous une forme semi-formelle de l'ontologie. Dans la phase d'implémentation, on va implémenter une ontologie dans un langage. Le langage choisi doit correspondre avec le modèle de formalisation.

Après avoir créé une ontologie, un processus pour évaluer cette ontologie est nécessaire. Les critères utilisés pour ce processus ont été proposés en se basant sur le but de création d'une ontologie : le partage des connaissances et l'inter-opérabilité. Dans (Gruber, 1993), Gruber a proposé cinq critères pour l'évaluation d'une ontologie qui sont la clarté, la cohérence, l'extensibilité, l'indépendance structurelle et la minimalisation de l'engagement ontologique.

L'ontologie peut être utilisée dans plusieurs applications. Mais dans ce rapport, nous nous concentrerons sur son application dans le domaine de la recherche d'images. Il existe deux approches de recherche d'images se basant sur des ontologies. L'une construit une ontologie basée sur les annotations et l'utilise pour recherche d'images. L'autre propose des ontologies de concepts. Ces concepts sont basés sur les données numériques qui viennent du traitement d'images bas niveau.

La première approche (Hyvonen, 2003) donne des bons résultats puisqu'on se base sur les annotations (recherche textuelle sans utiliser l'image). Mais cette ontologie a des inconvénients puisqu'elle demande beaucoup d'efforts pour faire l'annotation. Dans (Hyvonen, 2003), les auteurs ont créé une ontologie pour la base d'images du Helsinki University Museum. Ils l'ont utilisée pour la phase d'annotation ainsi que pour la phase de recherche d'images. Les auteurs ont constaté deux problèmes avec leur

ontologie. Quand on fait les annotations, cela peut provoquer des changements dans l'ontologie. Donc comment gère-t-on ces changements pour ne pas demander à l'annotateur de refaire les annotations ? Quand les programmeurs demandent l'ontologie et les annotations de nouvelles demandes pour satisfaire l'utilisateur, l'ontologie ainsi que les annotations doivent être changées.

La deuxième approche (Mezaris, 2003 ; Maillot, 2005 ; Maillot, 2004) essaie de commencer des données numériques qui sont extraites de l'image pour créer les concepts. Dans (Mezaris, 2003), les auteurs ont proposé une ontologie d'objets. Les caractéristiques bas niveau telles que la couleur, la position, la taille, etc. sont extraites d'une région et sont reliées avec les descripteurs du niveau intermédiaire.

Dans (Maillot, 2004), les auteurs ont créé une ontologie de concepts visuels. Cette ontologie contient 103 concepts visuels (concepts spatiaux, concepts couleur, concepts texture). Ces concepts ont été créés par une phase d'apprentissage. Ils sont représentés par un arbre de concepts visuels. Les auteurs ont essayé de l'appliquer au système de recherche d'images. Une base d'images avec 850 images a été utilisée pour faire un exemple d'apprentissage, une base contenant 2000 images a été utilisée pour tester. Les résultats obtenus sont très bons.

3.2. Notre système de recherche d'images par le contenu sémantique

3.2.1. Le retour de pertinence avec l'interface traditionnelle

Cette section se consacre à notre système d'indexation et recherche d'images par le contenu sémantique. Ce système n'est pas encore complètement terminé. Dans la section précédente, nous avons introduit deux approches pour ajouter de la sémantique : l'une est le retour de pertinence et

l'autre est l'ontologie. Mais maintenant, dans notre système, nous nous concentrons sur la partie de retour de pertinence. Nous nous intéressons aussi à la partie de l'ontologie. Nous voulons les combiner pour améliorer la performance du système de recherche d'images. Dans le futur, nous voulons développer des techniques d'ontologie et les appliquer à notre système.

Nous avons appliqué deux méthodes de retour de pertinence différentes dans notre système. Correspondant à chaque méthode, nous avons développé deux interfaces. Nous les appelons l'interface traditionnelle et l'interface graphique. Notre objectif est d'utiliser les techniques déjà existantes et d'essayer de les combiner, et de rajouter des nouvelles idées pour les améliorer.

Cette section se structure comme suit : D'abord nous allons présenter l'interface traditionnelle et détailler la technique appliquée pour cette interface. Ensuite, nous allons montrer nos idées pour l'interface graphique en comparant avec d'autres interfaces développées par d'autres équipes de recherche d'images. La comparaison entre les deux interfaces et quelques conclusions vont terminer cette section.

La figure 3-3 montre notre interface traditionnelle. Dans cette figure, l'image en rouge est l'image requête, et les seize premières images retrouvées sont affichées en ordre croissant de distance avec la requête. Avec une image retrouvée, nous fournissons 3 options : *Rel* (si cette image est pertinente), *No-option* (pas de commentaire par défaut), *No-rel* (si cette image n'est pas pertinente). A partir de ces 16 premières images résultats, l'utilisateur va choisir des images pertinentes, non-pertinentes et cliquer sur le bouton « Feedback » pour exécuter le processus de retour de pertinence.



Fig. 3-3 : *L'interface traditionnelle qui fournit le retour de pertinence avec 3 niveaux : pertinente, neutre ou non-pertinente*

Nous venons de présenter notre interface. Maintenant, nous allons entrer dans le noyau du système : le processus de retour de pertinence. Nous appliquons d'abord la méthode (Rui, 1998) dont nous ne présentons pas ici tous les détails. Nous allons seulement présenter les morceaux principaux. Tout d'abord, nous présentons le modèle pour présenter une image avec ses caractéristiques. Ensuite, nous appliquons ce modèle au système d'indexation et recherche d'images en utilisant le retour de pertinence. Le problème le plus important à résoudre dans le retour de pertinence, la mise à jour les poids, sera décrit ensuite.

Modèle

Ce modèle est présenté comme suit :

$$O = O(D, F, R)$$

où :

- D : une image
- F = {f_i} : l'ensemble de caractéristiques visuelles bas niveau telles que la couleur, la texture, la forme.

- $R = \{r_{ij}\}$: l'ensemble de représentation pour chaque caractéristique f_i .

Par exemple : les moments et l'intersection d'histogrammes ont été utilisés pour représenter la couleur.

r_{ij} est un vecteur qui contient plusieurs composantes r_{ijk} .

$r_{ij} = [r_{ij1}, r_{ij2}, \dots, r_{ijk}]$, K est le nombre de composants dans r_{ij} .

Nous appliquons ce modèle au système d'indexation et recherche d'images en ajoutant un ensemble de mesures $M = \{m_{ij}\}$.

$$S(r_{ij}) = m_{ij}(r_{ij}, W_{ijk}) \quad (3.1)$$

où W_{ijk} est le poids pour chaque composant r_{ijk} .

$$S(f_i) = \sum_j W_{ij} S(r_{ij}) \quad (3.2)$$

où W_{ij} est le poids pour chaque ensemble de représentation f_i .

$$S = \sum_i W_i S(f_i) \quad (3.3)$$

où W_i est le poids pour chaque caractéristique f_i .

Avant de combiner les distances, nous avons toujours besoin de les normaliser. L'objectif est d'insister également sur l'importance de chaque composant des caractéristiques ainsi que sur chaque caractéristique. Dans cette section, nous ne détaillons pas les méthodes de normalisation. Nous nous concentrerons seulement sur les mises à jour des poids.

Mise à jour inter-poids

Les W_{ij} associés aux r_{ij} reflètent les désirs de l'utilisateur. Les différents poids permettent à l'utilisateur de spécifier les informations dont il a envie. Dans cette section, nous discutons la façon pour mettre à jour W_{ij} selon le retour de pertinence de l'utilisateur.

Nous appelons RT, l'ensemble des N_{RT} images qui sont les plus similaires selon la valeur de similarité S. On peut écrire RT sous la forme suivante :

$$RT = [RT_1, RT_2, \dots, RT_{N_{RT}}]$$

A chaque image dans les images retrouvées a été donnée une. Cette note peut prendre une des valeurs suivantes :

- 3 : si l'utilisateur trouve cette image très pertinente
- 1 : si l'utilisateur trouve cette image pertinente
- 0 : si l'utilisateur ne donne pas d'opinion de cette image
- -1 : si l'utilisateur trouve que cette image n'est pas pertinente
- -3 : si l'utilisateur trouve que cette image n'est pas du tout pertinente

Le système donne 5 niveaux de retour de pertinence aux utilisateurs. Le système qui fournit le plus grand nombre de niveaux va rendre de meilleurs résultats. Mais cela entraîne certains inconvénients pour l'utilisateur.

RT^{ij} sont les résultats si le système n'utilise que r_{ij} pour comparer entre des images.

$$RT^{ij} = [RT_1^{ij}, RT_2^{ij}, \dots, RT_l^{ij}, \dots, RT_{N_{RT}}^{ij}]$$

Afin de compter les poids pour r_{ij} , on initialise $W_{ij}=0$ et utilise le processus suivant :

$$\begin{aligned} W_{ij} &= W_{ij} + \text{note}_l && \text{si } RT_l^{ij} \text{ appartient à RT} \\ W_{ij} &= W_{ij} + 0 && \text{si } RT_l^{ij} \text{ ne appartient pas à RT} \\ l &= 0, \dots, N_{RT} \end{aligned} \quad (3.4)$$

Après ce processus si $W_{ij} < 0$ nous fixons $W_{ij}=0$. On normalise les poids par les formules suivantes :

$$W_{Tij} = \sum W_{ij} \quad (3.5)$$

$$W_{ij} = \frac{W_{ij}}{W_{Tij}} \quad (3.6)$$

Mise à jour intra-poids

W_{ijk} réfléchit l'importance de chaque composante r_{ijk} dans le vecteur r_{ij} . On suppose que l'utilisateur va choisir M' images (très) pertinentes à partir des images résultats. On va présenter M' vecteurs r_{ij} (K dimensions) de ces images sous forme d'une matrice (M' lignes et K colonnes). On constate que si les composants r_{ijk} des images pertinentes ont des valeurs proches, alors cela signifie que r_{ijk} est un bon indicateur pour le vecteur r_{ij} et inversement. Selon cette observation, l'inverse de l'écart-type de la séquence r_{ijk} est une bonne estimation du poids W_{ijk} pour le composant r_{ijk} .

$$W_{ijk} = \frac{1}{\sigma_{ijk}} \quad (3.7)$$

où σ_{ijk} est l'écart-type.

Après avoir calculé les poids pour chaque composant dans le vecteur r_{ij} , on doit les normaliser par les formules suivantes :

$$W_{ijk} = \frac{W_{ijk}}{W_{Tijk}} \quad (3.8)$$

où $W_{Tijk} = \sum W_{ijk}$

3.2.2. Le retour de pertinence avec notre interface

Puisque l'interface traditionnelle a des limites (elle limite le nombre d'images affichées donc elle ne donne pas à l'utilisateur une vue globale), quelques nouvelles interfaces ont été proposées. L'objectif est de donner à l'utilisateur une interface dynamique et rapide. L'approche de développer des nouvelles interfaces est très récente. Quelques interfaces ont été créées mais

aucune a été prouvée efficace. Deux interfaces dominantes sont l'interface utilisant les cartes auto-organisatrices (en anglais SOM Self-organised Maps)(Deng, 2004) et l'interface utilisant la distance EMD (Earth Mover's Distance) (Rubner, 2003).

Dans (Rubner, 2003), les auteurs ont proposé une interface basée sur la distance EMD. Cette distance vient du problème de transport et elle a été souvent utilisée pour comparer les distributions multi-dimensionnelles. Les auteurs ont appliqué cette distance au système d'indexation et recherche d'images pour faire une interface. Dans cette interface, les images sont affichées en plan N dimensions (2 a été choisi pour N). Dans ce plan, deux images sont proches si la distance entre leurs caractéristiques est petite et inversement. Les auteurs ont testé l'interface avec deux caractéristiques : la couleur (par histogrammes) et la texture (par transformée de Gabor).

Dans (Deng, 2004), les auteurs ont proposé une interface basée sur les cartes auto-organisatrices. Ces cartes sont des modèles du réseau de neurones. Elles sont beaucoup utilisées pour grouper les données. Elles sont aussi appliquées dans le domaine de la recherche d'informations. L'objectif de ces cartes est de grouper et de localiser les vecteurs d'entrée sous forme de nœuds. Deux données entrées qui sont similaires sont localisées en un même nœud ou dans des nœuds proches.

Les résultats de ces deux interfaces sont prometteurs mais elles ont encore des limites. A l'heure actuelle, on ne peut pas afficher les images en fonction de plus d'une caractéristique en même temps. De plus, ces interfaces ne fournissent pas de retour de pertinence. C'est pourquoi nous voulons faire une interface qui donne une façon efficace et dynamique de présenter les images en fonction des distances entre elles et fournir aussi un retour de pertinence.

Nous supposons que les images qui sont pertinentes sont proches de l'image requête et inversement. Donc avec cette interface, l'utilisateur s'intéresse seulement à la région autour de l'image requête (qui contient des images les plus similaires) et il peut cliquer sur cette région pour raffiner la requête.

Pour réaliser nos idées, nous avons développé quelques versions différentes de notre interface. La figure 3-4 donne un exemple de notre ancienne version. Dans cette figure, les images sont affichées en un plan 2D, l'image requête est au coin inférieur gauche et en rouge, les autres ont des positions relatives avec l'image requête en fonction des distances entre elles et l'image requête.



Fig. 3-4 : *Notre première interface graphique en 2D, l'image requête est au coin inférieur gauche, en rouge, les autres images ont des positions relatives à l'image requête en fonction des distances entre elles et l'image requête.*

Le problème de cette interface est que toutes les images sont affichées du même côté de l'image requête et beaucoup d'images sont cachées par d'autres images. C'est pourquoi nous avons essayé encore de l'améliorer. Maintenant, nous avons développé l'interface qui est montrée à la figure 3-5. Dans cette interface, l'image requête est au milieu. Les images résultats sont affichées selon deux axes : l'un utilisant l'intersection d'histogrammes RVB et l'autre utilisant l'intersection d'histogrammes TSV. La position d'une image correspond à la distance entre elle et l'image requête.

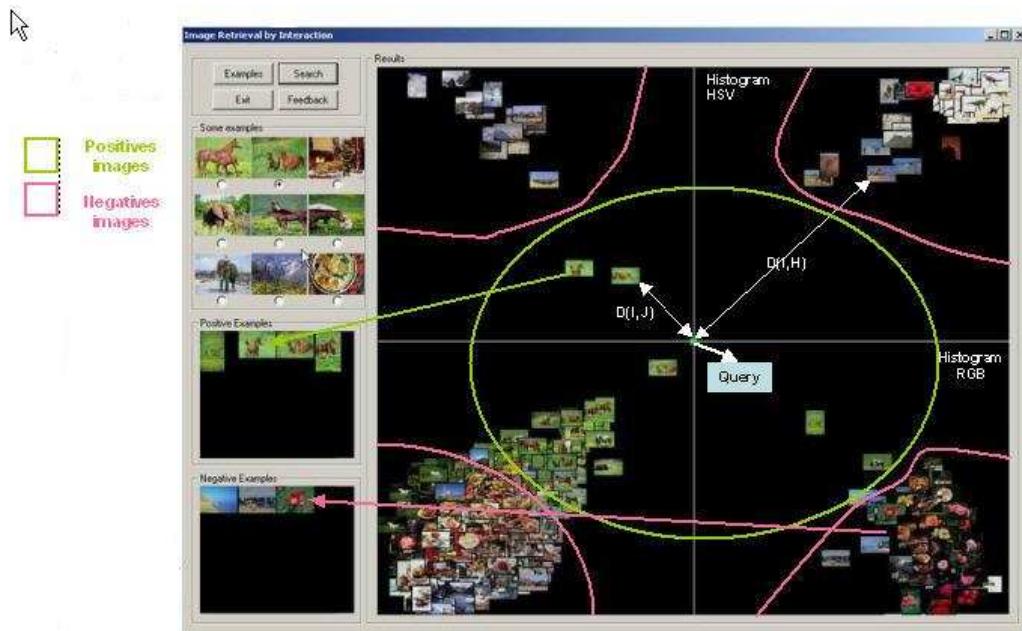


Fig. 3-5 : *Un exemple de notre nouvelle interface, les images sont affichées en un plan avec deux axes : l'intersection d'histogrammes RVB et l'intersection d'histogrammes TSV.*

Puisque toutes les distances sont positives, nous utilisons les entropies des images pour indiquer les signes des distances. Si l'entropie de l'image requête H est plus petite que celle de l'image I dans la base d'images, la distance entre H et I, $d(H,I)$, a un signe négatif et un signe positif inversement.

Avec cette interface, nous avons testé la méthode dans le cas multi-requêtes avec la combinaison par distance minimale pour le retour de pertinence. Cette méthode marche seulement avec les images positives. Nous allons présenter quelques résultats préliminaires dans la section suivante. Nous espérons pouvoir trouver une méthode de retour de pertinence qui va bien avec cette interface.

Conclusions

L'interface traditionnelle donne à l'utilisateur une voie simple, mais lui demande de faire beaucoup d'opérations. Cela peut provoquer l'ennui.

Cependant, l'interface graphique 2D donne à l'utilisateur une façon d'interagir très dynamique. Il semble que cela donne plus informations que via l'interface traditionnelle. Mais il est difficile de réaliser toutes les fonctions. A l'heure actuelle, notre interface graphique 2D n'est pas entièrement terminée. Nous devons la développer pour la rendre plus amicale et plus robuste, et résoudre quelques problèmes posés tels que : l'affichage des images lorsque la base d'images devient très grande; le processus de retour de pertinence pour cette interface avec des images négatives tout en tenant compte de la contrainte d'afficher les résultats 2D seulement. Nous avons fait cette interface en 2D, cela signifie que nous utilisons 2 caractéristiques de l'image. Cependant, nous avons d'autres caractéristiques. Nous regardons deux directions, l'une est de manipuler un plan de dimension supérieure à 2, l'autre est de laisser l'interface 2D en utilisant la technique de l'Analyse en Composantes Principales (en anglais Principal Component Analysis) pour réduire le nombre de dimensions à afficher.

3.2.3. Nos résultats

Dans cette section, tout d'abord nous allons présenter quelques résultats avec l'interface traditionnelle. Nous allons aussi donner les courbes de rappel et précision dans le cas sans retour de pertinence et dans le cas avec le retour de pertinence. Ensuite, nous allons montrer quelques résultats préliminaires avec l'interface 2D.

Avec l'interface traditionnelle, nous avons utilisé :

$$F = \{f_1, f_2\} = \{\text{la couleur, la texture}\}$$

$R = \{r_{11}, r_{12}, r_{13}, r_{14}, r_{21}\} = \{\text{l'intersection d'histogrammes RVB, l'intersection d'histogrammes TSV, les moments RVB, les moments TSV, les matrices de co-occurrences}\}$

Pour le retour de pertinence, nous avons utilisé 3 niveaux : pertinente, pas de commentaire, non-pertinente. Afin de faciliter le calcul, nous supposons que l'importance de chaque composant dans chaque caractéristique est égale. Donc nous faisons seulement la mise à jour inter-poids.

Les figures 3-6 et 3-7 donnent quelques résultats : Dans la figure 3-6, l'utilisateur veut trouver des images qui contiennent un bus, l'image de gauche montre les seize premiers résultats, les images en rouge sont les images non pertinentes, le reste est pertinent. L'image de droite montre les seize premiers résultats quand on fait une fois le retour de pertinence. Le nombre d'images non-pertinentes diminue et leurs rangs sont bien plus grands que ceux des images pertinentes. Dans la figure 3-7, l'utilisateur veut trouver des images qui contiennent un (des) cheval (chevaux). Avec le retour de pertinence, les résultats sont très bons. Parmi les seize images résultats, il y a seulement deux images non-pertinentes qui contiennent aussi des animaux (des éléphants).

Pour faire les courbes de rappel et précision, nous proposons une façon pour faire automatiquement des retours de pertinence. Nous indiquons les images dans les seize premières images qui appartiennent à la même classe comme images pertinentes et inversement. Nous avons testé le retour de pertinence avec la base de Wang. La figure 3-8 donne trois courbes de rappel et précision dans le cas sans retour de pertinence, avec un seul retour de pertinence et avec deux retours de pertinence. Ces courbes sont calculées pour toute la base de Wang. Cependant la figure 3-9 donne ces courbes pour la classe 3 dans la base d'images de Wang



Fig. 3-6 : *L'utilisateur veut trouver des images qui contiennent un bus, l'image de gauche montre les seize premiers résultats, les images en rouge sont les images non-pertinentes, le reste est pertinent. L'image de droite montre les seize premiers résultats quand on fait une fois de retour de pertinence*

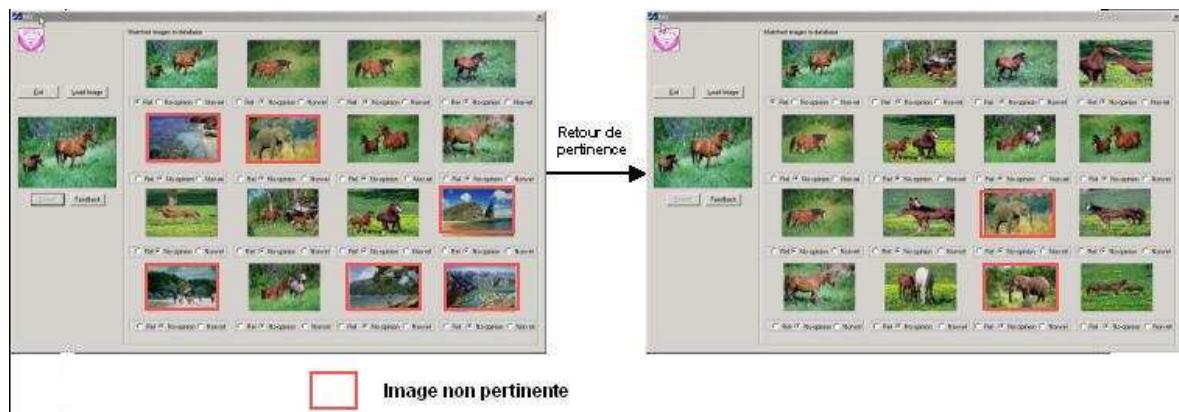


Fig. 3-7 : L'utilisateur veut trouver des images qui contiennent un (des) cheval (chevaux). Avec le retour de pertinence, les résultats sont très bons. Parmi les seize images résultats, il y a seulement deux images non-pertinentes qui contiennent aussi des animaux (des éléphants).

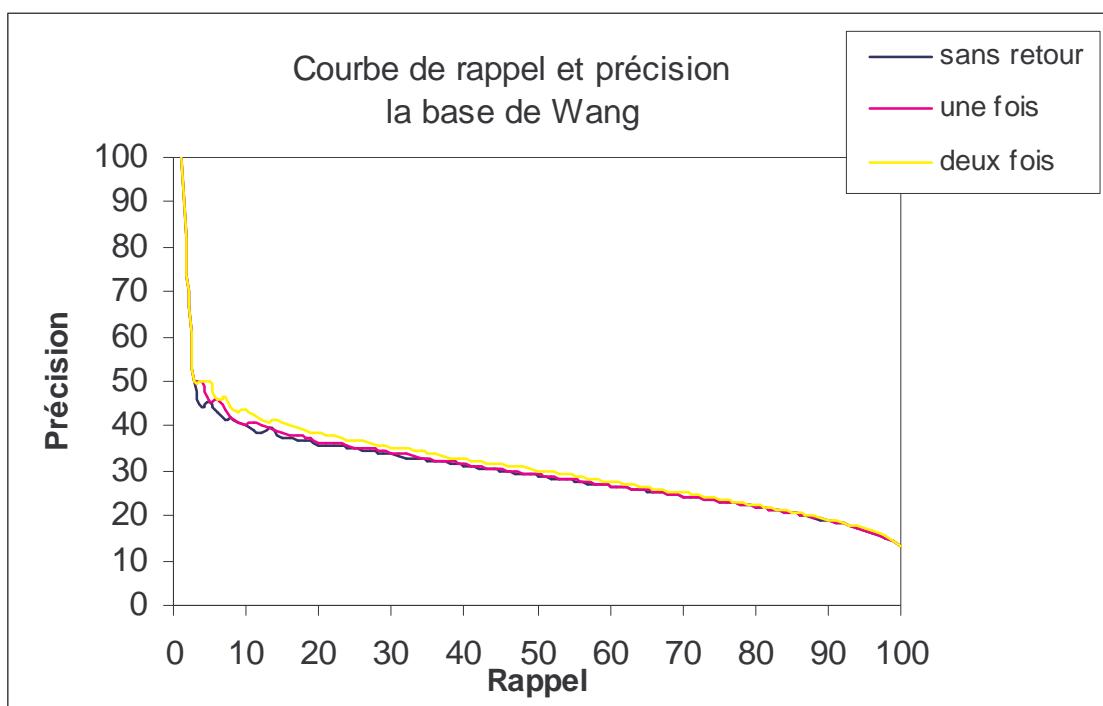


Fig. 3-8 : *Les courbes de rappel et précision de la base de Wang sans le retour de pertinence, avec un seul retour de pertinence et avec deux retours de pertinence*

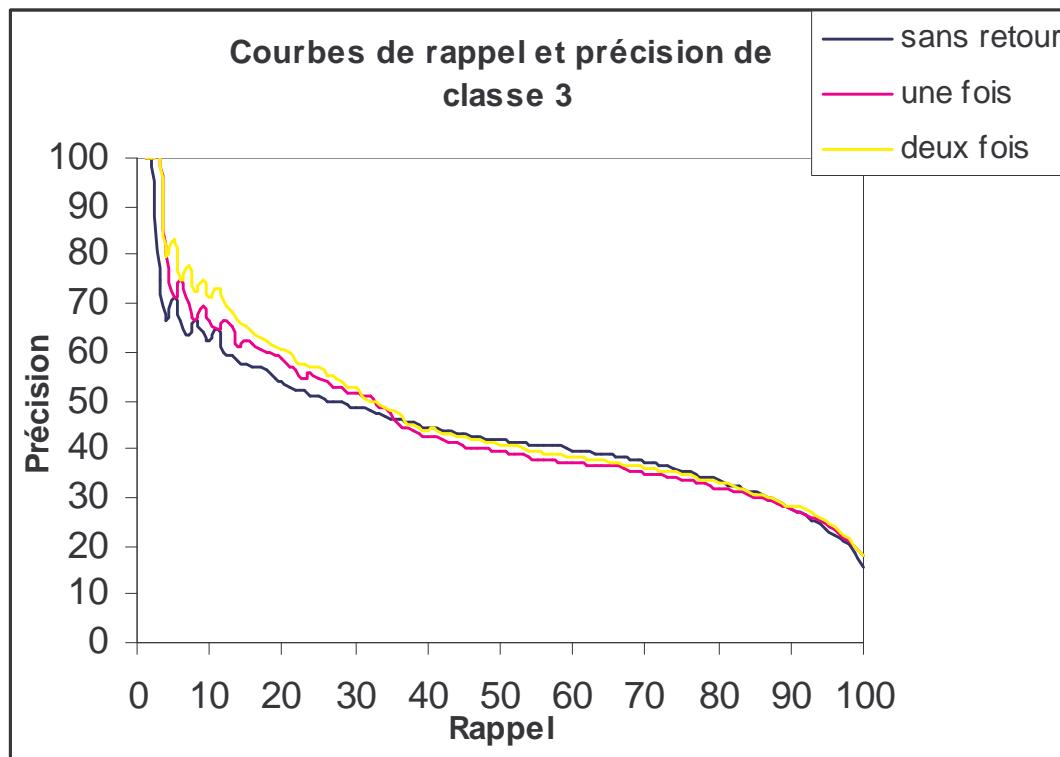


Fig. 3-9 : *Les courbes de rappel et précision de la classe 3 de la base de Wang sans retour de pertinence, avec un seul retour de pertinence et avec deux retours de pertinence*

A partir de ces résultats, nous pouvons voir que les résultats sont prometteurs mais ils ne sont pas encore parfaits. Il semble que plus nous faisons de retours de pertinence, plus les rappels et précisions augmentent. En comparant les courbes de rappel et précision entre deux étapes continues, les rappels et précisions augmentent mais ils n'augmentent pas beaucoup.

Aussi nous constatons que les résultats sont bons quand le nombre d'images retrouvées est petit, mais les résultats deviennent moins bons quand le nombre d'images retrouvées augmente. On peut expliquer cette observation par le fait que nous utilisons seulement les 16 premières images pour le retour de pertinence. Nous croyons que les résultats seront meilleurs si nous utilisons plus de 16 images pour le retour de pertinence. De plus, ces résultats seraient

encore améliorés en utilisant la mise à jour inter-poids, avec plus de caractéristiques et avec plus de niveaux de retour de pertinence.

La figure 3-10 montre notre résultat avec l'interface graphique 2D. L'image à gauche montre les résultats de recherche d'images sans retour de pertinence où la région en rouge indique l'image requête (une image qui contient des chevaux). A partir de ces résultats, nous choisissons 4 images positives et refaisons la recherche. L'image à droite montre les résultats de recherche d'images avec retour de pertinence où la région en bleue indique les images positives que nous avons choisies.

Nous avons constaté que les résultats sont très bons avec le retour de pertinence. Après un seul retour de pertinence, presque toutes les images pertinentes dans la base d'images sont affichées autour de l'image requête.

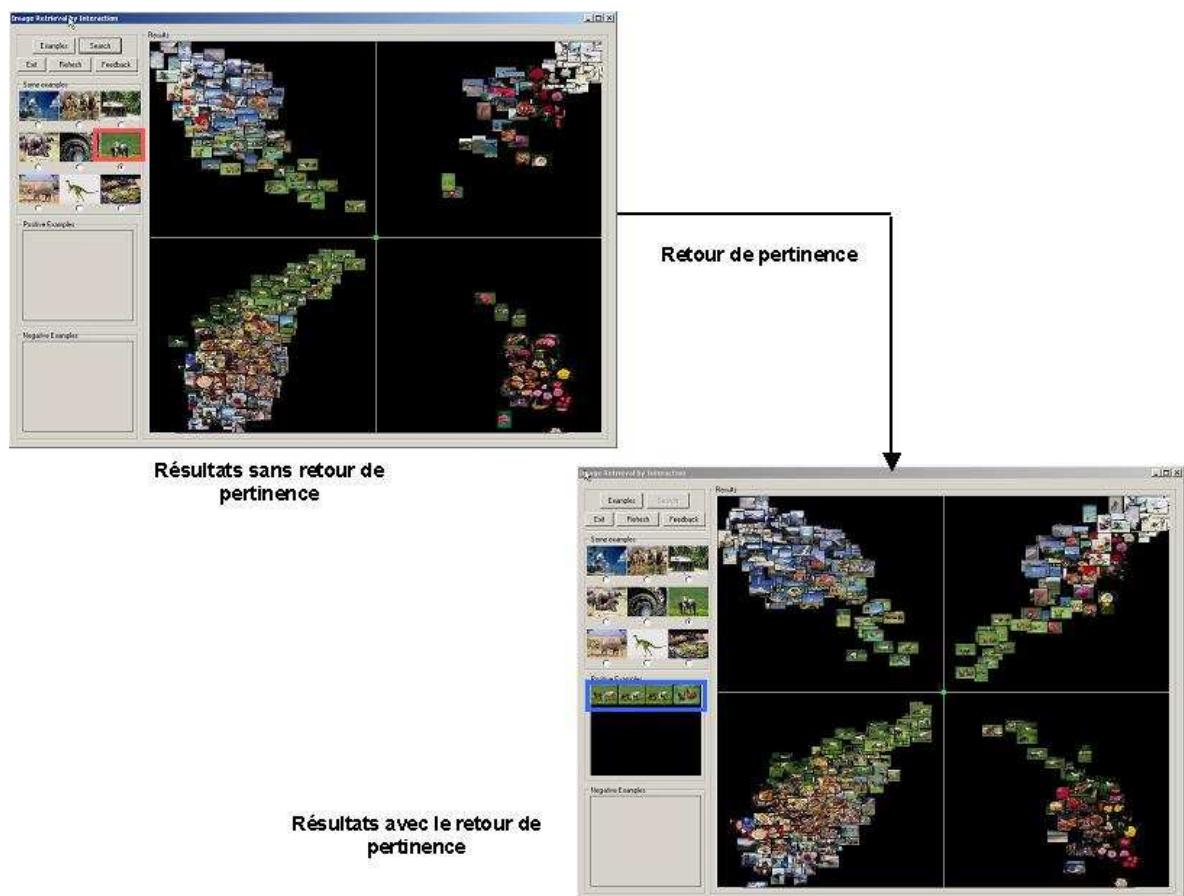


Fig. 3-10 : *L'image à gauche montre les résultats de recherche d'images sans retour de pertinence où la région en rouge indique l'image requête. L'image à droite montre les résultats de recherche d'images avec le retour de pertinence où la région en bleue indique les images positives que nous avons choisies.*

A partir de ces résultats prometteurs, nous allons encore poursuivre cette approche. Nous croyons que les résultats avec l'interface traditionnelle peuvent être encore améliorés et nous allons terminer complètement les fonctions pour l'interface 2D.

Chapitre 4: Conclusion et perspectives

4.1. Conclusions

Bien qu'ils existent depuis presque dix ans, les moteurs de recherche d'images arrivent peu à peu à maturité. Ils sont capables de comparer des images entre elles. Leurs résultats, qui devront être encore améliorés, sont déjà très satisfaisants.

Nous avons atteint presque tous ces buts que nous nous étions fixés dans la section 1.1.2. Tout d'abord, nous avons étudié l'état de l'art des systèmes d'indexation et de recherche d'images par le contenu symbolique ainsi que par le contenu sémantique. Ensuite, à MICA, nous avons développé un système d'indexation et recherche par le contenu. C'est le premier système de recherche d'images développé à MICA. Ce système fonctionne avec des caractéristiques symboliques qui sont extraites d'une d'images telles que l'intersection d'histogrammes RVB, l'intersection d'histogrammes TSV, les moments RVB, les moments TSV et la texture. Nous les avons aussi combinées pour améliorer les résultats. Nous avons aussi développé une interface qui permet à l'utilisateur de faire facilement une requête et de prendre les résultats. Notre système a été utilisé avec quelques bases d'images différentes : la base de Wang, la base de Pollen, la base de Coil, la base de CUReT et la base de FeiFei. Pour évaluer notre système, nous avons calculé les courbes de rappel et précision pour chaque base d'images avec les différents choix de caractéristiques. Nos résultats ne sont pas toujours excellents avec toutes les bases d'images mais ils sont cependant très prometteurs. Ils nous permettent de mieux comprendre le système d'indexation et recherche d'images et à partir de ces résultats, nous pouvons trouver plusieurs façons pour améliorer notre système.

Nous trouvons qu'un système d'indexation et recherche par le contenu ne satisfait pas toujours l'utilisateur. L'addition de valeurs sémantique dans le système d'indexation et recherche d'images est nécessaire. Il existe deux approches pour ajouter de la sémantique dans un système de recherche d'images (1) l'une recherche des méthodes pour comprendre l'objectif de l'utilisateur, le sens de sa requête ; (2) l'autre recherche des moyens pour « connecter » (ou lier) la connaissance sémantique humaine et l'apparence de l'image. Pour l'instant, nous nous concentrons seulement sur la première approche. Un système qui fournit le retour de pertinence avec l'interface traditionnelle ainsi qu'avec une interface graphique en 2D a été proposé.

4.2. Perspectives

Cette section se consacre à nos perspectives. Tout d'abord nous voulons améliorer notre système d'indexation et recherche d'images. Ensuite, nous voulons nous concentrer sur la partie sémantique. Avec le retour de pertinence, nous avons fait quelques expériences, mais il nous reste à inclure l'ontologie dans ce travail. Nous voulons combiner la technique de retour de pertinence avec l'ontologie. Nous allons faire des applications pour le Vietnam. Pour cela, nous avons obtenu un projet AUF PCSI (IRIS) pour développer les applications de recherche d'images pour le Vietnam. Nous sommes en train de construire une nouvelle base d'images sur le Vietnam et allons essayer d'utiliser nos résultats avec d'autres travaux pour caractériser cette base d'images. Nous allons détailler nos perspectives.

4.2.1. Améliorer notre système d'indexation et recherche par le contenu

- Notre système actuel de recherche d'images utilise des caractéristiques globales (au niveau image). Nous sommes en train de faire un système

qui fonctionne avec des caractéristiques globales (au niveau image) ainsi qu'avec des caractéristique locales (au niveau région).

- Nous allons tester avec plusieurs bases d'images ainsi qu'avec plusieurs poids pour chaque caractéristique. Les résultats de ces tests nous permettront d'identifier, pour chaque base d'images, les caractéristiques les plus influentes et, nous l'espérons, de calculer automatiquement les poids de chaque caractéristique en fonction du type d'images.
- Avec plusieurs images requêtes, nous voulons utiliser plusieurs bases d'images et calculer les courbes de rappel et précision pour pouvoir dire quelle méthode de combinaison est la meilleure pour chaque base d'images.

4.2.2. La base d'images MICA

Pourquoi avons-nous besoin de construire une nouvelle base d'images ?

Il y a deux raisons principales. Premièrement, nous voulons avoir une base d'images pour des conditions réelles. La plupart des bases d'images que nous utilisons pour tester notre système ont été acquises avec des conditions contrôlées. Deuxièmement, nous voulons avoir une base d'images où chaque image appartient à plusieurs classes. Dans les bases d'images utilisées, chaque image appartient à une seule classe. Habituellement, quand nous prenons une image, cette image contient plus d'un objet même si le focus est fait pour un objet spécifique. Le processus d'annotation et le processus d'évaluation devraient suivre cette idée. Nous voulons aussi fournir notre base d'images à d'autres équipes de recherche en images afin de comparer les systèmes de recherche d'images existants.

Notre base d'images à l'heure actuelle contient plus 8800 images représentant des scènes et des sujets différents tels que les fleurs, les animaux, les paysages, les gens. Ces images sont des images réelles, qui ont été acquises dans les plusieurs conditions différentes. Notre but est fournir le type d'images qu'une personne peut prendre dans sa vie quotidienne. Dans ces images, il y a des objets différents, des scènes différentes, des pays différents. Cette base devrait permettre d'évaluer les systèmes dans les conditions réelles. La figure 4-1 donne quelques exemples de notre base d'images.

Afin d'utiliser ces images, elles doivent être annotées et classifiées en classes. Il y a un problème quand nous classifions ces images parce que chaque image appartient à plus d'une classe. Cette propriété est très importante et opposée à d'autres bases d'images. Dans la plupart des cas, une image contient plus d'un objet. Cette image peut appartenir à quelques classes dépendant si l'utilisateur veut un animal spécifique ou un genre de scène particulier ou le pays où cette image a été prise. Nous avons écrit un programme pour annoter les images. Des mots clés ont été proposés en regardant toutes les images dans notre base d'images. La figure 4-2 représente l'interface du système d'annotation d'images. Les mots clés sont affichés sous forme d'arbre pour plus de facilité.



Fig. 4-1 : Quelques exemples dans notre base d'images



Fig. 4-2 : Interface de notre programme pour annoter une image, un arbre de mots clés (à gauche), image à annoter (à droite)

La figure 4-3 représente quelques images avec leurs mots clés choisis.



**Outside day,
moutain, sky, clouds,
water, Vietnam.**



**Outside day, animal,
deer, grass**



**Outside day, bush,
people, minorities,
Vietnam.**



**Outside day,
building, people,
road, sky, Hong
Kong (China).**



**Underwater, fish,
water.**



**Outside day, tree,
people, moto,
car, building,
Vietnam**

Fig. 4-3 : Quelques images annotées avec ses mots clés choisis

Le processus d'annotation est réalisé par les membres de MICA. Au jour d'aujourd'hui, ce processus n'est pas terminé, mais devrait l'être bientôt.

4.2.3. Caractérisation des bases d'images pour l'évaluation

La dernière décennie a vu une explosion du nombre d'articles publiés dans le domaine de l'indexation et de la recherche d'images par le contenu. Mais comme dans beaucoup de domaines, le problème de la validation et de la comparaison des résultats publiés par les différentes équipes de recherche demeure crucial. Dans le domaine de la recherche d'images, chaque équipe utilise souvent sa propre base d'images, soit par intérêt particulier ou tout simplement parce que les travaux ont une finalité vers une application spécifique, donc une base d'images spécifique. Les résultats d'une approche de manipulation (classification, recherche d'informations...) d'une base

d'images dépendent forcément de ses propriétés. Il est évident que si une base est « facile » alors on obtiendra de bons résultats et inversement. Cela cause des problèmes de comparaison des différentes approches proposées. Il existe une solution pour ce problème. Dans le domaine de recherche de textes par exemple, on a construit une base de textes commune (TREC-Text Retrieval Conference) et on l'utilise pour évaluer les systèmes de recherche de textes. Dans le domaine de recherche d'images, on a essayé de faire la même chose. C'est pourquoi ImageCLEF (Cross Language Evaluation Forum) a été créé. Mais la base d'images dans ImageCLEF n'est pas suffisante pour évaluer tous les systèmes de recherche d'images.

Plutôt que de contraindre les chercheurs à utiliser les mêmes bases d'images, est-ce qu'il serait possible de leur donner des outils permettant de comparer les différentes bases d'images entre elles ? C'est ce que nous tentons de faire en utilisant un outil classique dans le domaine de la classification que sont les arbres de décision. Ce travail sur la comparaison de bases d'images a été fait en collaboration avec M. Dang Thanh Ha, étudiant en co-direction de thèse entre le laboratoire LIP6 (Paris, France) et l'IFI (Hanoi, Vietnam)(Boucher, 2005). Dans cette collaboration, nous suggérons une méthode pour caractériser les bases d'images et ainsi permettre la comparaison des travaux de recherche. Cela permet d'estimer la difficulté des bases d'images pour donner une base de comparaison des méthodes de recherche d'images par le contenu.

Dans (Boucher, 2005) nous avons proposé de calculer un index caractéristique pour une base d'images comme suit :

Index = f(Base d'images, Attributs image, Méthode de composition)

Pour une base d'images, nous devons calculer des attributs sur chaque image et utiliser une méthode de composition de ces attributs pour produire

un index valable. Ensuite, si nous utilisons la même procédure pour une nouvelle base d'images, nous espérons obtenir un nouvel index qui permettra de comparer les deux bases d'images, et aussi ensuite les résultats d'algorithmes travaillant sur ces deux bases d'images. Pour cela, les attributs utilisés et la méthode de composition doivent être fixés et ne pas changer, car autrement la comparaison des index devient impossible. Comme nous nous intéressons au domaine de la recherche d'images similaires, nous avons choisi les attributs les plus fréquents pour ce domaine que sont la couleur et la texture. Nous avons ensuite choisi comme méthode de composition de l'index le calcul par arbres de décision, qui constitue une méthode simple, reproductible, indépendante, mais présentant des similarités avec le domaine ciblé.

Nous avons expérimenté la méthode à base d'arbres de décision sur les quatre bases d'images (la base de Wang, la base de Coil100, la base de CUReT et la base de Pollens) de la section 2.2.2 sur la base d'images utilisées et calculé quatre index différents : le taux de bonne classification, l'entropie de la base de test, le gain et le taux de gains d'information.

En comparant les résultats obtenus avec nos résultats de recherche d'images, nous trouvons que ces index peuvent identifier la difficulté de la base d'images. Parmi ces index, le taux de bonne classification et le taux de gain d'information sont des index adéquats pour caractériser les différentes bases d'images. Plus le taux est élevé pour une base d'images et plus cette base est considérée facile (uniquement pour le problème de la recherche d'images) et inversement.

Nous avons présenté dans cette section une nouvelle approche pour aider l'évaluation et la comparaison des systèmes d'indexation et recherche d'images par le contenu, en utilisant une technique de classification à base

d'arbres de décision pour classifier les bases d'images existantes et les ordonner. En publiant les résultats de leurs travaux sur une base d'images particulière, les auteurs pourraient ainsi donner en même temps les différents index associés à leur base afin de permettre une meilleure comparaison des résultats. Cette approche est selon nous possible dans le domaine de la recherche d'images par le contenu parce que les travaux dans ce domaine se font très souvent sur des applications générales et des bases d'images non-spécifiques, ce qui justifie le choix d'attributs d'images généraux comme la couleur et la texture. Bien que certains problèmes restent à résoudre, cette approche permet néanmoins de poser le problème de la validation sous un nouvel angle.

4.2.4. Le projet IRIS

A partir du besoin d'un système d'indexation et de recherche d'images au Vietnam, en octobre 2004, le Centre MICA a soumis une demande projet PCSI à l'AUF : IRIS (Indexation et Reconnaissance d'Images par la Sémantique). En janvier 2005, l'AUF a accepté ce projet. Ce projet a trois objectifs principaux comme suit :

- Développer une compétence et une recherche scientifique de qualité au Vietnam par des collaborations avec des laboratoires de recherche reconnus internationalement de France et de Belgique.
- Etudier et améliorer les connaissances existantes dans le domaine de l'indexation et de la recherche d'images, en particulier en traitant l'aspect sémantique de haut niveau lié aux images.
- Faire bénéficier le Vietnam de retombées à court terme de ce projet, en concevant une plate-forme logicielle de gestion d'une base d'images en

collaboration avec des acteurs choisis dans deux secteurs d'activité économique du Vietnam.

La recherche d'images est un sujet de recherche très actif dans la communauté internationale depuis plus d'une dizaine d'années. Les résultats obtenus jusqu'à présent sont très prometteurs. Mais un problème n'est toujours pas résolu dans cette recherche et concerne l'identification d'informations à caractère sémantique dans l'image.

En regardant les avancées récentes en matière de recherche, trois voies complémentaires sont possibles pour ajouter de la sémantique dans un processus d'indexation ou de reconnaissance des formes : le langage structuré de définition de concepts, l'apprentissage automatique et l'interaction homme-machine. Chacune de ces trois voies permet d'apporter une partie de la sémantique nécessaire pour un processus de vision par ordinateur, et ce d'une façon différente, mais incomplète

Dans le projet IRIS, le but est de coupler ces trois voies complémentaires en un seul système d'indexation et de recherche d'images. A notre connaissance, ceci n'a jamais été fait auparavant. Il existe des travaux utilisant une ou deux techniques d'apport de sémantique, avec des résultats intéressants mais incomplets. Cependant, il n'existe pas de travaux unissant ces trois voies pour compléter le manque de sémantique pour l'indexation et la recherche d'images. Il s'agirait d'une innovation pour la recherche en ce domaine.

Pour ce faire, l'expérience scientifique de chaque partenaire est indispensable :

- **L'équipe Orion (INRIA)** pour l'expertise reconnue dans le domaine des ontologies et des systèmes à base de connaissances ;

- **L’Institut Montefiore (Université de Liège)** pour l’expérience dans le domaine de l’apprentissage automatique en vision par ordinateur ;
- **Le centre de recherche MICA (IPH)** pour le travail et la recherche en cours dans le domaine de l’interaction homme-machine.

Le but de ce projet est non seulement de faire la recherche de haut niveau mais aussi de développer les applications pour le Vietnam. Il s’agit de deux secteurs d’activité importants pour le pays, soit le journalisme et le tourisme, et ayant en commun le besoin grandissant pour des outils de gestion spécialisés pour les grandes bases d’images numériques.

Ces deux applications différentes permettront de

- identifier les besoins spécifiques pour le Vietnam liés à l’indexation et à la recherche d’images, et ce dans des contextes différents ;
- tester et valider le système développé dans le cadre de ce projet ;
- diffuser des retombées mesurables et à très court terme des recherches financées dans ce projet IRIS.

Autres perspectives

Nous venons de présenter nos perspectives à court terme. La recherche d’images est une sous-partie de la recherche d’informations. A MICA, il existe d’autres travaux de recherche : la recherche de vidéos d’un étudiant Master Pham Quang Hai, la recherche audio de l’équipe de la parole. Ces recherches peuvent se regrouper dans un même système de recherche multi-modalités. A long terme, nous voulons les combiner pour développer un système unique de recherche d’informations.

Annexe1 – Transformation d'une image RVB en une image TSV

Pour transformer une image, on va transformer tous les pixels dans cette image. Avec un pixel qui a 3 valeurs (r, v, b), après avoir transformé, il a 3 valeurs (t, s, v). Les formules utilisées sont les suivantes :

$$v = r + v + b / 3$$

$$s = 1 - \text{MIN}(r, v, b) / (r + v + b)$$

$$\text{delta} = Ar \cos \frac{(0.5 * ((r - v) + (r - b)))}{\sqrt{(r - v)^2 + (r - b) * (v - b)}}$$

si ($v \geq b$) $t = \text{delta}$

si non $t = (2 * \pi - \text{delta})$

Annexe2- Normaliser la distance – Gaussien

$d=\{d_i\}$: ensemble des distances.

Nous calculons μ (l'espérance) et σ (l'écart type) de d .

$d_i = (d_i - \mu)/3\sigma$, 99% d_i est entre [-1, 1]

$d_i = (d_i + 1)/2$, d_i est entre [0,1]

Publications

1. *Le Thi Lan et Alain Boucher, An interactive image retrieval system: from symbolic to semantic*, International Conference on Electronics, Information, and Communications (ICEIC), 16-18 août 2004, Hanoi (Vietnam).
2. *Alain Boucher et Le Thi Lan, Comment extraire la sémantique d'une image ?* International Conference on Sciences of Electronic, Technologies of Information and Telecommunications (SETIT), mars 2005, Susa (Tunisie).
3. *Alain Boucher, Thanh-Ha Dang, Le Thi Lan, Classification vs recherche d'information : vers une caractérisation des bases d'images*, 12èmes Rencontres de la Société Francophone de Classification (SFC), mai/juin 2005, Montréal (Canada).

Bibliographie

Les références anglaises

1. Blobworld, <http://elib.cd.berkeley.edu/photos/blobworld/flowers.html>
2. Bonton P., Boucher A., Thonnat M., Tomczak R., Hidalgo P.J., Belmonte J., Galan C. (2002), « Colour image in 2D and 3D microscopy for the automation of pollen rate measurement », *Image Analysis and Stereology*, vol. 21, no 1, march 2002, pp. 25-30.
3. Chandrasekaran B., Josephson J.R., Benjamins V.R., “What are Ontologies and Why Do We Need Them ?” *IEEE Intelligent Systems*, 1999.
4. CUReT, <http://www1.cs.columbia.edu/CAVE/curet/>
5. Deng D., Zhang J., Purvis M. (2004), “Visualisation and Comparison of Image Collections based on Self-organised Maps”, *Proc. of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, Vol. 32, Dunedin, New Zealand, pp. 97-102, 2004.
6. Deselaers T. (2003), *Features for Image Retrieval*, Diploma Thesis, RWTH Aachen University. Aachen, Germany, December 2003.
7. Forsyth D.A., Ponce J.(2003), *Computer Vision – A Modern Approach* Prentice-Hall.
8. Gandon F.(2002), *Ontology engineering : A survey and a return on experience*, Tech. Rep. 4396, INRIA.
9. Gong Y., Proietti G., Faloutsos C.(1998), “Image Indexing and Retrieval Based on Human Perceptual Color Clustering”. *Proc. of International Conference on Computer Vision and Pattern Recognition(CVPR)*, June 1998

10. Gruber T.R. (1993), “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”, *In Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer The Netherlands, 1993, N.Guarino and R.Poli, Eds., Kluwer Academic Publishers.
11. Haralick R.M., Shanmugam K., Dinstein I. (1973), “Textural Features for Image Classification”, *IEEE Trans. On Systems, Man, and Cybernetics*, Vol. SMC-3, No. 6, November 1973, pp. 610-621.
12. Howarth P., Ruger S., “Evaluation of Texture Features for Content-based Image Retrieval”, *CIVR 04*, p. 326-334.
13. Huang J., Kumar S., Mitra M., Zhu W.J., Zabih R. (1997), “Image Indexing Using Color Correlograms”. *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, San Juan (Puerto Rico), pp. 762-768, 1997.
14. Hyvonen E., Styrman A., Saarela S. (2003), “Ontology-Based Image Retrieval”,
<http://www.cs.helsinki.fi/group/seco/presentations/www2003/p199-hyvonen.html>
15. Jiang, S., Huang, T., & Gao, W. (2004). “An Ontology based Approach to Retrieve Digitized Art Images”. *Paper presented at the Web intelligence IEEE/WIC/ACM WI 04*, Beijing, China.
16. Iqbal Q., Aggarwal J.K. (2002), “Combining structure, color and texture for image retrieval: a performance evaluation”. *Proc. of International Conference on Pattern Recognition (ICPR)*, Quebec (Canada), 2002.
17. Iqbal Q., Aggarwal J.K. (2003), “Feature Intergration, Multi-image Queries and Relevance Feedback in Image Retrieval”, *Conf. Visual*

- Information Systems (VISUAL 2003)*, pp. 467-474, Miami, Florida, Sep 24-26, 2003.
18. Jing F., Zhang B., Lin F., Ma W.Y., Zhang H.J.(2001), “A novel region-based Image Retrieval Method Using Relevance Feedback”, mir 2001.
19. Maillot N., Thonnat M., Boucher A. (2004), “Towards ontology-based cognitive vision”, *Machine Vision and Application*, 2004.
20. Maillot N., Thonnat M.(2005), “A Weakly Supervised Approach for Semantic Image Indexing and Retrieval”, *International Conference on Image and Video Retrieval (CIVR)*, July 20-22, 2005, Singapore.
21. Malki J., Boujemaa N., Nastar C., Winter A.(1999), “Region Queries without Segmentation for Image Retrieval by Content”, 3rd Int. Conference on Visual Information Systems(Visual’99).
22. Medioni G.. Sing Bing Kang S.B.(2005), “Emerging Topics in Computer Vision”. *IMSC Press Multimedia Series*, Prentice-Hall PTR, chapter 8, 2005.
23. Mezaris V., Kompatiariis I., Strintzis M.G.(2003), “An Ontology Aprroach to Object-Based Image Retrieval”, ICIP 2003.
24. Muller H., Müler V., McG.Squire D. (1999), “Performance Evaluation in Content-Based Image Retrieval : Overview and Proposals”, *Technical report vision*, N. 99.05, December 1, 1999, Universite de Geneve.
25. Nakazato M., Manola L., Huang T.S.(2002), “ImageGrouper: Search, Annotate and Organize Images by Groups,” *In Proc. of 5th Intl. Conf. On Visual Information Systems (VIS’02)*, 2002.

26. Nakazato M., Huang T.S.(2002), “Extending image retrieval with group-oriented interface”, *in IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, pp. 201-204, 2002.
27. Noy N.F., McGuinness D.(2001), “Ontology Development 101 : A Guide to Creating Your First Ontology”, 2001.
<http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguiness.html>
28. Park D., Park J., Han J.H.(1999), “Image Indexing using Color Histogram in the CIELUV Color Space”. *Proc. of the 5th Japan-Korea Joint Workshop on Computer Vision*, pp. 126-132, January 1999.
29. Pass G., Zabih R.(1996), “Histogram refinement for content-based image retrieval”. *Proc. of the Third IEEE Workshop on Applications of Computer Vision*, Sarasota (FL, USA), December 1996.
30. QBIC <http://www.heritagemuseum.org/cgi-bin/db2www/qbicLayout.mac/qbic?selLang=English>
31. Rubner Y., Tomasi C., Guibas L.J.(2003), “A Metric for Distributions with Applications to Image Databases”, Proc of ICCV03.
32. Rui Y., Huang T.S., Ortega M., Mehrotra S. (1998), “Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval”, *IEEE Transaction On Circuits and Video Technology*, 1998.
33. Sciascio E.D.(1997), “Similarity Evaluation In Image Retrieval Using Simple Features”. *Proc. of Storage and Retrieval for Image and Video Databases V*, February 8-14, San Jose (CA, USA), vol. 3022, pp. 467-477, 1997.
34. Smeulders A.W.M., Worring M., Santini S., Gupta A., Jain R.(2000), “Content-Based Image Retrieval at the End of the Early Years”. *IEEE*

- Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349-1380, December 2000
35. Smith J.R.(1998), “Image Retrieval Evaluation”, *IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL-98)*, June, 98.
36. Stricker M., Orengo M.(1995), “Similarity of Color Images”, *In Proceedings of SPIE, Vol. 2420(Storage and Retrieval of Image and Video Databases III)*, SPIE Press, Feb 1995.
37. Stricker M., Dimai A.(1996), “Color indexing with weak spatial constraints”. *SPIE Proceedings*, vol. 2670, pp. 29 - 40, 1996.
38. Sun Y., Ozawa S.(2004), “A novel image retrieval algorithm by using salient points in wavelet domain”. *Proc. of Asian Conference on Computer Vision (ACCV)*, 2004
39. Swain M.J., Ballard D.H.(1991), “Color indexing”. *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11-22, 1991.
40. Wang J.Z., Wiederhold G., Firschein O., Wei S.X. (1997), “Content-based image indexing and searching using Daubechies wavelets”, p. 311-328, Digit Libr 1997.
41. Yates R.B., Neto B.R. (1999), “*Modern Information Retrieval*”, ACM Press, 1999.
42. Zhou X.S., Petrovic N., Huang T.S.(2001), “Comparing Discriminating Transformations and SVM for Learning during Multimedia Retrieval”, *In Proceedings of ACM Multimedia'01*, 2001.
43. Zhu L., Zhang A.(2000), “Supporting Multi-example image queries in Image Databases”, *In International conference on Multimedia and Expo*, vol. 1, pp. 697-700, July 30-August 2, 2000.

Les références françaises

1. Boucher A., Ha. T.D, Lan L.T (2005), «Classification vs recherche d'information : vers une caractérisation des bases d'images », *12èmes Rencontres de la Société Francophone de Classification (SFC)*, mai/juin 2005, Montréal (Canada).
2. Lan N. T. H. (1999), « Traitement d'images », Bureau Asie du Sud-Est, 21 Le Thanh Tong – Hanoi, 1999/2000.
3. Vandenbroucke N.(2000), « Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés - Application à l'analyse d'images de football ». Thèse de doctorat, Université des Sciences et Technologies de Lille, 2000.