

MÉMOIRE DE MASTER DE RECHERCHE  
« ARCHITECTURES LOGICIELLES DISTRIBUÉES »

# Personnalisation de l'accès à l'information multimédia

*Efficacité et efficience ?*

Mme Karima AMROUCHE IHADDADENE

4 septembre 2006

encadré par José MARTINEZ

— Atlas-GRIM —

LABORATOIRE D'INFORMATIQUE DE NANTES-ATLANTIQUE



# Personnalisation de l'accès à l'information multimédia

## Efficacité et efficience ?

Mme Karima AMROUCHE IHADDADENE

### Résumé

L'usage de la personnalisation dans ce travail est d'améliorer la recherche d'information en termes d'efficience et d'efficacité. En effet, les systèmes actuels de recherche d'images restent toujours incapables de fournir à l'utilisateur les résultats souhaités, sans même mentionner la rapidité des réponses.

Tout d'abord, une difficulté majeure est le fossé entre les propriétés de bas niveau, utilisées pour représenter les images, et la sémantique de haut niveau de la perception humaine incluant les différentes interprétations que chacun peut donner à la même image.

En recherche d'information textuelle, la personnalisation est un domaine d'étude bien établi ; plusieurs travaux ont des apports en terme d'efficacité et d'efficience. Mais le texte est une abstraction construite par l'humain : il se compose d'un ensemble de mots, chacun souvent un sens précis donc « facile » à indexer et retrouver.

Pour l'image, sa description objective pose déjà un grand problème et, par la suite, son interprétation dépend fortement de chaque personne. C'est pour cette raison que l'intégration de l'utilisateur humain dans un système de recherche d'image est à la fois difficile et indispensable.

Par ailleurs, les temps de recherche restent extrêmement élevés par rapport à ceux auxquels les utilisateurs sont habitués dans les systèmes de gestion de bases de données ou les systèmes de recherche d'information textuelle.

Ce qui est présenté ici est l'utilisation des préférences des utilisateurs en observant leur activité dans un système de recherche d'images par le contenu et de les rassembler pour construire des profils qui vont servir à restreindre les futures recherches aux parties les plus pertinentes d'une grande base d'images.

Catégories et descripteurs de sujets : H.2.8 [**Database Applications**]: Image databases; H.3.3 [**Information Search and Retrieval**]: Clustering, Information filtering, Relevance feedback; H.3.7 [**Digital Libraries**]: User issues; I.5.3 [**Clustering**]: Algorithms, Similarity measures; I.4.10 [**Image Representation**]: Statistical, Multidimensional

Termes généraux : Algorithms, performance, experimentation, human factors, verification.

Mots-clés additionnels et phrases : Personnalisation, recherche d'images par le contenu, classification, rétro-action, apprentissage.



## Remerciements

Ce rapport est le fruit de six mois de stage au sein du LINA, le laboratoire d'informatique de Nantes-Atlantique.

Je tiens à remercier tous les membres de l'équipe Atlas-GRIM pour m'avoir gentiment accueilli et intégré. Mais je tiens plus particulièrement à remercier :

- M. Martinez José pour m'avoir proposé ce sujet de stage intéressant et d'avant garde mais aussi pour m'avoir encadré et aidé durant ces mois dans mon travail ;
- M. Mouaddib Nourredine, Directeur de l'équipe Atlas-GRIM, pour m'avoir donné la chance de réaliser mon stage dans ce laboratoire.

# Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Présentation du sujet</b>               | <b>8</b>  |
| <b>2</b> | <b>Introduction</b>                        | <b>11</b> |
| 2.1      | Objectifs poursuivis                       | 12        |
| 2.2      | Problématique                              | 13        |
| 2.2.1    | Modélisation et initialisation du profil   | 13        |
| 2.2.2    | Évolution et utilisation du profil         | 14        |
| <b>3</b> | <b>État de l'art</b>                       | <b>15</b> |
| 3.1      | Recherche d'information textuelle          | 15        |
| 3.1.1    | Étude du profil d'un utilisateur           | 16        |
| 3.1.2    | Étude du profil d'un groupe d'utilisateurs | 16        |
| 3.2      | Recherche d'images sur la Toile            | 19        |
| 3.3      | Recherche d'images par le contenu          | 21        |
| 3.4      | Évaluation                                 | 24        |
| <b>4</b> | <b>Formalisation</b>                       | <b>25</b> |
| 4.1      | Définitions de base                        | 26        |
| 4.1.1    | Descripteurs                               | 26        |
| 4.1.2    | Base d'images classifiée                   | 27        |
| 4.2      | Principe                                   | 28        |
| 4.2.1    | Sans le profil de l'utilisateur            | 29        |
| 4.2.2    | Avec le profil utilisateur                 | 29        |
| 4.3      | Schéma général                             | 31        |
| <b>5</b> | <b>Description du problème</b>             | <b>32</b> |
| 5.1      | Indexation                                 | 32        |
| 5.1.1    | Les descripteurs utilisés                  | 32        |
| 5.2      | Classification                             | 36        |
| 5.2.1    | Introduction                               | 36        |
| 5.2.2    | Mesure de similarité                       | 37        |
| 5.2.3    | Similarité multi-vectorielle               | 37        |
| 5.2.4    | Méthodes de classification                 | 38        |

|          |   |           |
|----------|---|-----------|
| 5.3      | Expérimentation . . . . .                 | 40        |
| 5.3.1    | Vue globale du système . . . . .          | 40        |
| 5.3.2    | Environnement expérimental . . . . .      | 40        |
| 5.3.3    | Classification en clusters . . . . .      | 42        |
| 5.3.4    | Personnalisation . . . . .                | 44        |
| 5.3.5    | Tests sur l'évolution du profil . . . . . | 44        |
| 5.4      | Résultats . . . . .                       | 45        |
| <b>6</b> | <b>Conclusion et perspectives</b>         | <b>47</b> |

# Chapitre 1

## Présentation du sujet

### Objectifs du sujet

Les systèmes de recherche d'information ont pour objectif l'assistance à l'accès à des masses d'information pas ou peu structurées. Une étude de l'université de Berkeley estime la quantité d'information produite en 2003 à cinq exaoctets, dont 92 % stockés de manière numérique. La loi de Moore, toujours valide pour prévoir l'évolution de la complexité des circuits intégrés, semble s'appliquer également à l'accroissement global de l'information produite : une augmentation régulière de 30 % par an a été constatée entre 1999 et 2002. Cela amène à traiter le vaste problème du « passage à l'échelle ». L'accroissement du volume des collections a des répercussions immédiates sur l'étape d'évaluation d'une requête tant sur l'efficacité (rapidité) que sur l'efficacité (qualité). L'efficacité correspond à la complexité en temps de l'interrogation. Le temps de réponse à une requête doit toujours rester très court (quelques secondes) quelle que soit la taille du corpus. Il devient donc indispensable de limiter très rapidement et drastiquement le sous-ensemble des documents candidats, ceux qui seront examinés plus finement. L'impact sur l'efficacité est lié à la quantité de documents qu'il faut classer selon leur pertinence vis-à-vis de la requête. Dans le cas des données multimédias, ces dernières sont déjà volumineuses (images) à très volumineuses (vidéos). Mais surtout, les métadonnées qui les indexent sont elles-mêmes volumineuses, notamment lorsqu'il s'agit d'information de bas niveau (couleur, texture, timbre, quantité de mouvement, trajectoires, etc.). Malheureusement, pour ces métadonnées là, le passage à l'échelle se heurte très rapidement à la « malédiction de la dimensionnalité ». Cela signifie qu'au-delà d'un seuil de 10 à 15 propriétés à indexer simultanément, les performances des index proposés (arbres -TV, -SR, -X, -A, etc.) sont aussi mauvaises que le parcours séquentiel total. Des études préliminaires ont mis en évidence des voies de recherche possibles. En conclusion, les éléments de solution liés au traitement des requêtes sont de plusieurs ordres :

- déporter autant que possible les traitements coûteux dans la phase d'indexation ;
- remplacer les traitements coûteux qui persisteraient dans la phase d'appariement par des heuristiques ;
- élaguer les traitements en tenant compte d'autres facteurs comme l'utilisateur, l'usage et le niveau d'abstraction des informations ;

- revoir les modèles et notamment établir de nouvelles mesures, les descripteurs statistiques « standards » devenant trop peu discriminant dans des collections volumineuses hétérogènes, incomplètes et multimédias.

Nous nous intéressons plus particulièrement à la prise en compte du profil de l'utilisateur. Dans la phase d'indexation, il permettra de classer les documents suivant les centres d'intérêt des groupes d'utilisateurs. Dans la phase de recherche, il permettra de réduire rapidement, de manière auto-adaptative, l'ensemble des documents candidats en fonction du profil de l'utilisateur. De manière plus générale, la prise en compte du profil de l'utilisateur dans toutes les étapes du processus de la recherche d'information devrait améliorer l'efficacité et l'efficacités dans les collections volumineuses.

## Travail à réaliser

Le travail s'articulera autour de la modélisation et de l'exploitation d'un profil utilisateur pour les données multimédias. Le but est d'exploiter le profil pour :

1. pré-classer les données ou les sources de données susceptibles de satisfaire l'utilisateur ;
2. éviter autant que possible des accès effectifs aux systèmes sous-jacents ;
3. mettre à jour le profil en fonction des accès réalisés et des résultats obtenus.

Ce travail s'inscrit dans les thèmes de recherche de l'équipe Atlas-GRIM. Cette équipe s'intéresse à la gestion de l'information multimédia et au résumé de bases volumineuses dans un cadre réparti.

Quelques éléments plus précis sur les axes de recherche sont :

1. gestion de l'information multimédia :
  - modélisation et indexation,
  - interrogation par le contenu et navigation,
  - résumés multimédias et personnalisation,
  - mobilité et services multimédias répartis ;
2. génération de résumés à partir de bases volumineuses :
  - outils de synthèse de l'information,
  - robustesse des informations synthétisées,
  - génération de résumés à partir de données complexes,
  - évaluation des résumés (granularité, disponibilité).

Dans ce travail, nous ne couvrirons bien entendu pas l'ensemble des points d'études. Il s'intéresse plus particulièrement à la gestion de l'information multimédia. Toutefois, ce travail, par la nécessité de classer les informations, adresse également le thème du résumé de bases volumineuses. Par ailleurs, l'aspect efficacité de l'étude est lié à un travail en cours qui exploite le parallélisme d'une grappe de machines.

## Contribution

L'absence de propositions, à notre connaissance, sur le profil des utilisateurs dans la gestion par le contenu des données multimédias nous a amené à prendre une direction de travail qui, tout en tenant compte des apports de la recherche d'information textuelle, s'en écarte assez. La formalisation retenue des profils a donc été celle imaginée *a priori* d'une relation pondérée entre (groupes d') utilisateurs et classes d'images.

Nous avons construit une vérité-terrain en construisant manuellement un tel profil pour l'auteur : une relation de préférences avec un ensemble de classes d'images construites sur des critères « objectifs ».

De premières expérimentations ont permis de vérifier que la prise en compte de la rétroaction dans les recherches permet de faire converger un profil initialement vierge vers le profil cible. Cette convergence reste toutefois lente, ce qui n'est pas nécessairement un mal puisque cela évite d'enfermer l'utilisateur dans un sous-ensemble d'images de la base avant qu'il n'ait effectivement pu donner un avis sur un échantillon représentatif de cette dernière. Toutefois, par manque de temps, nous ne savons pas si la convergence ne s'accélérait pas après un certain temps.

Une autre expérimentation a été entamée mais non menée à son terme. Le but était d'exploiter le profil pendant qu'il se construit. Sous l'hypothèse de convergence observée, nous pensons que l'introduction d'un biais en faveur des classes d'images les plus appréciées, toute chose étant égale par ailleurs, permettrait d'améliorer la convergence sans nuire à l'éviction de classes d'images vraiment peu appréciées. Le programme doit être entièrement automatisé pour permettre une analyse statistique, la rétroaction manuelle devant très rapidement lassante.

Enfin, une autre approche, visant directement à l'efficacité, a été de sélectionner les images en triant les clusters avant d'accéder à leurs images. La première version est trop brutale car elle suppose le profil (ou relation de préférence) déjà connu. Il faut donc combiner cette version avec la précédente pour effectuer les expérimentations avant de la rendre disponible à l'utilisateur.

De manière générale, il se dégage d'après les expérimentations que la personnalisation améliore progressivement l'efficacité du système et souvent le temps de réponse d'une recherche puisque le nombre de classes d'images sélectionnées va forcément diminuer.

## Plan de l'étude

Le plan de ce rapport se divise en quatre parties :

- La première partie comporte la présentation et la structuration du rapport.
- La seconde partie consiste en une étude bibliographique pour avoir connaissance de ce qui a déjà été fait dans le domaine et positionner le travail que nous devons réaliser. Dans cette partie, les objectifs que l'on souhaite atteindre ainsi que la démarche à suivre sont posés.
- La troisième partie décrit les décisions techniques prises ou validées afin de mettre en œuvre notre étude.
- La quatrième partie présente les étapes parcourues, les résultats obtenus et les conclusions qui en ont été tirées, ainsi que les perspectives envisageables.

# Chapitre 2

## Introduction

La dernière décennie a vu une explosion du nombre de données numériques (image, vidéo). La mise à portée de chacun d'appareils photos et de caméras numériques, conjuguée avec la montée en puissance et en stockage des ordinateurs, ont contribué à cette augmentation ce qui a créé un besoin pressant de savoir gérer ce volume afin d'en profiter au mieux. Parmi ces données, nous nous intéressons plus particulièrement aux images numériques.

Une fois les images stockées, elles restent noyées au sein d'une très grande quantité d'information. Il est possible de retrouver une image en utilisant un système de recherche « classique », c'est-à-dire textuel, mais le travail de saisie et de maintenance des annotations manuelles devient très pénible et/ou coûteux. Utiliser les descriptions textuelles situées à proximité des images, comme le font les moteurs de recherche sur la Toile, ne garantit pas l'adéquation de la description à l'image. Plus généralement, il reste encore difficile à très difficile de retrouver efficacement une image répondant à un besoin très précis lorsque l'on ne s'adresse pas un professionnel (ex. : Corbis<sup>TM</sup>).

Une approche nouvelle et complémentaire a vu le jour [DJW05]. Les systèmes de recherche d'images par le contenu (*Content-based Image Retrieval Systems* ou CBIRs) indexent des images selon des descripteurs extraits automatiquement depuis le contenu de l'image et permettent de rechercher les images au sein d'une base de données en fonction de leurs caractéristiques visuelles. Ces caractéristiques sont généralement stockées dans un « vecteur » numérique de descripteurs visuels. Dans ces systèmes, le résultat d'une requête est une liste d'images ordonnées en fonction de leur similarité entre leur descripteurs visuels et ceux de l'image requête, en utilisant une mesure de distance donnée. Plus la distance entre les descripteurs visuels de deux images tend vers zéro, plus les images sont considérées comme similaires.

Apparaissent alors deux problèmes importants :

- l'*efficacité*, ou performance dans les temps de recherche ;
- l'*efficacité*, ou qualité des résultats obtenus.

Trouver *rapidement* une image semblable à une image requête dans une grande collection est devenu un problème crucial. Au-delà d'un seuil de 10 à 15 propriétés à indexer simultanément, les performances des index proposés dans la littérature, très nombreux (arbres-TV [LJF94], -SR [KS97], -X [BKK96], -A [SYUK00], etc.), sont aussi mauvaises que le parcours séquentiel total. C'est ce que l'on dénomme la « *malédiction de la dimensionnalité* » [BBKK97] ; si elle

grève l'efficacité des méthodes proposées, elle limite aussi les performances de toute proposition future...

De plus, il existe entre les descripteurs, ou métadonnées de bas niveau, et la sémantique d'une image, une différence forte, ce que l'on appelle le « fossé sémantique / numérique ». Une image peut ainsi contenir plusieurs significations suivant l'interprétation des utilisateurs [SGJ01]. Ainsi, selon le contexte de la recherche, la similarité sémantique entre les images peut être très différente. Cela montre la difficulté de déterminer des caractéristiques visuelles correspondant à la sémantique. Plus encore que pour des recherches *via* des descriptions textuelles, la recherche par le contenu pose un problème d'efficacité.

Bien qu'à terme les deux approches doivent être combinées, nous nous focalisons dans ce travail sur la recherche par le contenu.

## 2.1 Objectifs poursuivis

Le problème n'est plus tant la disponibilité de l'information mais la temps *optimal* pour avoir des résultats *pertinents* relativement aux besoins précis d'un utilisateur. Cela est dû à plusieurs facteurs liés, d'une part, à la représentation de l'information (des images dans notre cas) et, d'autre part, au degré de prise en compte effective de l'utilisateur dans le modèle de recherche mis en œuvre.

Dans ce travail, nous allons aborder la personnalisation du processus de recherche comme un outil permettant d'améliorer l'efficacité des outils de recherche d'images par le contenu. En supposant que l'ensemble des images est préalablement classé (hors-ligne) en un ensemble de classes *similaires*, l'objectif est d'exploiter des informations extraites à partir de l'utilisateur (préférences ou comportements) dans le but d'améliorer une ou plusieurs phases du processus de recherche, et de les sauvegarder dans des *profils* afin de restreindre les futurs accès à l'ensemble des classes les plus pertinentes.

La figure 2.1 illustre l'idée générale de ce travail. On y voit un ensemble d'images classées. (Le processus de classification utilisé est jugé secondaire ici, même s'il nous faudra bien en choisir un durant la phase d'expérimentation et/ou faire des hypothèses sur l'algorithme lui-même : complexité, incrémentalité, etc., et les caractéristiques des classes obtenues : centroïdes, dispersions, etc.)

Si la requête d'un utilisateur sollicite, par exemple, mille classes (la figure n'en représente que trois), et que le système connaît les préférences de l'utilisateur parmi les classes retenues (les classes hachurées ne lui plaisent pas, par exemple, il déteste les images très sombres, etc.), alors sa requête peut être affinée. La recherche pourra se limiter avec bénéfice à l'intersection entre les classes susceptibles de contenir des images similaires à ce qui est spécifié dans la requête et celles contenant des images qui plaisent en général.

Le problème peut donc se définir comme deux sous-problèmes liés :

- Comment modéliser et faire évoluer ce profil afin qu'il regroupe les préférences d'un utilisateur ou d'un groupe d'utilisateurs ?
- Comment définir la relation entre profils et classes ?

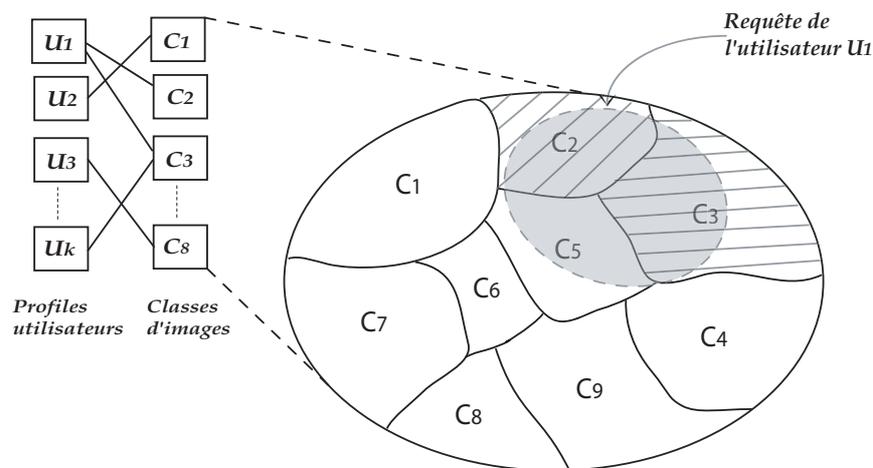


FIG. 2.1 – Schéma général et objectifs

Le travail bibliographique consiste à voir ce qui a été fait dans le domaine de la « *personnalisation* » et d'en dégager une synthèse pour résoudre le problème posé dans le cas de l'image, soit proposer une adaptation et/ou une amélioration de l'existant.

## 2.2 Problématique

Les problèmes majeurs de la personnalisation concernent :

1. la modélisation ;
2. l'initialisation ;
3. l'évolution ;
4. l'utilisation ;

du profil d'un utilisateur.

### 2.2.1 Modélisation et initialisation du profil

Les informations qui caractérisent un utilisateur sont représentées dans un profil. Le contenu et la manière de représenter les données du profil varient selon l'application. Intuitivement la manière la plus simple d'initialiser un profil est la saisie manuelle des paramètres par l'utilisateur. Malgré la simplicité de l'approche, les techniques actuelles vont vers une implication minimale du côté de l'utilisateur pour au moins deux raisons :

- Le processus de saisie manuelle peut être long ; il n'est pas raisonnable de s'attendre à une indexation pertinente si ce n'est pas un professionnel qui effectue le travail (assurant qualité et quantité d'annotations), ou a défaut un travail collaboratif (pouvant assurer quantité sinon qualité des annotations).

- Bien souvent le client a des idées floues sur ses futures exigences ; il a par conséquent du mal à exprimer clairement ses intentions. Il s'agit à ce niveau de trouver des techniques d'obtention automatique des besoins de l'utilisateur en utilisant son retour (*feedback*), en se basant sur le passé du client, voire sur l'expérience des *autres* utilisateurs.

### **2.2.2 Évolution et utilisation du profil**

L'évolution du profil est un processus délicat qui nécessite l'interprétation des actions de l'utilisateur. Cela va être exploré dans l'état de l'art. Quant à l'utilisation des informations du profil, la proposition faite ci-dessus n'est qu'une façon naïve d'en tirer partie dans un système de recherche d'images par le contenu. Il y a sans doute d'autres manières d'intégrer les préférences de l'utilisateur dans les requêtes ; nous espérons acquérir de l'expérience et en tirer des conclusions à l'issue de ce travail.

# Chapitre 3

## État de l'art

Cette partie du rapport aborde les problèmes mentionnés ci-dessus en analysant des approches existantes et en expliquant leur fonctionnement. En raison de la diversité des approches existantes, on peut dire que la personnalisation est un domaine de recherche en plein développement. En effet, les problèmes liés à la personnalisation ne sont pas encore maîtrisés ; en particulier, le problème de la création automatique des profils utilisateurs est jugé largement ouvert [WS02].

### 3.1 Recherche d'information textuelle

La personnalisation a été largement utilisée dans les systèmes de recherche d'information textuelle. C'est pour cette raison que nous tiendrons compte de ce qui a été fait dans ce domaine.

En général, les systèmes de personnalisation modélisent le profil de l'utilisateur et apprennent ses préférences. Ces informations sont ensuite utilisées pour assister les utilisateurs dans leurs requêtes. Cette modélisation peut se faire de deux manières :

- explicitement : avec l'aide active de l'utilisateur qui remplit un questionnaire sur son profil ;
- implicitement : sans l'intervention *directe* de l'utilisateur, mais en récupérant autant d'information que possible sur son comportement face au système.

On en déduit que le modèle peut respectivement privilégier une approche *statique* ou *dynamique*. Toutefois, et dans les deux cas, afin de garantir l'évolution d'un profil, la plupart des systèmes ont besoin d'un ensemble de données à étudier afin de construire et de faire évoluer les profils. En se basant sur ce critère, les approches peuvent se classer en deux catégories, que nous allons présenter plus en détail dans les sections qui suivent :

- La première prend en considération un seul utilisateur, par exemple le regroupement des pages Web en classes selon l'accès de l'utilisateur en se basant sur son historique, chaque classe indiquant les pages qui sont accédées « simultanément » lors d'une session, quand l'utilisateur sollicite l'une d'entre elles, le système lui propose les pages les plus pertinentes.
- La seconde vise le comportement d'un groupe d'utilisateurs afin de découvrir le comportement général (ces systèmes sont utilisés dans le commerce électronique par exemple).

| Nom du système               | Structure de donnée      | Évolution                          |
|------------------------------|--------------------------|------------------------------------|
| WebMate [CS98]               | Ensemble de vecteurs     | Bouclage de pertinence             |
| WebACE [HBG <sup>+</sup> 98] | Clusters                 | Mandataire                         |
| Personal WebWatcher [Mla99]  | Modèle vectoriel         | Mandataire                         |
| News Dude [BP99]             | Modèle probabiliste      | Bouclage de pertinence             |
| Alipes [WIY99]               | Modèle vectoriel         | Bouclage de pertinence (+ négatif) |
| Pretschner et Gauch [AS99]   | Hiérarchique (ontologie) | Bouclage de pertinence             |
| Hoashi [HMIH00]              | Modèle vectoriel         | Bouclage de pertinence (+ négatif) |

TAB. 3.1 – Profil d’un utilisateur pour la recherche textuelle

### 3.1.1 Étude du profil d’un utilisateur

Le tableau 3.1 résume les approches avancées par quelques propositions selon le premier axe, celui qui s’occupe du profil d’un seul utilisateur. Deux tâches sont cruciales dans la modélisation d’un profil :

- prendre en considération qu’un utilisateur peut avoir plusieurs domaines d’intérêts ;
- prévoir l’évolution incrémentale du profil.

Pour la modélisation des intérêts, WebACE [HBG<sup>+</sup>98] utilise un algorithme de classification pour partitionner les accès au journal d’un utilisateur en classes qui indiquent les domaines d’intérêts de l’utilisateur. Alipes [WIY99] et WebMate [CS98] exploitent un espace vectoriel où chaque domaine est représenté par un vecteur de terme. L’approche bayésienne naïve est utilisée dans News Dude [BP99] pour calculer la probabilité qu’un document appartienne à un domaine d’intérêt de l’utilisateur. Bien que ces systèmes prennent en compte une multitude de domaines par utilisateur, ils supposent que les domaines sont disjoints. Pretschner et Gauch [AS99] proposent un modèle qui organise les domaines d’intérêts d’un utilisateur en une structure hiérarchique au lieu d’un ensemble de domaines indépendants.

La prise en compte des changements d’intérêts nécessite des données fournies en exemples (*training data*) pour mettre à jour les profils. News Dude, WebMate, Hoashi *et al.* [HMIH00] et Alipes utilisent le bouclage de pertinence explicite, c’est-à-dire que l’utilisateur évalue manuellement la pertinence de la page. Dans la majorité des cas, seulement les exemples positifs sont pris en compte (c’est-à-dire les pages qui intéressent l’utilisateur). Alipes et Hoashi *et al.* [HMIH00] trouvent que les informations négatives peuvent être importantes pour éliminer l’incertitude dans les profils. Bien que le bouclage de pertinence soit une méthode sûre, les utilisateurs sont parfois impatients. Personal WebWatcher [Mla99] et WebACE utilisent le mandataire (*proxy*) pour obtenir des données exemples implicitement.

### 3.1.2 Étude du profil d’un groupe d’utilisateurs

Dans l’étude du profil d’un *groupe* d’utilisateurs, le principe fondamental se base sur l’extraction de l’information qui concerne le comportement de navigation d’un site Web donné (à partir des *cookies*, journaux, etc.). En appliquant des méthodes statistiques ou de fouille de données,

des groupes « *patterns* » qui décrivent le comportement de navigation d'un ensemble d'utilisateurs peuvent être identifiés [SCDT00]. Ce processus se fait en trois étapes :

1. la préparation des données ;
2. l'identification des groupes ;
3. l'analyses des groupes (composantes).

### **Collecte des données**

Les données sont extraite du serveur Web (journaux, *cookies*, etc.), du clients qui se connecte au serveur ou encore de sources intermédiaires telles que le serveur mandataire (*proxy*) ou les renifleurs (*sniffers*).

### **Pré-traitement des données**

Dans cette phase le bruit et les données redondantes sont éliminées.

### **Reconnaissance des groupes (*Pattern Discovery*)**

À ce stade, la découverte de connaissances se fait en appliquant des méthodes d'apprentissage (*machine learning*) ainsi que des techniques statistiques, clustering, classification, règles d'association et les « détections de formes séquentielles » (*sequential pattern discovery*). (Toutes ces méthodes sont très bien expliquées dans [SCDT00].) Les groupes extraits correspondent au comportement des utilisateurs et sont utilisés pour construire les profils des utilisateurs pour la personnalisation du site.

### **Post-traitement des connaissances**

Dans cette dernière partie, les informations extraites sont utilisées dans des modules de personnalisation.

La figure 3.1 résume graphiquement ces étapes. Dans la première partie, un pré-traitement est appliqué pour identifier les information des utilisateur : les sessions, les pages visitées, etc. Ensuite des méthodes statistiques ou de fouille de données sont appliquées à ces informations pour détecter les groupes qui représentent des informations importantes pour qu'elles puissent être analysées dans la dernière partie du processus.

Quelques systèmes modélisant les groupes d'utilisateurs sont représentés dans le tableau 3.2 avec des indications sur les techniques qu'ils exploitent. Les descriptions de quelques-uns sont fournies ci-dessous.

### **SETA**

SETA est un prototype qui peut être utiliser pour construire de site de commerce électronique personnaliser en se basant sur les préférences des utilisateurs. Le système initialise le profil utilisateur au début de la session et à partir de ce modèle il peut fournir les services :

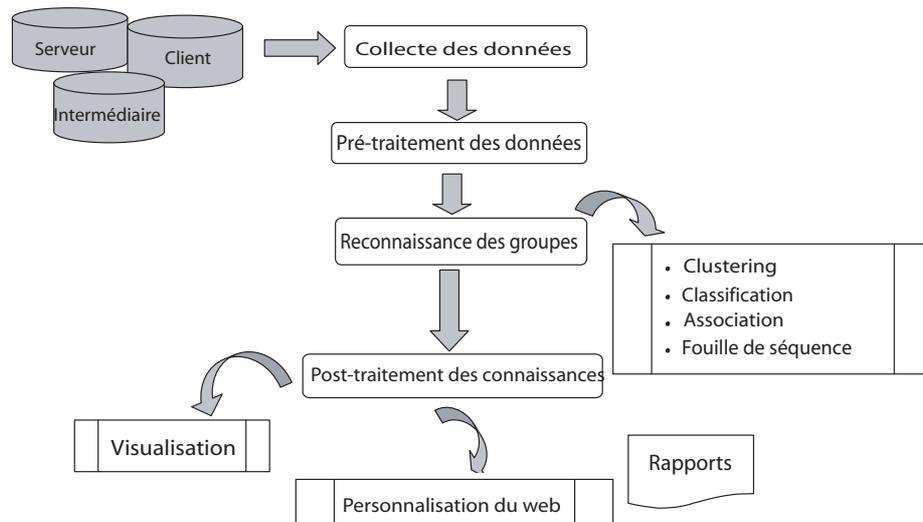


FIG. 3.1 – Processus de reconnaissance du comportement d’un groupe d’utilisateurs

- La recommandation des produits à partir de ce qui a été déjà visiter.
- La personnalisation du contenu en variant les descriptions techniques présentées pour chaque produit.
- La présentation de plusieurs variantes de combinaisons de produits.

## TELLIM

TELLIM est un autre prototype qui personnalise les apparences des pages Web à chaque étape de navigation de l’utilisateur. Les pages qui contiennent des détails sur les produits sont créés dynamiquement et présentées à l’utilisateur suivant ses préférences.

## Schwarzkopf

Le système Schwarzkopf a été utilisé pour personnaliser le site de la conférence UM2001 selon les intérêts des visiteurs. Le système collecte des informations à propos des utilisateurs du serveur pour construire le modèle qui les représente, ce dernier sera mise à jour au fur et à mesure par les recherches des visiteurs.

## NETMIND

NetMind est une solution personnalisée qui produit des recommandations multi-utilisateur ; un module spécialisé est chargé de faire une fonction de mandataire renversé « Reverse Proxy », ce dernier reçoit les requêtes du serveurs Web est les diffuse aux utilisateurs suivant leurs profils.

| Système                     | Collecte des donnée |         |       |          | Reconnaissance |                |             |            |
|-----------------------------|---------------------|---------|-------|----------|----------------|----------------|-------------|------------|
|                             | journaux            | cookies | Agent | Do. util | Clustering     | Classification | Association | Séquentiel |
| SETA [AT00]                 |                     |         | ✓     | ✓        |                |                | ✓           |            |
| Tellim [J99]                |                     |         | ✓     |          |                | ✓              |             |            |
| Schwarzkopf [Sch01]         | ✓                   |         |       |          |                |                | ✓           |            |
| Oracle9iAS <sup>1</sup>     |                     |         | ✓     | ✓        |                | ✓              | ✓           |            |
| Netmind <sup>2</sup>        | ✓                   |         |       |          | ✓              | ✓              |             | ✓          |
| Re :action <sup>3</sup>     |                     |         |       |          | ✓              |                | ✓           |            |
| WebPersonalizer [MCS00]     | ✓                   | ✓       | ✓     |          | ✓              |                |             |            |
| Yan <i>et al.</i> [YJGMD96] | ✓                   |         |       |          | ✓              |                |             |            |
| Kamdar et Joshi [KJ00]      | ✓                   | ✓       |       |          | ✓              |                |             |            |
| SiteHelper [NW97]           | ✓                   |         |       |          |                | ✓              |             |            |
| WUM [SPF99]                 | ✓                   |         |       |          |                |                |             | ✓          |

TAB. 3.2 – Profil d’un groupe d’utilisateurs pour la recherche textuelle

### Re:action

Enfin, Re:action est un système qui délivre des fonctionnalités personnalisée au visiteurs du site suivant leurs contextes actuels, leurs préférences, leurs historiques.

Dans cette section, nous avons effectué un bref survol des grands axes de recherche dans le domaine de la personnalisation dans la recherche textuelle. Cette étude nous est utile car tel que nous allons le voir par la suite, plusieurs propositions dans le domaine de recherche d’image, se sont inspirées de la recherche textuelle.

Dans le processus de personnalisation, la première tâche est de savoir quoi mettre dans le profil utilisateur, d’où tirer ces informations et comment les organiser.

Pour les systèmes qui étudient un seul utilisateur, les sources de données sont les données personnelles (questionnaire, historique, etc.) pour les groupes d’utilisateurs, ces sources se trouvent dans les serveurs, des mandataires en plus de l’utilisateur lui-même. La majorité des systèmes de sont mis d’accords que le bouclage de pertinence est une méthode importante pour l’évolution des profils.

## 3.2 Recherche d’images sur la Toile

Dans le domaine de la recherche d’images sur la Toile, on peut réutiliser la majorité des techniques proposées dans la recherche textuelle, du moment que l’indexation des images se base sur les mots apparaissant dans les pages.

[LYZ01] a abordé la personnalisation de la recherche d’image sur la Toile. En plus de définir le profil par rapport à ce que cherche l’utilisateur, un autre profil décrit « ce que l’objet est du point de vue de l’utilisateur » : c’est le *profil commun*. Ce dernier maintient pour chaque objet  $O_i$  de la base, de taille  $M$ , une liste de mots clés pondérés qui seront considérés comme des descripteurs sémantiques :

$$D_i = \{ \langle t_{i,1}, w_{i,1} \rangle; \dots; \langle t_{i,N}, w_{i,N} \rangle \}$$

où  $t_{i,j}$  est le  $j^e$  mot clé associé à  $O_i$ , avec le poids  $w_{i,j}$ . Le profil commun est représenté par  $\{D_i\}, i \in \{1, \dots, M\}$ .

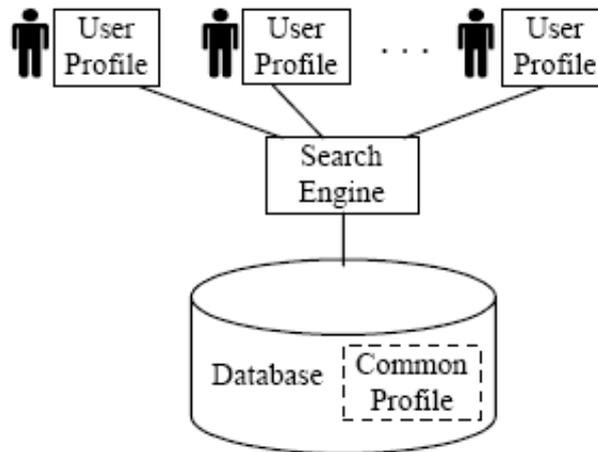


FIG. 3.2 – Modélisation du profil commun et le profil utilisateur

Le profil de l'utilisateur a la même représentation sauf que le poids du terme peut être négatif ce qui veut dire que l'image n'est pas décrite par ce terme.

Dans cette approche, l'utilisateur soumet une requête sous la forme de mots clés puis évalue la pertinence des images (positives ou négatives), ce qui permet au système de savoir que les images évaluées sont pertinentes ou pas par rapport aux termes de la requête. Le système peut alors ajouter ou enlever ces termes des descripteurs d'objets. Dans ce système, l'algorithme met à jour le profil commun à partir d'un exemple d'objet  $O_i$  issu du bouclage de pertinence comme suite :

1. S'il n'existe aucun descripteur dans le profil commun pour l'image  $O_i$  alors créer un descripteur vide.
2. Si  $O_i$  est évalué comme exemple positif, voir s'il existe un mot de la requête dans  $D_i$ . Si c'est le cas, le mot est ajouté avec un poids initial  $a$ , sinon le poids du mot est incrémenté de  $a$ .
3. Si  $O_i$  est évalué comme exemple négatif, voir s'il existe un mot de la requête dans  $D_i$ . Si c'est la cas, le poids du mot est décrémenté de  $c \times a$ . Si le résultat est inférieur à zéro, le mot est supprimé de  $D_i$ .

Dans la proposition de cette méthode, la constante  $a$  est fixée à un 1 et la constante  $c$  est égale à 5.

Le même algorithme est utilisé pour mettre à jour le profil de l'utilisateur, sauf que les termes négatifs sont gardés dans les descripteurs.

Ne nous attardons pas trop sur cette catégorie de recherche parce que la majorité des méthodes de la recherche textuelle peuvent être utilisées pour « pseudo-indexer » des images *via* les mots de la page dans laquelle elles se trouvent. De plus, nous nous intéressons ici à la recherche par le contenu uniquement.

### 3.3 Recherche d'images par le contenu

Il a été remarqué, durant cette recherche bibliographique, que le domaine de la personnalisation des systèmes de recherche d'images par le contenu étant relativement récent, les publications sur la modélisation de l'utilisateur ne sont pas très nombreuses.

Contrairement aux documents textuels, les documents multimédias ne possèdent pas d'indices naturels comme les mots (ou les groupes de mots) qui peuvent servir à représenter leur contenu et à partir desquels on peut les retrouver. Ils utilisent des descripteurs extraient automatiquement (couleur, texture ou format). Ceux-ci peuvent être associés à l'image dans son ensemble ou à une partie seulement de l'image.

Cependant, les mots sont une manière naturelle et conviviale pour exprimer le besoin humain, c'est pour cette raison que les premiers systèmes les ont utilisés pour décrire les images. Puis, au vu du nombre toujours croissant d'images à indexer, il a fallu passer à l'extraction automatique des descripteurs d'images. Ils sont compliqués à intégrer à une requête, d'où plusieurs solutions ont été proposées pour formuler une requête compréhensible par l'utilisateur et le système en même temps :

- Certains systèmes ont adopté la méthode de la requête par exemple : l'utilisateur commence sa requête par une image qui lui semble proche de son besoin (image de la base, dessinée par l'utilisateur, etc.). Comme il est difficile d'exprimer, *via* un seul croquis ou une seule image exemple, la similarité attendue entre l'exemple et les réponses valides, le bouclage de pertinence permet de raffiner la première requête.
- D'autres systèmes ont essayé de transformer les descripteurs de bas niveau des images en termes linguistique (exemple : à partir de l'histogramme de couleur, on peut tirer plusieurs informations : image prise le jour, la nuit, couleurs dominantes, etc.) afin de se rapprocher de la recherche d'information textuelle.
- La combinaison des deux approches, les *mots* et les *images*, devient de plus en plus intéressante, de l'avis de plusieurs auteurs [CBP99, MKH<sup>+</sup>00, CJBC01], pour la personnalisation des systèmes de recherche d'images.

Dans l'approche de personnalisation proposée par [CLY02], le système utilise des descripteurs textuels et des exemples visuels en plus du bouclage de pertinence pour exprimer le besoin d'un utilisateur. Dans cette étude, le problème du fossé sémantique est supposé réglé en utilisant des termes linguistiques au lieu de descripteurs de bas niveau et la subjectivité de l'utilisateur est traduite par le bouclage de pertinence. Ce système traite la personnalisation d'un autre point de vue, il crée un profil personnel *pour chaque image*.

Ce système se base essentiellement sur les descripteurs de texture. En effet le profil se construit à partir d'un algorithme de génération de fonctions d'appartenance (*c-means*) pour remplir les termes linguistiques définis dans le tableau 3.3. (On notera la difficulté à proposer des termes distincts et clairs pour la texture, alors que cela est bien plus simple pour les couleurs.)

L'évolution du profil se fait par le bouclage de pertinence afin de détecter la texture des images.

Cette proposition traite la personnalisation du système de recherche d'images par le contenu pour la description de la texture d'image. Cependant l'utilisation des descripteurs flous va engendrer une perte de précision. En plus, la généralisation pour les autres descripteurs n'est pas

| Tamuras' Features | Linguistic Terms     |                 |                    |             |                  |
|-------------------|----------------------|-----------------|--------------------|-------------|------------------|
| Coarseness        | very fine            | fine            | medium coarse      | coarse      | very coarse      |
| Contrast          | very low             | low             | medium contrast    | high        | very high        |
| Directionality    | very non-directional | non-directional | medium directional | directional | very directional |
| Line-likeness     | very blob-like       | blob-like       | medium line-like   | line-like   | very like-line   |
| Regularity        | very irregular       | irregular       | medium regular     | regular     | very regular     |
| Roughness         | very smooth          | smooth          | medium rough       | rough       | very rough       |

TAB. 3.3 – Proposition de termes linguistiques pour les propriétés de texture

toujours possible.

D'autres travaux ont abordé le problème de la personnalisation sous une autre forme, on trouve rarement des définitions d'un modèle utilisateur mais il est pris en compte en essayant de l'intégrer dans le processus de recherche et d'interpréter ses actions afin d'enrichir le système.

Dans [ZYL02], un modèle de graphe à deux niveaux est présenté pour mémoriser les corrélations sémantiques (entre les images) progressivement tirées à partir du bouclage de pertinence. Une image est décrite à travers sa corrélation avec d'autres images *via* deux graphes (cf. figure 3.3) : le graphe visuel et le graphe sémantique. Chaque nœud représente une image et chaque arc correspond à la corrélation entre deux images ainsi reliées, le degré de corrélation est représenté par le poids de l'arc.

Au niveau visuel, les liens sont obtenus par des calculs « hors-ligne » de similarité entre les descripteurs visuels des images ; un lien est créé entre deux images qui ont une similarité très grande seulement, pour « éviter les liens inutiles ».

Pour la recherche d'image, les liens sémantiques sont d'une grande importance puisque les gens jugent la similarité entre les images d'un point de vue sémantique. Or, il y a pas de moyen automatique d'acquérir cette information. Alors, dans cette proposition, et durant le processus de recherche « en-ligne », le lien sémantique entre les images se crée après le bouclage de pertinence suivant un algorithme très simple et intuitif :

- Pour toute image qui appartient à l'ensemble des images positives, s'il y a pas un lien entre elle et l'image exemple, alors créer le lien, sinon incrémenter le poids du lien de 1.
- Pour toute image qui appartient à l'ensemble des images négatives, s'il y a un lien entre elle et l'image exemple, alors diviser le poids par 4, et si le résultat est inférieur à 1 alors supprimer le lien.

Cette approche surpasse des approches traditionnelles tant à court terme (l'intra-session) qu'à long terme (l'inter-session) ! Cependant, l'utilisation de cette méthode a deux effets secondaires :

- l'espace de recherche est réduit en un petit ensemble (le graphe) ;
- le nombre de liens augmente très rapidement avec le nombre d'images et d'interactions, rendant le stockage et l'accès extrêmement cher.

Des fonctionnalités de classification et de navigation peuvent être envisagées pour profiter du potentiel de ce modèle.

Une autre proposition [YLZ02, YLZ03] peut être considérée comme la combinaison entre les deux précédentes [LYZ01, ZYL02]. Dans cette approche deux types de profils sont définis : un général et l'autre personnel (comme dans le cas de [LYZ01]). En revanche, au lieu d'utiliser des

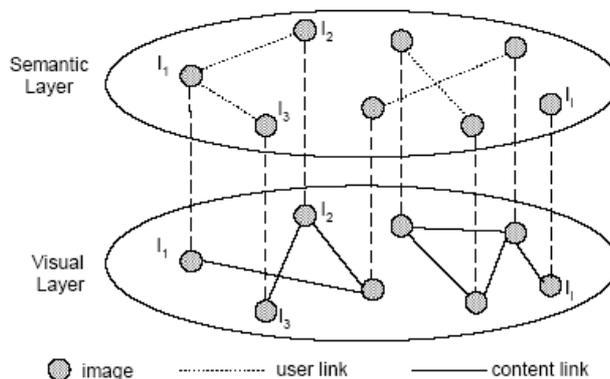


FIG. 3.3 – Modélisation des corrélations sémantiques entre les images

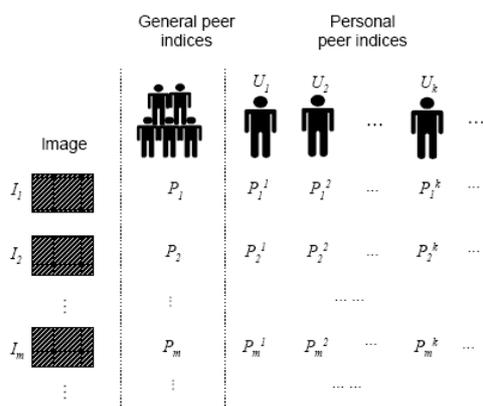


FIG. 3.4 – Index des images par d'autres images

mots clés, les auteurs ont opté pour l'indexation d'images par d'autres, corrélés avec elles (principe de [ZYL02]). Les auteurs essaient de suivre l'intuition de l'être humain pour rapprocher le système de recherche d'image au système visuel humain.

Dans l'image 3.4, on remarque que pour chaque image  $I_m$ , on peut trouver :

- une représentation générale  $P_m$  :

$$P_m = \{ \langle p_{m,1}, w_{m,1} \rangle, \dots, \langle p_{m,i}, w_{m,i} \rangle, \dots, \langle p_{m,N_m}, w_{m,N_m} \rangle \}$$

où  $p_{m,i}$  représente l'image qui est sémantiquement corrélée avec  $I_m$  jugée positive lors du bouclage de pertinence par la totalité des utilisateurs, avec un poids  $w_{m,i}$  indiquant le degré de pertinence ;

- une représentation personnelle de l'image  $I_m$  pour un utilisateur  $k$  est donnée par :

$$P_m^k = \{ \langle p_{m,1}^k, w_{m,1}^k \rangle, \dots, \langle p_{m,i}^k, w_{m,i}^k \rangle, \dots, \langle p_{m,N_m}^k, w_{m,N_m}^k \rangle \}$$

où  $p_{m,i}^k$  représente l'image qui est sémantiquement corrélée avec  $I_m$  jugée positive lors du bouclage de pertinence par l'utilisateur  $k$ , avec un poids  $w_{m,i}$  indiquant le degré de pertinence.

De par l'analogie qui existe entre cette proposition et l'indexation par « mots clés visuels », les techniques de recherche d'information textuelle ont été réutilisées. Bien que l'idée soit intuitive pour l'utilisateur, le passage à l'échelle demande le développement de stratégies efficaces pour le stockage et la recherche de tels index.

Dans la recherche d'image par le contenu, les méthodes avancées par les propositions précédentes visent essentiellement à combler le manque de la sémantique des descripteurs de bas-niveau des images. La transformation de ces dernières en termes linguistiques apporte un peu plus de sémantique aux descripteurs mais aussi des pertes de précisions. Les autres études visent à modéliser la corrélation sémantique qui existent entre les images, Pour toutes ces méthodes le passage à l'échelle s'avère très difficile.

### 3.4 Évaluation

Quelques modèles décrits ci-dessus présentent certains inconvénients concernant l'initialisation, la représentation et l'évolution du profil de l'utilisateur. Lorsque la construction du profil de l'utilisateur est guidée par le contrôle de pertinence, le système n'est opérationnel que lorsque l'utilisateur l'a *entraîné* durant un certain temps, avec un fort risque de démotivation de ce dernier. C'est sans doute essentiellement pour cette raison que les systèmes qui construisent les profils en analysant des données déjà existantes dans le système (tel que les journaux) ont un avantage par rapport au autres.

Dans tout les systèmes on remarque que le bouclage de pertinence est essentiel pour la personnalisation, pour rapprocher la vision du système de celle de l'humain.

Pour la recherche d'images par le contenu, les travaux concernant la personnalisation ne sont pas nombreux et très récents par rapport au domaine textuel. On peut dire que c'est un domaine qui n'est pas encore mature.

Dans ces travaux, la modélisation d'un profil de l'utilisateur n'a pas été faite telle que cela a pu être fait dans le domaine textuel ; c'est sans doute dû à la jeunesse du domaine ainsi qu'à la difficulté de le faire avec des descripteurs de bas niveau.

Dans notre travail, nous allons essayer de tirer profit de ce qui a été avancé par toutes les recherche précédentes (l'extraction de données et les modèles utilisateur et groupe d'utilisateurs dans la recherche textuelle, la prise en comptes de la sémantique des images et la subjectivité de l'utilisateur dans la recherche d'images par le contenu ) sans oublier le facteur de passage à l'échelle, et cela pour proposer une idée à développer et faire évaluer par des tests sur une vérité terrain.

# Chapitre 4

## Formalisation

Dans cette deuxième partie de l'étude bibliographique, malgré les difficultés à trouver une base de travail stable dans la littérature, nous allons proposer une formalisation de notre travail, tenant compte des objectifs initiaux et des constatations faites dans l'état de l'art.

L'un des premiers problèmes rencontrés dans la recherche d'images par le contenu est le choix des propriétés qui représentent au mieux les images. L'expressivité de ces propriétés dépend des types d'images. Par exemple, la couleur dominante permet de décrire correctement un coucher de soleil dans la figure 4.1, en revanche elle n'est pas très significative dans la figure 4.2 qui est mieux décrite par la texture.

Plusieurs techniques de classification [ZCH<sup>+</sup>04, CWK05, SB02], ou proches, visent à organiser les images en classes (clusters). Le principal critère de répartition assure que les images les plus proches au sens d'une mesure de similarité ou d'une distance donnée se retrouvent dans la même classe et que les classes sont les plus distinctes et éloignées possible les une des autres [Ani06]. Ces mesures sont délicates à définir.

Un autre problème important est la subjectivité de la perception humaine [RHOM98], différentes personnes, ou la même personne dans différents contextes, pouvant percevoir l'image de différentes manières. Par exemple, un groupe d'utilisateurs peut considérer que les deux images (a) et (b) de la figure 4.3 sont similaires (sur la granularité) alors qu'un autre groupe pourra dire que les images (b) et (c) sont davantage similaires (sur la couleur).

Le problème revient à définir, à partir de la classification précédente, quelles sont les propriétés qui décrivent au mieux les images *du point de vue utilisateur*.

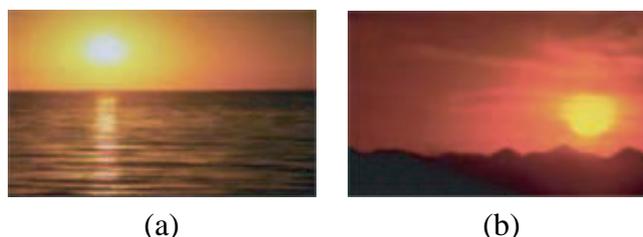


FIG. 4.1 – Exploitation de la couleur

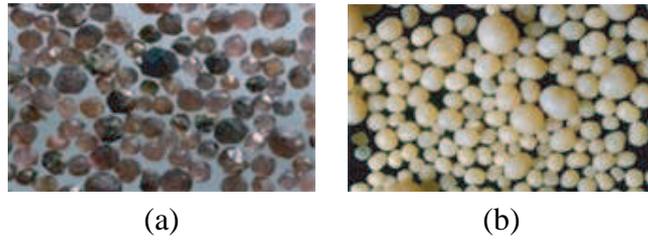


FIG. 4.2 – Exploitation de la texture

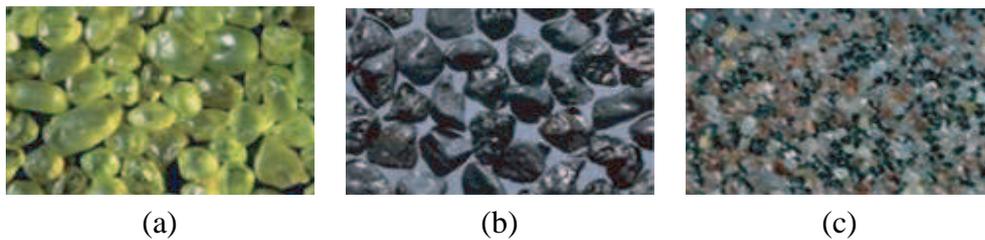


FIG. 4.3 – Différences de perception

## 4.1 Définitions de base

Nous présentons la définition de quelques termes utiles à la compréhension des principes.

### 4.1.1 Descripteurs

Il nous faut tout d'abord préciser la façon dont sont représentées les métadonnées associées aux images. Dans l'absolu, la définition doit rester très générale.

**Définition 1 (Descripteur visuel)** *Un descripteur visuel  $d(i)$  est un groupe d'informations qui résume certaines caractéristiques visuelles d'une image  $i$  au niveau des pixels, de régions de l'image et/ou de l'image dans sa totalité. Ces caractéristiques peuvent être soit globales (couleur, texture, forme), soit locales (point d'intérêt).*

En pratique, il faut bien finir par instancier cette définition. On parle souvent de « vecteur » de métadonnées. En effet, les descripteurs sont souvent simplement concaténés afin de pouvoir, par simplification, procéder à des calculs de similarités sur la base d'un espace réputé vectoriel (les données de bas niveau étant quasi-exclusivement numériques).

**Définition 2 (Distance)** *Une distance  $\Delta$  est une métrique dans l'espace des descripteurs, souvent considéré comme un espace vectoriel :*

$$\Delta : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}^+$$

*qui possède les propriétés suivantes :*

1. *séparation* :  $\forall x, y \in (\mathbb{R}^p)^2, \Delta(x, y) = 0 \leftrightarrow x = y$
2. *symétrie* :  $\forall x, y \in (\mathbb{R}^p)^2, \Delta(x, y) = \Delta(y, x)$
3. *inégalité triangulaire* :  $\forall x, y, z \in (\mathbb{R}^p)^3, \Delta(x, y) \leq \Delta(x, z) + \Delta(z, y)$

Dans le cas d'un espace vectoriel, les distances sont souvent des normes de différence. Les distances les plus fréquemment utilisées sont les distances de Minkowski, cas particuliers de la  $p$ -norme,  $\sqrt[p]{\sum_{\forall i} |x_i - y_i|^p}$ , le plus souvent pour  $p = 1, 2, \infty$  :

- distance de Manhattan :  $\sum_{\forall i} |x_i - y_i|$  ;
- distance euclidienne :  $\sqrt{\sum_{\forall i} (x_i - y_i)^2}$  ;
- distance de l'échiquier :  $\sup_{\forall i} |x_i - y_i|$ .

#### 4.1.2 Base d'images classifiée

Une distance sera indispensable pour comparer les images d'une base d'images à une image requête mais également entre elles afin de classifier les images.

**Définition 3 (Base d'images)** Une base d'images  $B$  traduit un ensemble images  $I = \{i_1, i_2, \dots, i_n\}$  par différents descripteurs  $D = \{d_1, d_2, \dots, d_m\}$  :

$$B = \bigcup_{i \in I} \bigcup_{d \in D} d(i)$$

**Définition 4 (Cluster)** Un cluster  $C_d^k$ , pour un descripteur visuel donné  $d$ , est un sous-ensemble de la base :

$$C_d^k = \{d(i) : i \in I' \subseteq I\}$$

vérifiant certaines contraintes quant à la proximité des descripteurs au sein du cluster  $C_d^k$  et vis-à-vis des autres clusters  $C_d^{k'}$  avec  $k' \neq k$ .

Plusieurs caractéristiques peuvent être associées à chaque cluster, telles que sa forme géométrique, sa taille, la distribution statistique de ses vecteurs ou encore son centroïde.

**Définition 5 (Partition de la base d'images)** La base est partitionnée selon chaque descripteur  $d \in D$ , c'est-à-dire :

$$B = \bigcup_{k=1}^{K(d)} C_d^k$$

où  $K(d)$  est le nombre de clusters pour le descripteurs  $d$ .

Cette définition est illustrée sur la figure 4.4, où l'on peut modéliser le partitionnement de la base d'image  $B$  selon chaque descripteur  $D_i$  (en haut) par plusieurs agrégations qui expriment

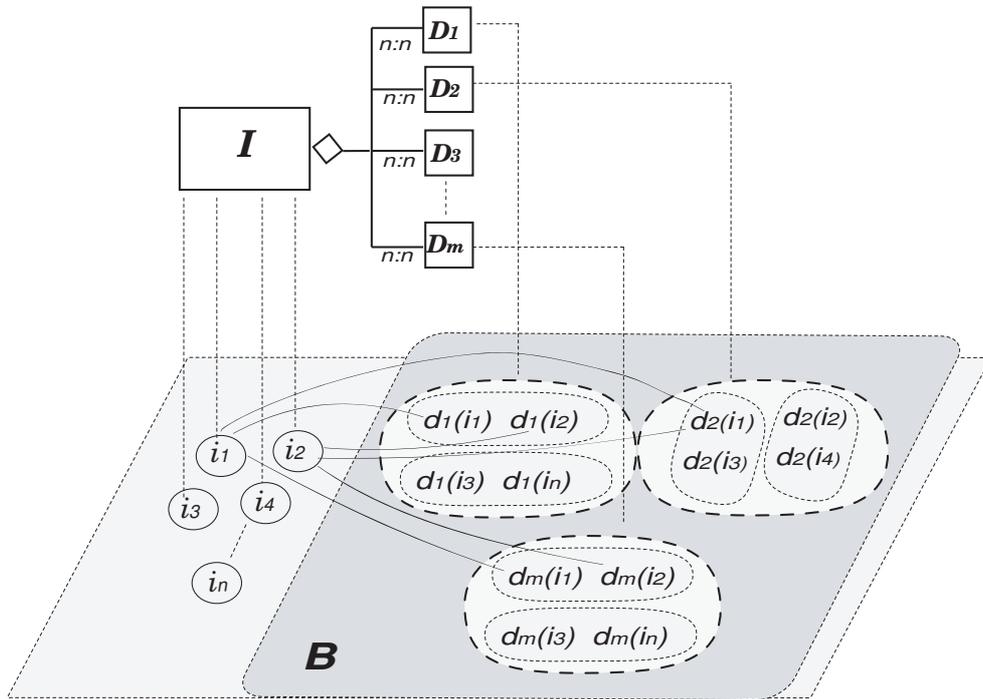


FIG. 4.4 – Définition des clusters en intension (haut du schéma) et en extension (en bas)

que chaque ensemble d'images de la base peut être décrit par un ensemble d'ensembles de descripteurs. Cela est illustré au niveau des instances dans la partie basse de la figure. Chaque descripteur  $D_i$  partitionne la base d'images en un ensemble de *clusters*. L'union des ces derniers représente la base entière décrite par chaque descripteur.

Soulignons que les définitions précédentes se simplifient considérablement si l'on ramène  $D$  à un singleton. Il vient alors que chaque image est décrite par un unique vecteur et que la base est partitionnée d'une seule manière. Par ailleurs, on ne manipule plus qu'une unique distance ; autrement dit, la combinaison de différentes métriques est prise en compte de manière homogène plutôt que devant être fourni comme un mécanisme supplémentaire.

## 4.2 Principe

Avant d'aborder le problème il faut d'abord connaître :

- Quelle est la contribution de l'utilisateur dans le système de recherche ?
- Comment apprendre à partir de ce comportement ?
- Comment sauvegarder l'information utile de manière efficace ?
- Comment l'utiliser dans le futur ?

### 4.2.1 Sans le profil de l'utilisateur

Si l'on ne dispose pas du profil de l'utilisateur, durant l'étape de recherche, les implications sont les suivantes :

1. L'utilisateur soumet au système une requête. Son but est de trouver toutes les images qui satisfassent cette requête selon certains critères visuels.
2. Les plus proche voisins d'une image requête  $I_q$  sont calculés à partir des clusters  $C_s$  les plus proches de  $I_q$  pour chaque type de descripteur. Soit  $C_q = \{C_1, C_2, C_3, \dots\}$  l'ensemble des clusters sélectionnés et triés en se basant sur leurs caractéristiques (par exemple, l'évaluation de la distance du centroïde du cluster à l'image-requête, ainsi que les distance minimum et maximum de la requête au cluster).
3. Les clusters sélectionnés sont ensuite parcourus dans l'ordre de leurs pertinences pour produire le résultat le plus vite possible ;
4. Les résultats intermédiaires sont présentés à l'utilisateur jusqu'à ce qu'il valide le résultat ou qu'il n'ait plus de clusters à parcourir.

### 4.2.2 Avec le profil utilisateur

Pour une autre requête  $I'_q$ , même similaire, tout le processus de calcul et d'optimisation sera à refaire. Notre but est de regrouper les connaissances acquises des recherches précédentes dans des « classes utilisateurs » afin d'éviter autant que possible de refaire les calculs et ainsi apporter des améliorations en terme d'efficacité. Nous proposons :

1. l'utilisation d'une méthode d'apprentissage afin d'extraire les informations utiles à sauvegarder à partir des journaux des recherches précédentes ;
2. le regroupement de ces informations dans des classes similaires (les critères de similarité restent à définir) pour faciliter la recherche ;
3. la présentation à l'utilisateur des résultats intermédiaires (s'ils existent) à partir des classes utilisateurs avant de passer aux clusters de la base ;
4. l'arrêt de la recherche si l'utilisateur juge que le résultat intermédiaire est pertinent pour sa recherche.

Il faut d'abord définir le modèle du profil utilisateur  $P_i$  qui sera le regroupement des préférences pondérées des utilisateurs par rapport aux clusters issus de la classification selon le degré d'appréciation des clusters ("aime bien", "aime", "déteste", etc.)  $D' = \{(C_1, p_1), (C_2, p_2), \dots, (C_k, p_k)\}$  regroupé à partir d'une méthode de classification sur les informations disponibles du système (par exemple, les journaux) ou introduites par l'utilisateur (en cas d'absence de journaux).

La mise à jour des profil se fait à partir du bouclage de pertinence de l'utilisateur. L'algorithme général est décrit dans la figure 4.5.

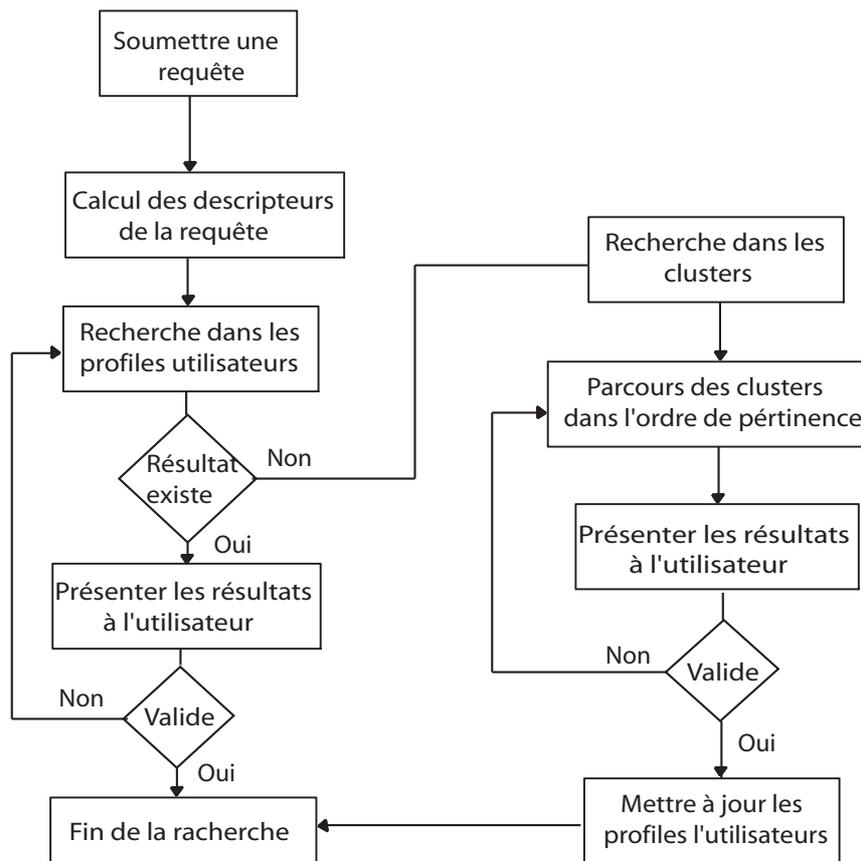


FIG. 4.5 – Algorithme général

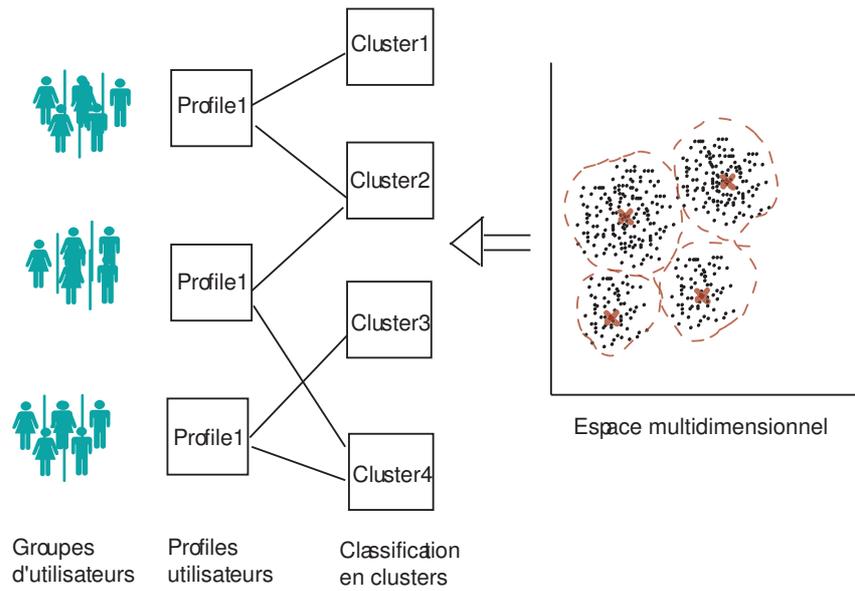


FIG. 4.6 – Fonctionnement

### 4.3 Schéma général

Pour éviter de parcourir de façon séquentielle tout les clusters candidats à une image requête, le processus de recherche par contenu peut être adapté pour proposer des résultats intermédiaires à partir d'une recherche préalable dans les « profils », d'une part, pour améliorer le temps de recherche et, d'autre part, de réduire le nombre de cycles de bouclage de pertinence.

# Chapitre 5

## Description du problème

Dans cette partie, nous décrivons les décisions techniques que nous avons dû prendre ou valider en ce qui concerne :

1. le choix d'un modèle de descripteurs ;
2. la métrique à utiliser pour calculer la distance entre les métadonnées des images ;
3. la technique de classification (ou clustering).

Nous continuons en décrivant brièvement les possibilités existantes en matière d'amélioration des résultats de recherche d'image par l'exemple, qui permettent d'obtenir finalement les images qui peuvent satisfaire au mieux la requête de l'utilisateur en se basant sur ses préférences. Enfin, nous présentons l'ébauche d'un mécanisme de suivi exploitant ces résultats.

### 5.1 Indexation

La mise en place du système nécessite une étape préalable d'indexation des images, afin d'extraire des descripteurs qui serviront à classer la base et à évaluer la similarité entre images.

Les contraintes principales imposées par le problème de l'indexation d'images par le contenu sont de trois types. La première contrainte concerne le temps de calcul, la réponse à une requête doit être rapide. La deuxième contrainte est très fortement liée à la précédente et concerne la taille de la mémoire du système. Avec la baisse du coût de ces mémoires, cette dernière contrainte peut apparaître comme mineure, cependant la taille des descripteurs est une composante importante particulièrement dans le cas des grandes bases d'images (composées de plusieurs millions d'images). La troisième contrainte est liée à la satisfaction du but de l'utilisateur puisqu'un système qui ne répond pas aux attentes de ce dernier n'est d'aucune utilité.

#### 5.1.1 Les descripteurs utilisés

Le but de l'indexation est de fournir une représentation des images permettant des recherches efficaces. Il ne s'agit pas de coder toute l'information portée par l'image mais de se concentrer

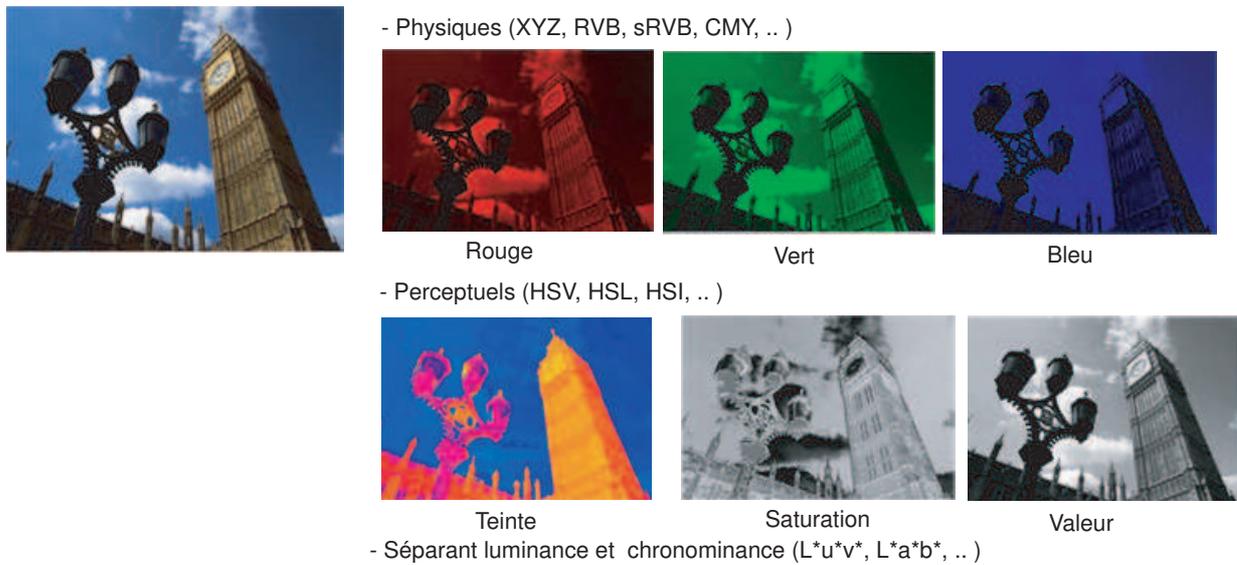


FIG. 5.1 – Espaces de couleurs

sur l'information qui permet de traduire efficacement une similarité proche des besoins exprimés par un utilisateur.

Il n'existe pas d'indexation généralisable à tout type d'application. L'analyse faite du signal se focalise généralement autour de caractéristiques simples, intuitives et génériques telles que la couleur, la texture, la forme, la géométrie et la topologie.

La norme MPEG-7 [MSS02] introduit six niveaux de description :

- format (images en couleur, en niveaux de gris, en infra-rouges, etc.) ;
- physique (couleurs principales, textures, etc.) ;
- perceptuel (couleurs chaudes, texture grossière, etc.) ;
- structurel (régions d'une image, etc.) ;
- métadonnées intrinsèques (« objets » du monde réel associés à une région, mots clés, etc.) ;
- annotations.

Dans cette étude, nous avons privilégié les caractéristiques de format et de couleur pour décrire les images.

### Descripteurs de couleur

La couleur, de par son pouvoir discriminant, est un des attributs les plus employés en indexation d'images. Les auteurs se concentrent autour de quelques grands thèmes tels que l'identification de l'espace couleur le plus discriminant [Stricker et al.95], l'étude des problèmes d'invariance aux conditions d'illumination et de prise de vue [Gros et al.97] ainsi que la combinaison avec des attributs complémentaires tels que la texture [Setchell et al.99] ou l'information spatiale [Stricker et al.96].

Les espaces de couleur le plus utilisée sont :



FIG. 5.2 – Histogramme de couleur -1

- XYZ : étalon (CIE, 1931) ;
- RGB : physique (moniteurs mais aussi oeil humain) ;
- HSV : perceptuel (teinte/saturation/valeur ; ex. : jaune/profond/clair) ;
- $L^*a^*b^*$  : euclidien.

La forme la plus simple de dispositif de couleur est l'intersection d'histogrammes de couleurs, créé par Swain et Ballard [SB91]. Les histogrammes sont faciles et rapides à calculer, robustes à la rotation et à la translation. De plus, ils imposent peu de contraintes lorsqu'on les applique sur des images.

Mais ils posent quelques problèmes :

- De grandes tailles, il est difficile de créer une indexation rapide et efficace en les utilisant tels quels.
- Ils sont sensibles à de petits changements de luminosité, ce qui est problématique dans la comparaison des images similaires acquises dans des conditions différentes.
- Ils ne possèdent pas d'informations spatiales sur les positions des couleurs.
- Ils sont inutilisables dans la comparaison partielle des images, puisque la valeur est calculée globalement sur toute l'image.

Pour résoudre ce problème, il y a deux approches : l'une ajoute des informations spatiales, l'autre trouve d'autres espaces de couleurs qui se rapprochent de la perception humaine.

La première approche ajoute des informations spatiales aux histogrammes. Stricker et Dimai [SD96] ont divisé une image en cinq blocs fixes et ils ont extrait les trois premiers moments d'inertie de chaque bloc pour créer un vecteur de descripteurs.

Pass et Zabih [GR97] ont ajouté la cohérence spatiale dans des histogrammes. Un pixel est cohérent s'il appartient à une région validée par la segmentation et incohérent dans d'autre cas. Les valeurs d'un histogramme sont divisées en deux classes en fonction des pixels correspondants à ces valeurs : classe cohérente et classe incohérente.

La deuxième approche essaie de trouver 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

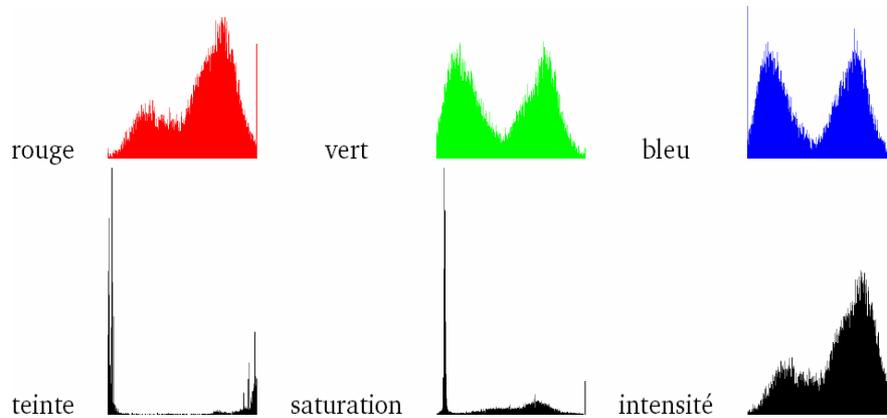


FIG. 5.3 – Histogramme de couleur -2

mais il n'est pas forcément le mieux adapté. Smeulders *et al.* [AMaR00] ont présenté quelques expérimentations intéressantes pour utiliser d'autres espaces de couleurs en indexation, tandis que [YC98] employé l'espace HVC. Selon les auteurs de ce dernier exemple, cet espace de couleur est plus près de la perception humaine et donne des résultats de requêtes plus précis.

Notre travail reprend une approche simple : diviser l'image en cinq zones (centre, haut, bas, gauche et droite), quelques descripteurs seront ainsi locaux.

### Descripteurs de format

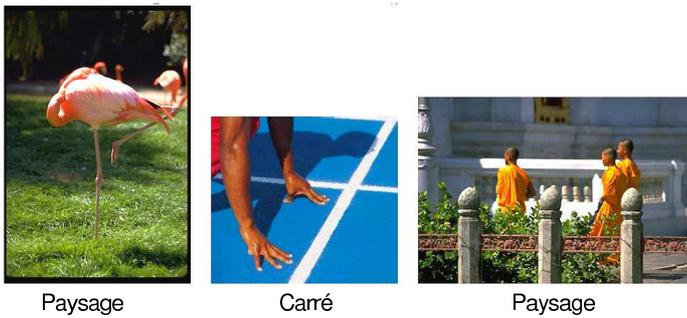
- couleur



- surface ( $= l * h$ )



– orientation ( $= \frac{1-\cos(2\alpha)}{2} \in ]0, \pi[$  avec  $\alpha = \tan^{-1} \frac{l}{h}$  l'angle de la diagonale)



– élongation ( $= | \frac{4\alpha}{\pi} - 1 | \in ]0, \frac{\pi}{2}[$ )



## 5.2 Classification

### 5.2.1 Introduction

#### Choix de la distance

Le choix de la mesure de distance entre objets est très important. Malheureusement, trop souvent, il s'agit d'un choix arbitraire, sensible à la représentation des objets, et qui traite tous les attributs de la même manière.

Une solution pour pallier à cette limitation est celle de la prise en compte de la connaissance d'un expert, qui identifiera certains attributs, considérés comme plus pertinents que d'autres pour la tâche de classification, et leur attribuera un poids plus important lors du calcul des distances entre objets.

Mais cette solution devient très difficile à mettre en œuvre lorsque :

- le nombre d'attributs décrivant les données est trop grand ;
- il n'y a pas d'expert humain.

Dans notre travail, nous avons choisi d'utiliser les fonctions de répartition des images dans la base par rapport aux descripteurs, afin de déterminer la pertinence de ces derniers. Par exemple, dans une base qui contient 90 % d'images de grande taille on a de fortes chances, si l'on prend deux images au hasard, de tomber sur deux images grandes. Du coup le descripteur surface n'est (généralement) pas pertinent (il ne le devient que si l'on s'intéresse à la petite population « déviante »).

## 5.2.2 Mesure de similarité

Pour estimer la similitude entre deux images, nous devons évaluer les similarités entre leurs descripteurs correspondant aux mêmes attributs. Il y a deux manières pour combiner ces similarités. La première est une approche parallèle, où une fonction globale de similarité  $D_G$  est calculée comme la somme « pondérée » des similarités :

$$D_G = w_c d_c + w_t d_t + w_f d_f$$

où  $d_i$  et  $w_i$  représentent respectivement la distance normalisée et le poids associé à l'attribut  $i$  ( $c$  pour couleur,  $t$  pour texture, et  $f$  pour forme).

Le poids peut être fixé interactivement par l'utilisateur en accord avec sa requête ou évalué automatiquement par le système quand la base d'images est connue. La distance de Kullback est utilisée pour la couleur [IRB02] tandis qu'il est fait appel à la distance  $L_1$  pour la texture et la forme. La seconde approche est hiérarchique, les attributs sont alors considérés successivement par ordre décroissant d'importance, en évaluant les distances  $d_c$ ,  $d_t$  et  $d_f$ .

## 5.2.3 Similarité multi-vectorielle

Lorsque l'image n'est plus indexée par un seul vecteur mais par un ensemble de vecteurs se rapportant aux différents espaces de caractéristiques (couleur, texture, forme, etc.), alors se pose le problème de la fusion d'informations issues de modèles distincts.

La première possibilité consiste à concaténer les différents descripteurs et à utiliser une des mesures classiques de comparaison entre vecteurs. Les sous-vecteurs sont préalablement centrés et réduits afin de se ramener à une échelle de valeurs commune aux différents attributs. Il est également possible d'adapter la fonction de similarité à la différence d'échelle, comme le permet la distance de Mahalanobis [Mahalanobis30], par une analyse des descripteurs de la base d'images. Selon la dénomination de Rui et Huang [Rui et al.00], le modèle de similarité qui résulte de ce traitement par concaténation est appelé « modèle à plat » (*Flat Model*). Il est par exemple utilisé par le système Mindreader [Ishikawa et al.98].

La seconde possibilité consiste à traiter chaque espace d'attributs indépendamment. La similarité globale est calculée en fusionnant les scores de similarités relatifs à chaque espace de caractéristiques. On parle alors de « modèle hiérarchique » (*Hierarchical Model*) [Rui et al.00].

La fusion des similarités est classiquement réalisée par une simple combinaison linéaire (somme pondérée) [Heinrichs et al.00, Newsam et al.01]. Les coefficients de la somme règlent le poids relatif de chaque attribut dans la similarité globale. Ils peuvent être réglés manuellement ou bien encore automatiquement par des expériences psycho-visuelles ou par interaction avec l'utilisateur.

## 5.2.4 Méthodes de classification

Le problème de la classification en général est de construire une procédure permettant d'associer une classe à un objet. Ce problème se décline en deux variantes : l'approche supervisée et l'approche non-supervisée.

Dans la première, on connaît les classes par avance et on dispose d'un ensemble d'objets déjà classés, servant d'ensemble d'apprentissage. Le problème est alors d'être capable d'associer à tout nouvel objet sa classe la plus adaptée, en se servant des exemples déjà étiquetés.

Dans la seconde (la classification non-supervisée), les classes possibles ne sont pas connues à l'avance et les exemples disponibles ne sont pas étiquetés. Le but est donc de regrouper dans un même cluster (ou groupe) les objets considérés comme similaires, pour constituer les classes (créer, par exemple, différents groupes de textes, à partir d'un ensemble de textes de tous genres, selon la similarité de leur contenu).

Dans ce cas, le problème est de définir cette similarité entre objets. Typiquement, la similarité entre objets est estimée par une fonction calculant la distance entre ces objets. Une fois cette fonction distance définie, la tâche de clustering consiste à réduire au maximum la distance entre membres d'un même cluster, tout en augmentant au maximum la distance entre clusters.

De nombreuses techniques de clustering peuvent alors être envisagées : hiérarchique, par partition, statistique, flou, par réseaux de neurones, par algorithmes génétiques, par recherche Tabou [JMF99].

### Classification non-supervisée

De nouvelles difficultés apparaissent en classification non-supervisée lorsque l'on tente de prendre en compte le fait que certains attributs composant les données ont plus ou moins d'importance dans la construction de certains clusters.

Cette difficulté est depuis longtemps connue en linguistique : il s'agit du problème de la prise en compte du contexte dans les textes.

Typiquement, on représente un texte par un vecteur de mots, et on associe à ceux-ci leur nombre d'occurrences dans le texte. Ainsi, pour établir une distance entre textes, il suffit alors d'évaluer la similarité de ces vecteurs.

L'ambiguïté des mots, problème bien identifié en linguistique, réapparaît alors en classification non-supervisée [ST00].

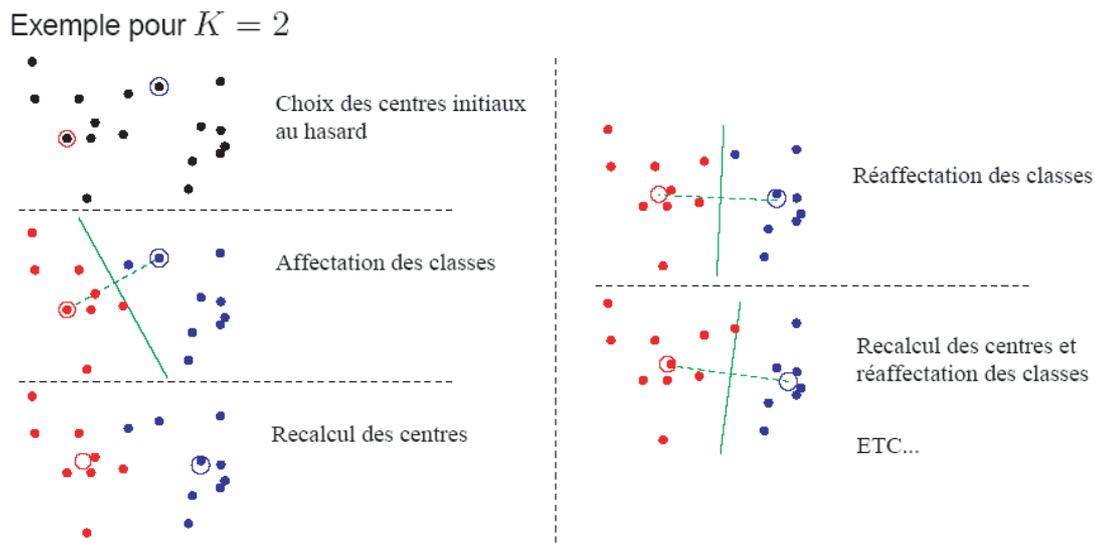


FIG. 5.4 – Algorithme des *k-means*

Ce besoin de prendre en compte le contexte pour juger de la pertinence d'un attribut semble alors naturel, même sur des données quelconques, non textuelles.

Enfin, une dernière problématique se pose en classification non-supervisée, quelle qu'en soit la nature : celle d'avoir à faire à de larges bases de données.

Or, on s'aperçoit que nombreuses sont les solutions qui ne peuvent gérer ce problème, la complexité de l'algorithme envisagé étant trop importante. En effet, typiquement, ces algorithmes calculent les distances entre objets, deux à deux. La complexité de l'algorithme est alors quadratique par rapport au nombre de données en entrée, alors que sur des problèmes concernant de larges bases de données, pour que la complexité soit raisonnable, il faut qu'elle soit linéaire par rapport au nombre de données en entrée, ou tout au plus en  $O(n \cdot \log n)$ .

### Méthode des *k-means*

La méthode des *k-means* initialise  $k$  clusters contenant  $k$  premiers vecteurs quelconques qui servent de centres de gravité initiaux des futurs clusters. Chaque vecteur restant est ensuite ajouté au cluster dont le centre est le plus proche. Les  $k$  clusters sont produits à l'issue de *k-means* de façon à minimiser la fonction d'objectif suivante :

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

où  $x_i$  est un point quelconque du cluster  $C_j$  et  $g_j$  le point moyen de  $C_j$ ,  $j = 1, \dots, k$ .

Les paramètres de l'algorithme des *k-means* sont :

- le nombre  $k$  de clusters, restant largement empirique ;
- la mesure de similarité, qui varie selon le type de descripteur choisi.

L'algorithme se résume aux quatre étapes suivantes :

1. Choisir  $k$  vecteurs aléatoirement pour initialiser les centres des clusters.
2. Affecter chaque vecteur au centre le plus proche. Il en résulte une partition en  $k$  clusters  $C_1, C_2, \dots, C_k$ .
3. Recalculer les centres de chacun de ces clusters. Chaque cluster est représenté par son centre.
4. Les étapes 2 et 3 sont répétées jusqu'à ce que le choix des centres des clusters se stabilise.

Il existe de nombreuses méthodes de type *k-means*, globalement dénommées méthodes des nuées dynamiques. Elles diffèrent des unes des autres suivant le mode de sélection des vecteurs initiaux, le calcul des centres des clusters ou le calcul de la distance entre les vecteurs.

La complexité du temps de calcul de l'algorithme *k-means* est fonction du nombre de distances calculées :  $O(l.k.N)$  où  $l$  est le nombre d'itérations (inconnu *a priori*),  $k$  le nombre de clusters et  $N$  le nombre de vecteurs de la base de données. En principe,  $l$  et  $k$  sont très petits par rapport à  $N$  puisqu'il s'agit d'une grande base. La complexité du *k-means* peut donc être ramenée à  $O(N)$ . Toutefois, dans le cadre d'autres travaux de l'équipe, le paramètre  $k$  pourrait prendre comme valeur  $\sqrt{N}$ , ce qui en fait un algorithme assez lent.

## 5.3 Expérimentation

### 5.3.1 Vue globale du système

La figure 5.5 donne un aperçu du système. Un pré-traitement initial est réalisé, l'objectif est d'obtenir les descripteurs des bas niveau qui caractérisent les images puis de les classer en un ensemble de clusters. L'utilisateur essaiera d'évaluer ses préférences par rapport à ces clusters afin de construire le profil de référence.

La recherche d'images similaires à une image exemple se fait par rapport aux images de la base, notre système tentera d'évaluer les choix de l'utilisateur effectué dans cette recherche pour mettre à jour un profil dit évalué. A la fin de la recherche une distance entre les deux profils est calculée pour chaque itération, le graphe qui représente ces distances nous permettra, s'il y a convergence du profil évalué vers le profil de référence, d'évaluer la vitesse de cette convergence.

### 5.3.2 Environnement expérimental

Les algorithmes d'extraction de descripteurs et de clustering sont implémentés en C++. Nous avons utilisé le système *LightInferImAGE* développé dans l'équipe Atlas-GRIM pour l'indexation et la classification d'images. Tous les tests de notre méthode sont effectués sur PC sous Linux Mandriva 10.2 (Pentium 4 à 3 GHz avec 1 Go de mémoire centrale).

#### Base de test

Le système que nous utilisons compte parmi les systèmes de recherche d'images par le contenu. Son objectif est, d'une part, de bien organiser les images en classes pour faciliter la

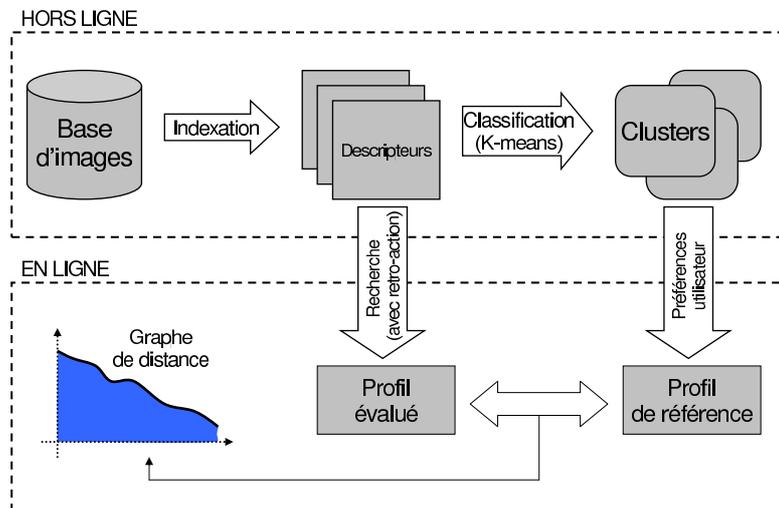


FIG. 5.5 – Vue globale du système

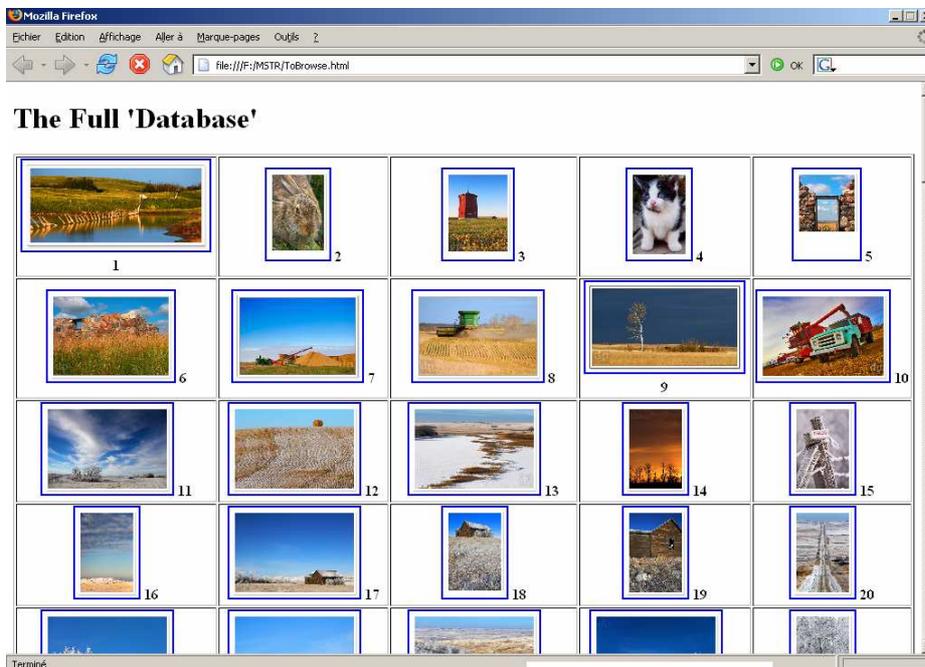


FIG. 5.6 – Base d'images

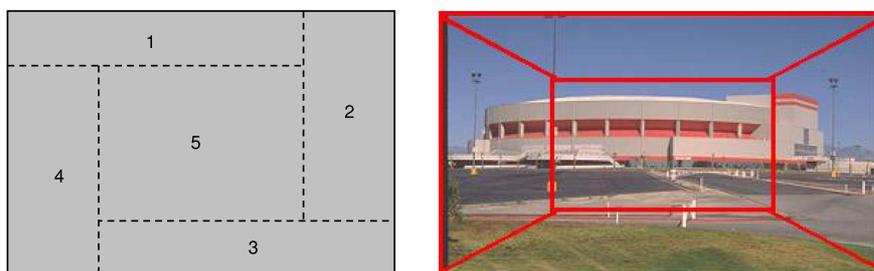


FIG. 5.7 – Découpage de l'image

recherche et, d'autre part, d'améliorer le temps de réponse d'une requête donnée en introduisant la personnalisation. Pour évaluer la performance du système, nous avons essayé de travailler sur une base d'image suffisamment grandes (3 000 images) de tailles et de natures diverses (figure 5.6).

Toutes ces images sont décrites par deux types de descripteurs MPEG-7, le premier de couleur, le deuxième de format.

**Découpage de l'image** L'image est découpée en cinq régions (figure 5.7) pour avoir des descripteurs plus significatifs tel que les histogrammes de couleurs

Donc le nombre de descripteurs dans les expérimentions s'élève à 78 :

- trois de format : aire, orientation et élongation ;
- quinze de couleur qu'il faut multiplier par cinq zones de l'image : un histogramme perceptuel de sept teintes plus les quatre premiers moments d'inertie de l'intensité et de la saturation.

Sous réserve de proposer une distance qui s'intègre bien avec les précédentes, nous aurions pu obtenir jusqu'à 118 dimensions en ajoutant des descripteurs de texture : contraste, corrélation, entropie, uniformité et quatre directions, tout cela pour chaque zone de l'image.

### 5.3.3 Classification en clusters

La classification est faite en utilisant l'algorithme des *k-means* qui prend comme paramètre d'entrée le nombre de clusters de sortie ; dans notre cas  $k = \sqrt{n}$  avec  $n$  le nombre d'images dans la base.

L'algorithme de *k-means* est un algorithme stochastique, c'est pour cette raison qu'on le lance plusieurs fois et, à la fin, on retient le meilleur résultat (local) de classification. Le but est de maximiser la densité inter-cluster et de minimiser les variances intra-clusters. Dans notre base de test, l'algorithme des *k-means* a eu en entrée 3 000 images ce qui engendré 63 clusters.

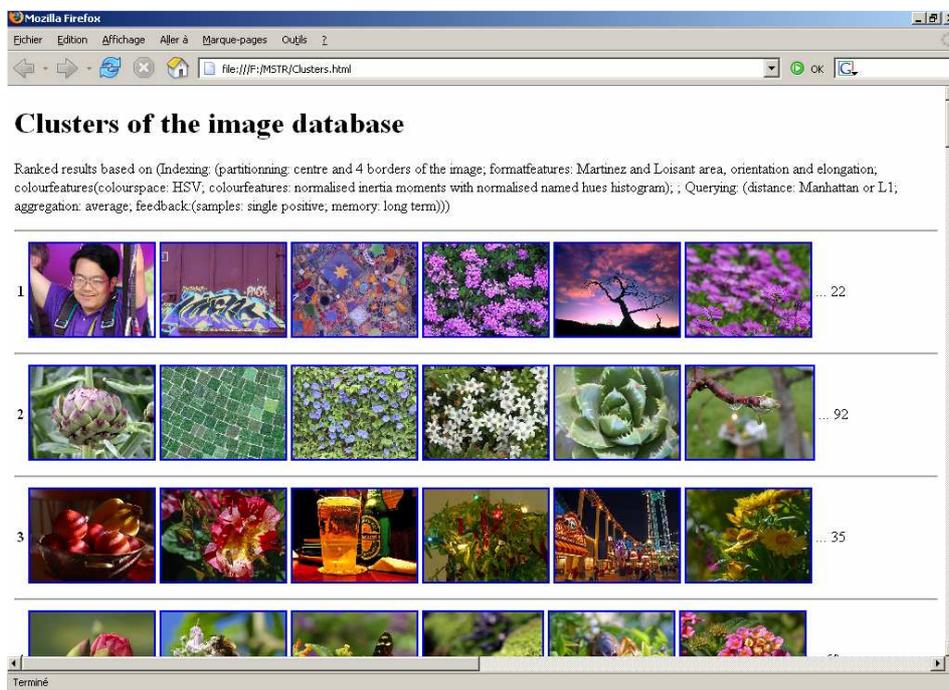


FIG. 5.8 – Les clusters a évaluer par l'utilisateur

```

<list element="PreferenceEntry">
</list>
<PreferenceEntry>
4 N 100 </PreferenceEntry>
</PreferenceEntry>
<PreferenceEntry>
6 N 55 </PreferenceEntry>
</PreferenceEntry>
<PreferenceEntry>
13 N 70 </PreferenceEntry>
</PreferenceEntry>
<PreferenceEntry>
16 N 70 </PreferenceEntry>

```

FIG. 5.9 – Fichier des préférences

### 5.3.4 Personnalisation

La première étape, avant de lancer le processus de recherche, est d'établir une vérité-terrain qui consiste à passer en revue, pour un utilisateur donné, l'ensemble des clusters (figure 4.4<sup>1</sup>) afin que ce dernier puisse évaluer ses préférences par des valeurs entre 0 et 100 %, représentant le pourcentage d'images dans le cluster qu'il apprécie visuellement (figure 5.9).

### 5.3.5 Tests sur l'évolution du profil

**Recherches sans profil de l'utilisateur** L'utilisateur lance sa requête, c'est-à-dire un ensemble d'images choisies à partir de la base ; la recherche s'effectue par rapport au centre de tous les clusters de la base, puis à l'intérieur des clusters eux-mêmes. Enfin, on ordonne les résultats du plus proche cluster au plus éloigné.

**Recherches avec profil de l'utilisateur** Dans ce cas, la requête s'effectue en priorité dans les clusters qui intéressent l'utilisateur, par ordre de préférence décroissant.

**Comment faire ?** Dans notre cas, vu le nombre très important d'images de la base, la recherche peut devenir très pénible, voire impossible à réaliser, de plus, les fichiers journaux sont indisponibles pour notre système.

L'idée est de construire une vérité terrain (Le fichier des préférences, figure 5.9), le programme crée ensuite un profil initiale neutre pour l'utilisateur par rapport à toutes les classes

<sup>1</sup>Le premier cluster contient 22 images à très forte dominance violette ; le deuxième cluster contient 92 images à très forte dominante verte ; le troisième cluster regroupe des images dont les deux teintes principales sont l'orange et le vert avec de larges plages désaturées, c'est-à-dire plus ou moins grises ; etc.

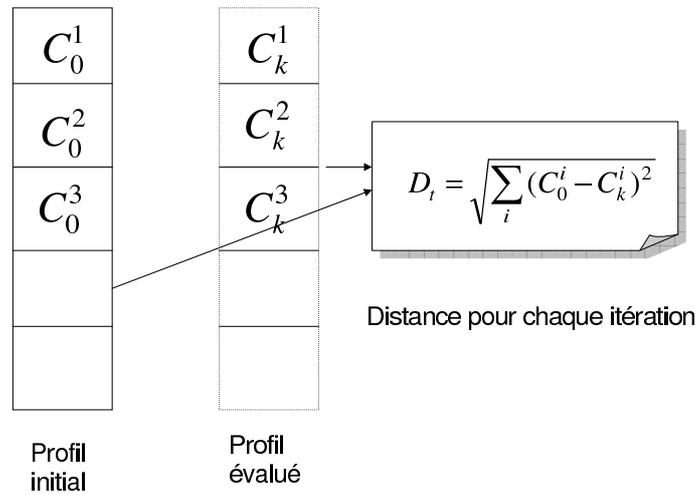


FIG. 5.10 – Calcul des distances

d'images (50% dans le profil évalué). L'évaluation de performance s'effectue de façon à calculer une distances entre le profil initial et le profil courant à chaque itération de la recherche, dans notre cas on a choisit une distance euclidienne (figure 5.10). L'évolution et l'évaluation du profil se fait après chaque requête ; on ajoute un pourcentage de valeur ( $\frac{1}{|C_i|}$ ) à chaque fois que le cluster  $C_i$  est sélectionné.

## 5.4 Résultats

Nous présentons quelques exemples de résultats expérimentaux obtenus à partir de quelques situations (figure 5.11) ; dans la majorité des cas la distance entre les profils évalués et le profil de référence est décroissante. On peut dire en un premier lieu que la connaissance préalable des préférences des utilisateurs peut être reconstruite, mais la vitesse de cette décroissance est assez longue et peut décourager l'utilisateur.

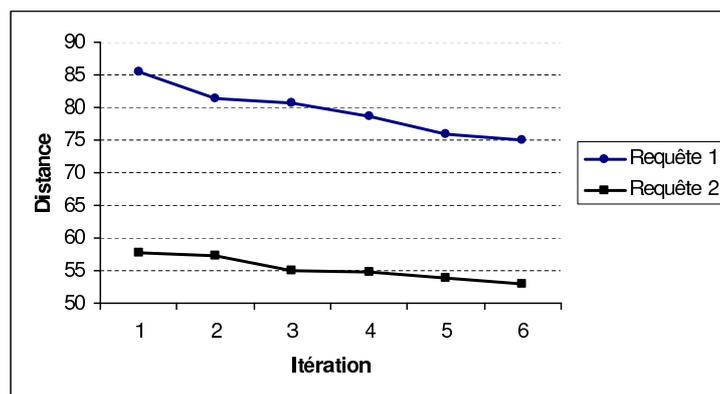


FIG. 5.11 – Évaluation des résultats

# Chapitre 6

## Conclusion et perspectives

À l'issue de ce travail, nous pouvons résumer les étapes parcourues afin de mettre en évidence les apports, puis d'en tirer des conclusions et enfin d'offrir des perspectives de réflexion et développements ultérieurs.

### Résumé du travail effectué

Dans le cadre de ce stage, le travail a été principalement orienté vers la communication entre modules existants : indexation, classification et recherche afin d'y ajouter la gestion de la personnalisation.

L'enchaînement des étapes pour réaliser cela fut le suivant :

- état de l'art sur la personnalisation dans la recherche d'images (par le contenu) ;
- formalisation du problème ;
- prise en main des méthodes d'indexation et de classification d'images existante ;
- implémentation du module de personnalisation et mise en place de la communication entre ce dernier et le système existant ;
- évaluation des performance du nouveau système.

### Conclusion

Durant ce stage, nous avons complété le développement du projet *LightInferImAGE*, permettant ainsi le traitement des préférences des utilisateurs pour l'amélioration de la recherche d'images par l'exemple.

Lors de l'implémentation, nous avons dû faire face à plusieurs difficultés :

- Contraintes matérielles : essentiellement en mémoire centrale ce qui nous a obligé de traiter quelques milliers d'images seulement au lieu de plusieurs milliers, voire un million d'images.

## **Perspectives de recherche**

Afin de compléter nos méthodes, nous souhaitons dans un premier temps continuer à améliorer nos algorithmes. En effet, la suite du développement consiste à faire l'évaluation de la relation entre les profils utilisateur et les clusters à partir des informations déjà existantes dans le système (ex : les fichiers journaux, les mandataires..) en utilisant des méthodes d'extraction de connaissances.

# Bibliographie

- [AMaR00] A.W.M.Smeulders, M.Worring, S.Santini and A.Gupta, and R.Jain. Ieee transactions on pattern analysis and machine intelligence. In *Content based image retrieval at the end of the early years*, volume 22, DECEMBER 2000.
- [Ani06] Koumou Choupo Anicet. *Améliorer la recherche par similarité dans une grande base d'image fixes par des techniques de fouille de données*. PhD thesis, université de Rennes 1, 2006.
- [AS99] A.Pretschner and S.Gauch. Ontology based personalized search. In *In Proceedings of 11th IEEE International Conference On Tools with Artificial Intelligence*, 1999.
- [AT00] L. Ardissono and P. Torasso. Dynamic user modeling in a web store shell. In *Proceedings of the 14th Conference ECAI*, pages 621–625, Berlin,Germany, 2000.
- [BBKK97] S. Berchtold, C. Böhm, D.A. Keim, and H.P. Kriegel. Search in high-dimensional data space : Performance of multimedia databases. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, March 1997.
- [BKK96] S. Berchtold, D. A. Keim, and H.P. Kriegel. The X-tree : An index structure for highdimensional data. In *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB)*, pages 28–39, Mumbai (Bombay), India, September 1996.
- [BP99] D. Billsus and M. J. Pazzani. A personal news agent that talks, learns and explains. In *In Proceedings of the Third International Conference on Autonomous Agents*, pages 268–275, Seattle, 1999.
- [CBP99] J. M. Corridoni, A. Del Bimbo, and P. Pala. Image retrieval by color semantics. *Multimedia Systems*, 7(3) :175–183, May 1999.
- [CJBC01] Jörgensen Corinne, Alejandro Jaimes, Ana B. Benitez, and Shih-Fu Chang. A conceptual framework and empirical research for classifying visual descriptors. *Journal of the American Society for Information Science and Technology(JASIST)*, 52.11 :938–947, 2001.
- [CLY02] Chiny-Yi Chiu, Hsin-Chih Lin, and Shi-Nine Yang. Learning user preference in a personalized cbir. In *International conference on pattern recognition (ICPR)*, 2002.
- [CS98] L. Chen and K. Sycara. Webmate : A personal agent for browsing and searching. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multi Agent Systems*, pages 10–13,, Minneapolis, MN, 1998.

- [CWK05] Y. Chen, J. Z. Wang, and R. Krovetz. Clue : Cluster-based retrieval of images by unsupervised learning. *IEEE Trans. Image Processing*, pages 1187–1201, 2005.
- [DJW05] Ritendra Datta, Jia Li James, and Z. Wang. Content-based image retrieval-approaches and trends of the new age. In *ACM ,Multimedia Information Retrieval (MIR 05)*, Singapore, 2005.
- [GR97] G.Pass and R.Zabih. Ieee comp. soc. conf. comp. vis. and patt. image indexing using color correlograms. In *Histogram refinement for contentbased image retrieval*, pages 762–768, 1997.
- [HBG<sup>+</sup>98] E. H. Han, D. Boley, M. Gini, R.Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Web agent for document categorization and exploration. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 408–415, 1998.
- [HMIH00] K. Hoashi, K. Matsumoto, N. Inoue, and K. Hashimoto. Document filtering method using non-relevant information profile. In *In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 176–183, 2000.
- [IRB02] K. Idrissi, J. Ricard, and A. Baskurt. International conference on image processing. In Rochester, editor, *An objective performance evaluation tool for color based image retrieval systems*, volume 2, pages 389–392, USA, September 2002.
- [J99] T. Jörding. Atemporary user modeling approach for adaptive shopping on the web. In *Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW*, pages 75–79, Canada, 1999.
- [JMF99] A. Jain, M. Murty, and P. Flynn. Acm computing surveys. In *Data clustering : a review*, 1999.
- [KJ00] T. Kamdar and A. Joshi. On creating adaptive web sites usingweblog mining. Technical report, Technical Report TR-CS-00-05.Department of Computer Science and Electrica Engineering Universityof Maryland, Baltimore, 2000.
- [KS97] Norio Katayama and Shinichi Satoh. The SR-tree : an index structure for high-dimensional nearest neighbor queries. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 369–380, Tucson, Arizona, May 1997.
- [LJF94] King Ip Lin, H. V. Jagadish, and Christos Faloutsos. The TV-tree : An index structure for high-dimensional data. *VLDB Journal*, 3(4) :517–542, October 1994.
- [LYZ01] Qing Li, Jun Yang, and Yueting Zhuang. Web-based multimedia retrieval : Balancing out between common knowledge and personalized views. In *Proceedings of 2nd International Conference on Web Information System Engineering*, pages 92–101, Kyoto, Japan, 2001.
- [MCS00] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on webusage mining. *Communications of the ACM*, 43(3) :142–141, 2000.

- [MKH<sup>+</sup>00] Z. Mojsilovic, J. Kovacevic, J. Hu, R. J. Safranek, and S. K. Ganathy. Matching and retrieval based on vocabulary and grammar of color patterns. *IEEE Transaction on Image Processing*, 9 (1) :38–55, 2000.
- [Mla99] D. Mladenic. Machine learning used by personal webwatcher. In *In Proceedings of ACAI-99 Workshop on Machine Learning and Intelligent Agents*, 1999.
- [MSS02] B. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7 : Multimedia Content Description Interface*. Wiley, 2002.
- [NW97] D.S.W. Ngu and X. Wu. Sitehelper :a localized agent that helps incremental exploration of the world wide web. *Computer Networks and ISDN Systems :The International Journal of Computer and Telecommunications Networking*, 29(8) :1249–1255, 1997.
- [RHOM98] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback : A power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 644–655, September 1998. Special Issue on Segmentation,Description, and Retrieval of Video Content.
- [SB91] M.J. Swain and D.H Ballard. *Color indexing*, volume 7 of 1. International Journal of Computer Vision., November 1991.
- [SB02] B. Le Saux and N. Boujemaa. Unsupervised robust clustering for image database categorization. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2002.
- [SCDT00] J. Srivastava, R. Cooley, M. Deshpande, and P. T. Tan. Web usage mining :discovery and applications of usage patterns from web data. *SIGKDD Explorations*, pages 12–23, 2000.
- [Sch01] E. Schwarzkopf. An adaptive web site. In *Proceedings of the UM2001Workshop on Machine Learning for User Modeling*, pages 77–86, 2001.
- [SD96] M. Stricker and A. Dima. *Color indexing with weak spatial constraints*. Storage and Retrieval for Image and Video Databases, 1996.
- [SGJ01] S. Santini, A. Gupta, and R. Jain. Emergent semantics through interaction in image databases. *IEEE Transactions on Knowledge and Data Engineering*, 13 :337–351, 2001.
- [SPF99] M. Spiliopoulou, C. Pohle, and L.C. Faulstich. Improving the effectiveness of a web site with web usage mining. In *Proceedings of the 1999 Knowledge Discovery and Data Mining Workshop (KDD) on Web Mining*, San Diego CA, 1999.
- [ST00] N. Slonim and N. Tishby. Research and development in information retrieval. In *Document clustering using word clusters via the information bottleneck method*, 2000.
- [SYUK00] Asushi Sakurai, Masatoshi Yoshikawa, Shunsuke Uemura, and Haruhiko Kojima. The A-tree : An index structure for high-dimensional spaces using relative approximation. In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB)*, pages 516–526, Cairo, Egypt, September 2000.

- [WIY99] D. H. Widyantoro, T. R. Ioerger, and J. Yen. An adaptive algorithm for learning changes in user interests. In *Proceedings of the Eighth International Conference on Information Knowledge Management*, pages 405–412, 1999.
- [WS02] M. Wallace and G. Stamou. Towards a context aware mining of user interests for consumption of multimedia documents. *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, August 2002.
- [YC98] G. Proietti, Y. Gong, and C. Faloutsos. Ieee computer society conference on computer vision and pattern recognition. In *Image Indexing and Retrieval Based on Human Perceptual Color Clustering*, ISBN :0-8186-8497-6, page 578, 1998.
- [YJGMD96] T.W. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. *WWW5/Computer Networks*, 28 (7-11) :1007–1014, 1996.
- [YLZ02] Jun Yang, Qing Li, and Yueting Zhuang. Image retrieval and relevance feedback using peer indexing. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 2002.
- [YLZ03] Jun Yang, Qing Li, and Yueting Zhuang. Modeling data and user characteristics by peer indexing in content-based image retrieval. In *Proceedings of the 9th International Conference on Multimedia Modeling*, Taiwan, 2003.
- [ZCH<sup>+</sup>04] X. Zheng, D. Cai, X. He, W. Y. Ma, and X. Lin. Locality preserving clustering for image database. In *Proceedings of the ACM International Conference on Multimedia*, 2004.
- [ZYL02] Yueting Zhuang, Jun Yang, and Qing Li. A graphic-theoretic model for incremental relevance feedback in image retrieval. In *Proceedings of the International Conference on Image Processing (ICIP)*, New York, September 2002.

# Table des figures

|      |  |    |
|------|--|----|
| 2.1  | Schéma général et objectifs . . . . .  | 13 |
| 3.1  | Processus de reconnaissance du comportement d'un groupe d'utilisateurs . . . . .         | 18 |
| 3.2  | Modélisation du profil commun et le profil utilisateur . . . . .                         | 20 |
| 3.3  | Modélisation des corrélations sémantiques entre les images . . . . .                     | 23 |
| 3.4  | Index des images par d'autres images . . . . .   | 23 |
| 4.1  | Exploitation de la couleur . . . . .   | 25 |
| 4.2  | Exploitation de la texture . . . . .   | 26 |
| 4.3  | Différences de perception . . . . .  | 26 |
| 4.4  | Définition des clusters en intension (haut du schéma) et en extension (en bas) . . . . . | 28 |
| 4.5  | Algorithme général . . . . .   | 30 |
| 4.6  | Fonctionnement . . . . .   | 31 |
| 5.1  | Espaces de couleurs . . . . .  | 33 |
| 5.2  | Histogramme de couleur -1 . . . . .  | 34 |
| 5.3  | Histogramme de couleur -2 . . . . .  | 35 |
| 5.4  | Algorithme des <i>k-means</i> . . . . .  | 39 |
| 5.5  | Vue globale du système . . . . .   | 41 |
| 5.6  | Base d'images . . . . .  | 41 |
| 5.7  | Découpage de l'image . . . . .   | 42 |
| 5.8  | Les clusters a évaluer par l'utilisateur . . . . .                                       | 43 |
| 5.9  | Fichier des préférences . . . . .  | 44 |
| 5.10 | Calcul des distances . . . . .   | 45 |
| 5.11 | Évaluation des résultats . . . . .   | 46 |

# Liste des tableaux

|     |  |    |
|-----|--|----|
| 3.1 | Profil d'un utilisateur pour la recherche textuelle . . . . .                | 16 |
| 3.2 | Profil d'un groupe d'utilisateurs pour la recherche textuelle . . . . .      | 19 |
| 3.3 | Proposition de termes linguistiques pour les propriétés de texture . . . . . | 22 |