Analyse multirésolution pour la recherche et l'indexation d'images par le contenu dans les bases de données images - Application à la base d'images paléontologique Trans'Tyfipal

Jérôme Landre

UNIVERSITÉ DE BOURGOGNE
U.F.R. SCIENCES ET TECHNIQUES
ÉCOLE DOCTORALE BUFFON
IMAGES ET MODÉLISATION DES OBJETS NATURELS

THÈSE

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ DE BOURGOGNE

discipline : **Instrumentation et Informatique de l'Image**

ANALYSE MULTIRÉSOLUTION POUR LA RECHERCHE ET L'INDEXATION D'IMAGES PAR LE CONTENU DANS LES BASES DE DONNÉES IMAGES - APPLICATION À LA BASE D'IMAGES PALÉONTOLOGIQUE TRANS'TYFIPAL

Présentée par

Jérôme LANDRÉ

Soutenue le 7 décembre 2005

JURY

Djamal BENSLIMANE	Professeur, Université de Lyon 1	Rapporteur
Sophie MONTUIRE	Maître de conférences, École Pratique des Hautes Études	Examinatrice
Jack-Gérard POSTAIRE	Professeur, Université de Lille 1	Rapporteur
Frédéric TRUCHETET	Professeur, Université de Bourgogne	Directeur
Kokou YETONGNON	Professeur, Université de Bourgogne	Président

«Si on supprime l'image, c'est l'univers entier qui disparaît.»
Joseph (Nicéphore) NIÉPCE (1765-1833)

À la Vie, long chemin à tracer,
à Nathalie, à Aubin, à Noé, . . .

Ces travaux de recherche ont été menés dans le cadre de l'école doctorale Buffon (Images et modélisation des objets naturels), au sein du laboratoire Électronique, Informatique et Images (Le2i) de l'université de Bourgogne sur le site de l'Institut Universitaire de Technologie du Creusot.

Université de Bourgogne
École doctorale Buffon - Images et modélisation des objets naturels
Institut Universitaire de Technologie
Laboratoire Électronique, Informatique et Images
U.M.R. C.N.R.S. 5158
12, rue de la Fonderie
71 200 Le Creusot
FRANCE
tél. +33 (0)3–85–73–10–00
fax +33 (0)3–85–73–10–99
http ://www.u-bourgogne.fr
http ://iutlecreusot.u-bourgogne.fr
http ://vision.u-bourgogne.fr/le2i

Table des matières

Remerciements

Entreprendre une thèse n'est pas une chose facile, c'est se lancer sans garantie de succès sur un chemin scientifique inconnu semé d'embûches. J'ai débuté ce travail de recherche alors que j'étais ingénieur d'études à l'I.U.T. du Creusot sous l'impulsion de nombreuses personnes qui m'ont encouragées. Je tiens à remercier M. Frédéric TRUCHETET alors directeur du Le2i et M. Patrick GORRIA alors responsable de l'équipe de recherche du site du Creusot qui m'ont permis de commencer ce travail. Je remercie aussi M. Fabrice MÉRIAUDEAU responsable actuel de l'équipe du Creusot ainsi que M. Michel PAINDAVOINE, directeur du laboratoire Le2i qui ont assuré la continuité de la thèse.

Mes remerciements s'adressent également à M. Jean-Luc GISCLON directeur de l'I.U.T. du Creusot et M. Pierre SUZEAU alors chef du département Génie Électrique et Informatique Industrielle pour m'avoir permis d'aménager mes horaires de travail lors de mon D.E.A. qui fut l'origine de mes travaux de recherche puis lors des deux premières années de thèse.

Lorsque j'ai débuté cette thèse, je ne me rendais pas compte de l'étendue du travail à accomplir. Il y a eu des moments heureux où le travail avançait bien, où il était reconnu par la communauté scientifique sous la forme de publications et où les difficultés étaient faciles à surmonter. Et il y a eu tous les moments de doute, de remise en cause de la méthode, les difficultés de programmation, de pertinence et d'interprétation des résultats. Et pour tous ces moments pénibles, je tiens à remercier de tout cœur les personnes qui m'ont toujours redonné le moral et la confiance si indispensables dans les périodes difficiles.

Une thèse est un travail assez personnel qui s'inscrit toutefois dans une équipe. Je ne pouvais rêver mieux que l'équipe du laboratoire Le2i dans laquelle l'ambiance reflète l'état d'esprit qui y règne. Je remercie donc tous mes collègues doctorants, tous les membres temporaires ou permanents que j'ai croisé au cours de cette expérience (ils sont trop nombreux pour pouvoir les citer tous).

Je n'oublie pas tous les membres du personnel de l'I.U.T. du Creusot qui sont mes collègues de travail et chez qui j'ai toujours senti un soutien moral infaillible pour la réussite de cette entreprise.

Je remercie mon directeur de thèse et encadrant Frédéric TRUCHETET pour m'avoir encadré, m'avoir toujours fait confiance, et avoir guidé la conduite de la thèse aux cours de ces années. Je

le remercie pour sa patience et sa confiance qui ont été durement éprouvées pendant ces longues années.

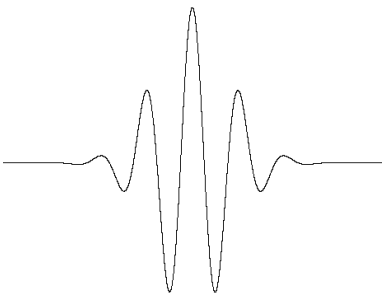
Je tiens à adresser mes remerciements à Philippe VAN HAM, Professeur à l'université libre de Bruxelles pour son accueil à Bruxelles et nos divers échanges, à Jack-Gérard POSTAIRE pour son accueil à Lille et la discussion au sujet de ma méthode de classification ainsi qu'à M. Pierre PERRUCHET, professeur de psychologie à l'université de Bourgogne pour ses conseils concernant le mode opératoire de la phase des tests psycho-visuels avec les utilisateurs du système. Merci aussi à Marie-Noëlle TERRASSE, Éric LECLERCQ, Laurent BESSON et Arnaud DA COSTA pour les échanges d'informations intra-Le2i au sujet de l'indexation d'images.

J'exprime ma gratitude à M. Djamal BENSLIMANE, Professeur de l'université de Lyon 1 ainsi qu'à M. Jack-Gérard POSTAIRE, Professeur à l'université de Lille 1 pour avoir accepté de juger ces travaux en tant que rapporteurs.

Je tiens à exprimer ma reconnaissance à M. Kokou YETONGNON, Professeur de l'université de Bourgogne, pour avoir accepté de présider ce jury. Merci à Mme Sophie MONTUIRE, maître de conférences de l'École Pratique des Hautes Études, habilitée à diriger les recherches qui a accepté de faire partie de ce jury.

Je tiens à rendre hommage aux logiciels libres et à leurs créateurs qui ont permis à chacun de disposer d'outils informatiques gratuits et performants. Merci GNU Linux, LaTeX, the Gimp, PHP, Apache, MySQL et tous les autres...

Enfin — *last but not least* — j'exprime toute ma gratitude à Nathalie, à Aubin et à Noé qui ont supporté avec moi les bons et les mauvais moments et qui m'ont donné la force d'aller au bout de cette aventure.



Résumé

ANALYSE MULTIRÉSOLUTION POUR LA RECHERCHE ET L'INDEXATION D'IMAGES PAR LE CONTENU DANS LES BASES DE DONNÉES IMAGES - APPLICATION À LA BASE D'IMAGES PALÉONTOLOGIQUE TRANS'TYFIPAL

Les systèmes de recherche d'images par le contenu récents utilisent une approche de navigation visuelle interactive dans les bases d'images. Ces méthodes utilisent une classification des images (hors-ligne) dans un arbre de recherche pour une navigation visuelle (en ligne) des utilisateurs. Cette approche possède trois problèmes principaux : 1) La taille du vecteur descripteur (n>100) rend les calculs de distance sensibles à la malédiction de la dimension, 2) La présence d'attributs de nature différente dans le vecteur descripteur ne facilite pas la classification, 3) La classification ne s'adapte pas (en général) au contexte de recherche de l'utilisateur.

Nous proposons dans ce travail une méthode basée sur la construction de hiérarchies de signatures de tailles réduites croissantes qui permettent de prendre en compte le contexte de recherche de l'utilisateur. Notre méthode tend à imiter le comportement de la vision humaine.

Le vecteur descripteur contient des attributs issus de l'analyse multirésolution des images. Ces attributs sont organisés par un expert du domaine de la base d'images en plusieurs hiérarchies de quatre vecteur signature de taille réduite croissante (respectivement 4, 6, 8 et 10 attributs). Ces signatures sont utilisées pour construire un arbre de recherche flou grâce à l'algorithme des nuées dynamiques (dont deux améliorations sont proposées). Les utilisateurs en ligne choisissent une hiérarchie de signature parmi celles proposées par l'expert en fonction de leur contexte de recherche.

Un logiciel de démonstration a été développé. Il utilise une interface web dynamique (PHP), les traitements d'images (optimisés) sont réalisés grâce aux librairies Intel IPP et OpenCV, le stockage et l'indexation sont réalisés par une base de données MySQL, une interface de visualisation 3D (Java3D) permet de se rendre compte de la répartition des images dans la classification.

Un protocole de tests psycho-visuels a été réalisé. Les résultats sur la base paléontologique Trans'Tyfipal sont présentés et offrent des réponses pertinentes selon le contexte de recherche. La méthode donne de bons résultats, tant en temps de calcul qu'en pertinence des images résultats lors de la navigation dans les bases d'images homogènes.

Mots-clés : base d'images, indexation par le contenu, navigation psycho-visuelle, analyse multirésolution, classification, organisation hiérarchique, arbre de recherche flou.

Abstract

MULTIRESOLUTION ANALYSIS FOR CONTENT-BASED IMAGE INDEXING AND RETRIEVAL IN IMAGES DATABASES - APPLICATION TO THE PALEONTOLOGICAL IMAGES DATABASE TRANS'TYFIPAL

Recent content-based image retrieval systems offer an interactive visual browsing of images databases. These methods perform a classification of images (offline) into a search tree for users browsing (online). This approach shows three main problems : 1) The size of decriptor vector (n>100) makes distance computing sensitive to dimensionality curse, 2) Having many different kinds of attributes into descriptor vector does not help classification, 3) In general, classification does not take in consideration users' search context.

In this work, we propose a method based on building hierarchical signatures having small increasing sizes, this allows to take users' search context into consideration. Our method tries to reproduce human vision behavior.

Descriptor vector contains attributes coming from multiresolution analysis of images. These attributes are organized by an expert of the images domain into several hierarchies made of four signature vectors of small but increasing sizes (4, 6, 8 and 10 attributes). These signatures are used to build a fuzzy research tree with k-means classification algorithm (two improvements of this algorithm are given). Online users choose a hierarchy of signature between those built by expert following their search context.

A demonstration software has been developed. It uses a dynamic web interface (PHP), optimized image processing tasks using Intel IPP and OpenCV libraries, a MySQL relational database for storage and indexation, a Java3D interface to see images classification results.

A testing psycho-visual protocol has been proposed. Results on Trans'Tyfipal paleontological images database are presented and offer good answers to users' queries. Our method gives good results in response time and accuracy during visual browsing.

Keywords : images database, content-based image indexing and retrieval, psycho-visual browsing, multiresolution analysis, classification, hierarchical organisation, fuzzy search tree.

Première partie

Introduction

Chapitre 1

Des images...

Dans la société du $21^{ème}$ siècle, l'image est omniprésente, la télévision, les magazines, la presse en général, Internet sont autant de moyens de communication où l'on trouve des images. Il serait à l'heure actuelle difficile de se passer d'images et il est difficile à croire que l'image moderne n'est âgée que d'à peine 200 ans. Ce chapitre présente la problématique de la recherche d'images en traçant l'historique de l'évolution de l'utilisation des images dans notre vie quotidienne.

1.1 La notion d'image

Avant d'entrer dans le sujet proprement dit, il est important de comprendre la nature des objets que nous allons manipuler. Intéressons-nous à la notion d'image, qu'est-ce qu'une image ? La définition du dictionnaire Larousse donne : «**image** : n.f. (lat. imago) [...] Représentation imprimée d'un sujet quelconque [...]».

Une image est une représentation imprimée d'un sujet quelconque. Cela signifie qu'une image nécessite un support sur lequel elle sera imprimée. Ainsi une photographie papier, une peinture sont des exemples d'images au même titre qu'une image numérique affichée sur un écran d'ordinateur.

Informatiquement, une image sera une représentation numérique en mémoire d'un sujet imprimé sur une rétine artificielle (matricielle comme le capteur d'un appareil photographique numérique ou la scène virtuelle d'une image de synthèse ou bien linéaire comme le capteur optique du télécopieur, du photocopieur ou du scanner). Nous allons donc travailler sur des

ensembles de nombres numériques codés sur un ordinateur.

Les premières images sont nées il y a de nombreuses années sous une forme primitive, certes, mais déjà très fidèle à la réalité du monde.

1.2 Trente mille ans d'images

Les images imprimées les plus vieilles (fig. 1.1) sont agées de plus de trente mille ans. En ces temps reculés, l'Homo Sapiens Sapiens — dont nous sommes les représentants actuels — a appris à confectionner des peintures d'excellente qualité et à peindre les murs des grottes qu'il habitait. La technique du dessin, l'utilisation de différentes couleurs et la fidélité des détails des animaux sont surprenantes et posent de nombreuses questions aux spécialistes de la préhistoire au sujet des peintures et de l'art pariétal. Il y a trente mille ans, l'homme utilisait déjà des images...

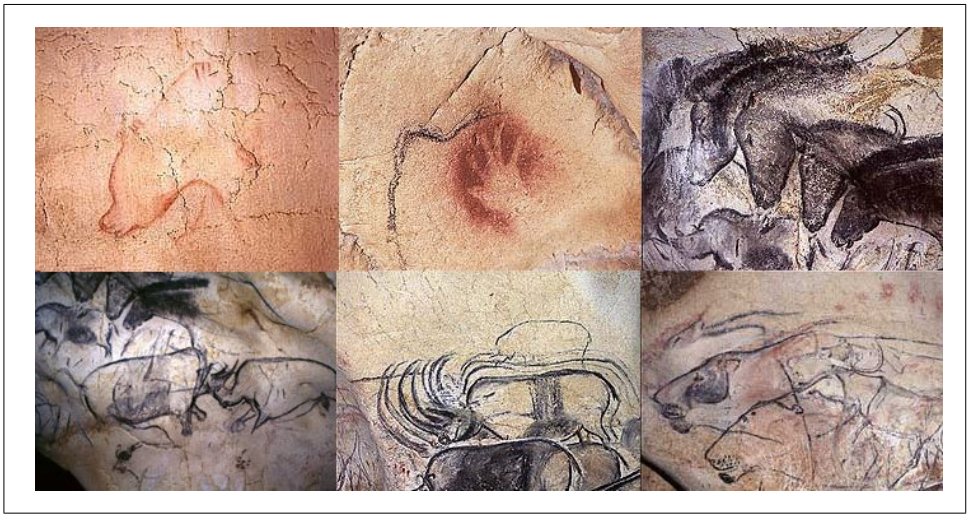


FIG. 1.1 – Les peintures rupestres des grottes Chauvet-Pont-d'Arc [9] (Ardèche), agées de plus de trente mille ans sont les plus vieilles images connues à ce jour.

Ainsi, bien avant l'invention de l'écriture, les hommes avaient choisi comme moyen de communication les images. Pourquoi ?

Dessiner, c'est reproduire sur un support une image stockée ou créée par le cerveau humain. C'est une action simple qui ne nécessite aucune connaissance particulière sinon celle de tracer des traits. La magie de ces images c'est que trente mille ans après leur création, il est très facile à une personne de reconnaître les animaux qui y figurent sans être un spécialiste mondial de l'ère aurignacienne. On reconnaît au premier coup d'oeil un ours, une main, des chevaux, des rhinocéros, des lions. Les images apportent des informations sur les animaux que les hommes

de l'époque chassaient, mangeaient ou redoutaient.

L'image est donc porteuse de deux types d'informations :

– d'une part par **son contenu** : le dessin en lui-même apporte des informations importantes sur la taille de l'image, sur l'utilisation de la couleur ou non et sur la technique de tracé utilisée,
– d'autre part par **son interprétation** : Les dessins représentent les animaux que les hommes préhistoriques connaissaient. On ne peut qu'émettre des hypothèses quant à leur signification exacte : un paléontologue y verra la mémoire d'un clan, un artiste la naissance de la peinture, un théologien la naissance des croyances... Ainsi, chacun perçoit les images selon sa propre vision du monde.

Le premier type d'information est appelé niveau **syntaxique** (ou graphique) qui nous donne des renseignements sur la scène que représente l'image. Le second se nomme niveau **sémantique**, c'est la phase d'interprétation de l'image qui varie d'une personne à une autre en fonction des connaissances de chacun et du contexte d'observation.

Il faut prendre en compte dans la sémantique le contexte social et sociétal de l'observateur de l'image. Une image n'a pas la même signification selon la société de l'observateur. Les différentes cultures et les modes de vie peuvent influencer l'interprétation d'une scène.

L'aspect syntaxique et l'aspect sémantique sont importants pour la recherche d'images et nous verrons plus tard qu'il est très difficile de décoder le niveau syntaxique pour arriver au niveau sémantique (c'est pourtant le plus important problème à résoudre).

L'image est depuis plusieurs millénaires une source très instructive pour les historiens. Les peintures rupestres, les décorations des tombes égyptiennes, les fresques romaines, les céramiques grecques, les manuscrits enluminés du Moyen-Âge, les peintures de la renaissance, les toiles représentant des évènements importants ou bien des scènes de la vie quotidienne sont autant de sources de connaissances de la vie à une certaine époque. Les images existent depuis la nuit des temps et leur nombre croît exponentiellement.

Mais sans la vision, sans ce formidable outil optique qu'est l'œil, il n'y aurait pas d'images.

La vision est le plus important de nos cinq sens, c'est celui qui nous permet de percevoir notre environnement et d'interagir avec lui. La vision coordonne notre attention, nos mouvements, nos réactions, elle oriente nos décisions. Elle permet de différencier les couleurs, les formes, les textures, les visages, les objets, les scènes. Elle nous donne beaucoup plus d'information que n'importe quel autre moyen de description. Le proverbe «Un bon dessin vaut mieux qu'un long discours.» est plus que jamais d'actualité.

La vision est un sens inné. On n'apprend pas à voir comme on apprend à parler, à lire, à écrire, on voit. Un bébé met quelques mois pour apprendre à parler, à marcher, il voit et il

regarde son environnement dès sa naissance. Par contre, il devra apprendre à interpréter ce qu'il voit. Si plusieurs personnes observent une image, tous perçoivent la même image mais chacun l'interprète à sa façon, selon ses connaissances, son passé, son contexte social et sociétal et sa propre vision du monde.

Depuis une dizaine d'années, la vision est au cœur de l'informatique. Les interfaces graphiques ont permis de simplifier l'utilisation des ordinateurs. Aujourd'hui, les ordinateurs disposent d'une interface simple et conviviale ou l'aspect visuel est très important. Il n'y a plus besoin de connaître les commandes du système d'exploitation, les actions sont effectuées simplement à l'aide de la souris. Les interfaces visuelles et la baisse des prix sont à l'origine de l'augmentation des ventes d'ordinateurs dans le monde.

Les langages de programmation, les bases de données deviennent eux aussi visuels. Les objets informatiques manipulés sont représentés par des icônes et on peut voir les liens entre ces objets dans le programme ou la base de données. Grâce à de nombreux logiciels, on peut concevoir une application rapidement sans connaître de langage de programmation (Labview [16] pour ne citer que lui est un excellent exemple de programmation visuelle).

Ce sont toutes ces raisons qui font que l'image a toujours joué, joue encore et jouera toujours un rôle essentiel dans notre vie quotidienne.

1.3 Les images aujourd'hui

Depuis l'invention de la photographie par Joseph (Nicéphore) Niépce [19] en 1822 (en Bourgogne), le nombre d'images photographiques produites par l'homme n'a jamais cessé de croître. Le cinématographe des frères Auguste et Louis Lumière [11] (à Lyon) a apporté en 1895 une dynamique à l'image au prix d'une multiplication du nombre d'images (25 par seconde suffisent, le cerveau s'occupe du reste). Depuis ces deux inventions majeures, le nombre d'images produites dans le monde n'a cessé d'augmenter.

Avec les technologies de l'électronique et de l'informatique, les images numériques se multiplient à une vitesse vertigineuse. Les estimations prévoient que la création d'images numériques prendra un essor tel que la quantité d'images produites dans les quelques années à venir sera plus importante que la quantité d'images créées depuis le début de l'humanité. Ce phénomène est dû principalement à l'essor de l'informatique domestique (ordinateurs personnels, caméscopes, scanners, appareils photographiques numériques, téléphones portables) et à la généralisation du réseau Internet.

Toutefois le média le plus générateur d'images reste la télévision. Avec la multiplication des chaînes, l'Institut National de l'Audiovisuel [12] qui a pour mission d'être la mémoire de la diffusion d'images télévisées (actuellement 1,5 millions d'heures de programmes archivées) doit archiver des milliers d'heures de programmes chaque année, ce qui représente un volume

numérique gigantesque. Et dans les années à venir, chacun aura accès à encore plus d'images avec le développement de la Télévision Numérique Terrestre (TNT).

La conséquence directe de ce phénomène est que la quantité d'images stockées à travers le monde est sans cesse en expansion. Il y a seulement quelques années, le problème principal était le stockage de tels volumes d'information. Grâce à la technologie, aux capacités sans cesse croissantes des disques durs des ordinateurs, aux technologies de compression de données, ce problème est maintenant pratiquement résolu.

Aujourd'hui, le problème principal posé par cette expansion très rapide est qu'il existe peu de systèmes de recherche d'images suffisamment performants pour permettre de retrouver une certaine image dans une base de quelques milliers (voire millions) d'images. Par exemple, les utilisateurs d'Internet ne disposent pas (ou très peu) d'outil permettant de retrouver rapidement une image dont ils ont besoin. C'est pour cette raison que de nombreux chercheurs à travers le monde s'intéressent de près à la recherche d'images. C'est un problème ouvert qui met à l'épreuve des thématiques de recherche diverses :

- Les **bases de données** pour le stockage et l'indexation des images,
- Le **traitement d'images** pour l'extraction de paramètres discriminants des images,
- Les **statistiques**, les **distances métriques** pour l'évaluation de distances entre images,
- La **classification** pour le classement des images les unes par rapport aux autres,
- L'aspect **réseau** pour la gestion de la base de données à distance avec tous les problèmes de sécurité et de transmission de l'information qui en découlent.

De plus en plus de laboratoires des disciplines citées ci-dessus s'intéressent au problème de la recherche d'images. Les chercheurs français ne sont pas en retard, bien au contraire, des sessions spéciales recherche d'images sont organisées dans de nombreux colloques nationaux. Le GDR ISIS [10] a organisé en 2002-2003 une action spécifique (A.S.) «fouille d'images» qui a permis à tous les acteurs nationaux du domaine d'assister à des présentations très intéressantes et d'échanger des idées sur le sujet. J'ai participé à cette A.S. en assistant à de nombreuses réunions et en présentant le principe de ce travail de recherche devant les collègues de toute la France [LT03a].

J'ai choisi ce sujet de recherche avec Frédéric Truchetet parce qu'il était et reste un problème ouvert et difficile et parce qu'il est adapté à mes compétences. Ma formation initiale en bases de données et intelligence artificielle puis en traitement du signal et des images ainsi que mon expérience professionnelle en réseaux [Lan04, Lan05a, La05] informatiques m'ont beaucoup servi lors de ce travail de recherche et m'ont permis de mieux comprendre et appréhender l'indexation et la recherche d'images par le contenu.

1.4 La recherche d'information

De nos jours, la recherche d'information est primordiale dans tous les secteurs d'activités : dans le monde industriel, juridique, médical, scientifique, économique et bien sûr informatique... La plupart des questions que nous pouvons nous poser sont déjà connues et résolues. Il faut donc trouver l'information où elle se trouve rapidement pour ne pas perdre de temps. Grâce à l'Internet, on trouve de plus en plus d'information de tous types, il convient de faire le tri et de posséder des outils de recherche performants et d'utilisation facile.

De nombreuses techniques de recherche d'information ont vu le jour ces dernières années dont la fouille de données (*Data Mining*). La fouille de données consiste à analyser, par des méthodes statistiques et mathématiques, un large volume de données, afin d'en faire ressortir des tendances ou des règles. Il correspond à l'ensemble des techniques et des méthodes qui, à partir de données, permettent d'obtenir des connaissances exploitables.

Les techniques de recherche de textes sont très efficaces car elle bénéficient de travaux de recherche éprouvés depuis une dizaine d'années [31]. Ainsi, les moteurs de recherche basés sur le texte sont de plus en plus performants. Malheureusement, la recherche d'images pose encore de nombreux problèmes en raison de la difficulté à mettre en œuvre des algorithmes efficaces d'analyse de scène dans une image.

1.5 La recherche d'images

Le but de la recherche d'images est de retrouver une (ou plusieurs) image(s) parmi une base de quelques milliers (voire millions) d'images pour répondre à une requête d'un utilisateur. Comme on le verra dans le chapitre consacré aux bases de données images, plusieurs moyens sont proposés à l'utilisateur pour formuler sa requête.

La recherche sur les bases d'images nécessite de posséder une base d'images sur laquelle on puisse tester les méthodes mises en œuvre. Certaines bases d'images disponibles gratuitement existent sur Internet [5, 1, 25], chacune contenant quelques milliers d'images. D'autres bases d'images contenant plusieurs millions d'images existent également mais sont commerciales [3, 6, 4]. Enfin, la plus grande source d'images au monde c'est l'Internet. Les moteurs de recherche comme google [22] contiennent les adresses de millions d'images, d'où l'intérêt commercial de la recherche d'images.

Le choix de la base de données est très important pour l'étude d'un système de recherche d'images par le contenu. En effet, le contenu des images va influencer très fortement les choix de la méthode de recherche. Ainsi une méthode qui marche sur une base donnée peut donner des résultats très différents sur une autre base. Dans la suite de cette thèse, le champ d'application de notre méthode sera défini plus précisémment.

Pour tester notre méthode, nous avons utilisé la base d'images paléontologique Trans'TyFiPal avec l'aimable autorisation de l'équipe du laboratoire de biogéosciences de l'université de Bourgogne ainsi qu'une base gratuite téléchargée sur Internet. Ces banques d'images sont décrites ci-dessous.

1.6 Les bases d'images utilisées

Cette section présente les diverses bases d'images utilisées pour tester les algorithmes de recherche d'images mis en œuvre. Ces bases d'images sont très différentes par leur contenu et permettent de valider notre méthode.

1.6.1 La base d'images Trans'Tyfipal

La base d'images Trans'Tyfipal [7] est la base d'images paléontologique de l'université de Bourgogne. Cette base est née d'une informatisation de l'ancienne base de fiches papier (Tyfipal) du laboratoire de biogéosciences de l'université de Bourgogne. Cette base est une référence nationale et internationale en paléontologie et les chercheurs du domaine l'utilisent pour référencer leurs travaux de recherche.

Tyfipal est un acronyme désignant les TYpes et FIgurés en PALéontologie. Cette base contient environ 60 000 spécimens d'objets paléontologiques répartis en plusieurs familles. Comme la base provient d'une base de fiches papier, chaque enregistrement de la base contient d'une part une ou plusieurs image(s) du spécimen lui-même et d'autre part un ensemble de champs de métadonnées comme le lieu de découverte du spécimen, son type, son embranchement, sa famille, l'ère à laquelle il vivait sur Terre, éventuellement la profondeur à laquelle il a été trouvé.

Bien que cette base de données soit une base spécialisée, les photographies des spécimens sont très différentes les unes des autres même au sein d'une même famille. On trouve de nombreuses familles de végétaux et d'animaux dont la plupart sont sous forme de fossiles puisqu'ils datent de quelques milliers (voire millions) d'années. Comme on le verra plus tard, la couleur des spécimens n'est pas un critère de classification en famille puisque certains fossiles ont dû être colorés artificiellement avant leur photographie pour faire ressortir les détails effacés après quelques milliers d'années passées immergés dans la mer ou enfouis dans le sol. Quelques exemples de spécimens de Trans'Tyfipal sont donnés en figure 1.2.

Les images sont au format JPEG. La taille des images de la base varie beaucoup d'un specimen à un autre selon les conditions de prise de vue au moment de la photographie et bien sûr selon le niveau de détail retenu. La plupart des images sont de taille supérieure à 1024 par 1024 pixels.

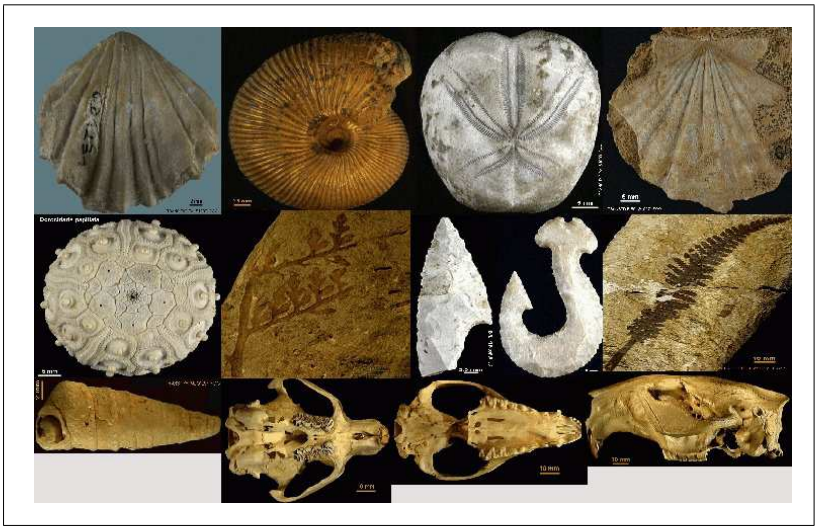


FIG. 1.2 – Quelques exemples d'images issues de la base d'images paléontologique trans'tyfipal.

1.6.2 La base Columbia

La base d'images de l'université Columbia [1] est composée de cent objets photographiés sous soixante-douze positions différentes. Il y a donc au total 7 200 images. Ces images, au format JPEG, sont présentées en figure 1.3. Les images sont toutes de taille carrée : 128 par 128 pixels.

Cette base présente l'avantage de posséder des objets de différentes couleurs et de formes variées. De plus, comme on connaît a priori les cent objets, on peut comparer les classes d'images obtenues aux familles d'objets.

FIG. 1.3 – Les cent objets de la base d'images Columbia.

1.7 Notre contribution

Les détails de notre contribution au domaine de la recherche d'images par le contenu sont donnés au chapitre 6. Notre approche est basée sur l'extraction de vecteurs descripteurs d'images multirésolution. Une phase de sélection des attributs dans le vecteur descripteur nous permet de construire, avec l'aide d'un expert, une base hiérarchique de (quatre) vecteurs signature de taille réduite $n < 10$. Ce sont ces signatures qui sont classées automatiquement en familles grâce à l'algorithme des nuées dynamiques pour former un arbre de recherche qui est proposé à l'utilisateur pour sa navigation visuelle.

Notre méthode effectue un classement des images lors d'une phase hors-ligne pour proposer un arbre de recherche lors d'une phase en ligne. Le temps de calcul de la classification est donc peu important car il se déroule hors-ligne (aucun utilisateur n'est connecté au système). Toutefois, nous avons souhaité optimiser les calculs et la classification afin de pouvoir autoriser une classification en ligne pour une taille de base d'images de l'ordre de 10 000 images. Cette optimisation a pour but de permettre d'effectuer le classement en ligne et d'autoriser une méthode de bouclage de pertinence pour améliorer la qualité des résultats. On trouvera donc dans cette thèse de nombreux points sur l'optimisation qui ne sont pas utilisés directement par notre méthode, mais qui peuvent l'être dans le cas d'une méthode avec apprentissage et bouclage de pertinence (perspectives de la thèse).

Les méthodes employées sont des techniques :

- **de traitement d'images** en ce qui concerne les transformées des images en ondelettes grâce au «*lifting scheme*» pour l'extraction d'attributs,
- **de classification** pour le classement des images en familles visuellement similaires,
- **de gestion de bases de données** afin d'assurer le stockage et la recherche des informations,
- **de programmation de pages web dynamiques** dans le but d'offrir à l'utilisateur une interface conviviale de navigation visuelle dans la base grâce à la souris.

Par rapport aux méthodes «classiques» de recherche d'images par le contenu l'approche retenue permet :

- d'extraire des images des attributs multirésolutions en nombres entiers (gain de temps, aspect multirésolution),
- de travailler avec des vecteurs signatures — construits à l'aide des attributs multirésolution organisés hiérarchiquement — de taille volontairement réduite, typiquement $n \leq 10$ (gain de vitesse, organisation de la recherche selon le contexte),
- d'utiliser ces vecteurs par taille croissante selon une hiérarchie prédéfinie (gain de temps, gain de précision dans la requête),

– de proposer à l'utilisateur un système de navigation visuelle dans la base.

Notre méthode tente d'imiter le comportement de la perception visuelle humaine grâce à l'extraction de caractéristiques visuelles (*gestalts*) et à leur classification multirésolution dans une approche de type grossier-à-fin (*coarse-to-fine*). L'interface utilise des images modèles volontairement floues proposées à l'utilisateur lors de sa navigation.

1.8 Présentation des chapitres

La seconde partie de ce mémoire présentera l'état de l'art de la recherche d'images en détaillant la multidisciplinarité de cette thématique de recherche. Au chapitre 2 seront abordées les bases de données d'images ainsi que les méthodes d'indexation et les techniques de recherche. Le traitement d'images sera détaillé au chapitre 3. L'extraction d'attributs et leur organisation dans des vecteurs descripteurs d'images feront l'objet du chapitre 4. Les techniques de classification de données multidimensionnelles seront étudiées au chapitre 5.

La troisième partie proposera une vue générale des outils disponibles pour créer un système de recherche d'images efficace et robuste. Le chapitre 6 exposera le cahier des charges de notre système de navigation dans les bases d'images. L'implémentation réalisée et les choix informatiques retenus seront présentés au chapitre 7. Les résultats et leur interprétation feront l'objet du chapitre 8.

La quatrième partie proposera une conclusion sur le travail effectué. Pour finir, le chapitre 9 discutera des résultats obtenus par la méthode et donnera les perspectives futures pour l'amélioration de ce travail.

Les librairies Intel utilisées pour le traitement d'images seront abordées de façon détaillée en annexe A. Le calcul du degré d'appartenance d'une image à une classe sera expliqué en annexe B.

Deuxième partie

La recherche d’images

Chapitre 2

Les bases d'images

L'étude de quelques notions de base de données «classiques» permet de mieux comprendre leur architecture et de mieux appréhender la problématique des bases d'images. Ce chapitre décrit les bases de données et les bases d'images en précisant les notions importantes et les principaux problèmes à résoudre pour arriver au système de recherche d'images idéal.

2.1 Problématique

2.1.1 Les bases de données

L'appellation «gestion de bases de données» [Gar83] désigne la branche de l'informatique qui étudie le stockage et l'interrogation des données numériques. Une base de données informatique est donc un ensemble d'informations numériques stockées selon un modèle dans le but de les conserver, de les enrichir et de les interroger avec la garantie de l'intégrité de ces données. Ces informations peuvent être de n'importe quel type : texte, image, son ou vidéo car informatiquement, ces données sont représentées par un ensemble de nombres binaires en mémoire.

Le modèle de la base de données permet d'organiser les informations et de leur ajouter un sens, une sémantique qui représente les relations entre ces objets et le monde réel.

Le système de gestion de bases de données idéal doit fournir un certain nombre de services :

- La centralisation de la gestion des données (mais pas des données elles-mêmes qui peuvent être réparties) doit permettre un regroupement logique des informations,
- L'indépendance des applications par rapport à la structure des données doit faciliter l'évo-

lutivité des applications,
- L'environnement de programmation doit être non procédural, l'utilisateur spécifie ce qu'il veut et non la procédure à suivre pour l'obtenir (QUOI mais pas COMMENT),
- L'environnement d'utilisation doit être convivial et doit offrir un meilleur accès à l'information,
- Le niveau de sécurité et l'intégrité des données doivent être garantis (contrôle d'accès, cryptage, transaction, respect des contraintes sur les données...),
- Les données et les applications doivent être portables sur différents systèmes (indépendance vis-à-vis de l'architecture matérielle et logicielle).

En 1975, l'organisme de normalisation américain ANSI (*American National Standard Institute*) a proposé un modèle normalisé de base de données assurant les services ci-dessus.

2.1.2 Architecture des bases de données

Le schéma 2.1 donne les différentes couches d'abstraction dans un système de gestion de bases de données (SGBD) telles qu'elles ont été normalisées par l'ANSI [ML90]. Ce modèle est composé de quatre parties : le schéma physique, le schéma interne, le schéma conceptuel et le(s) schéma(s) externe(s).

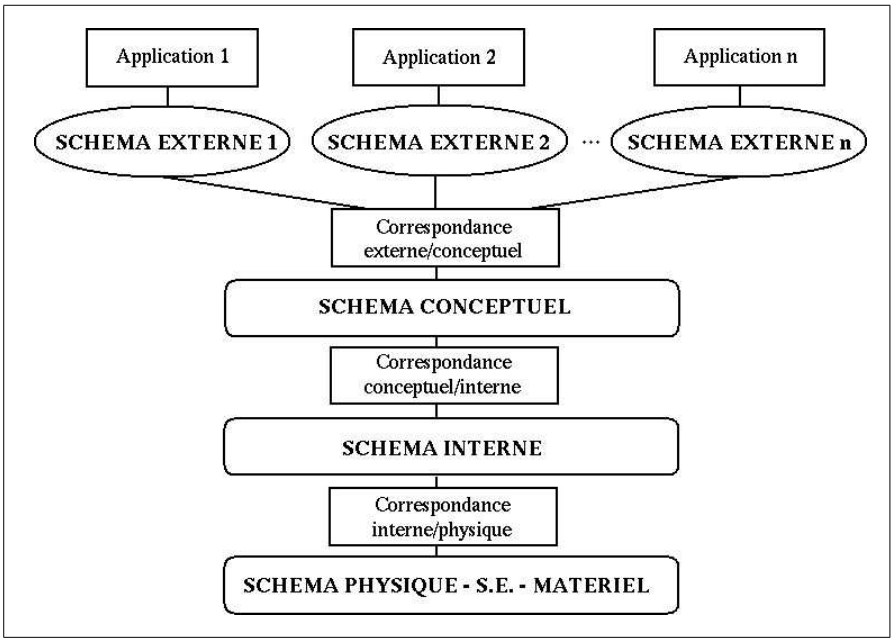


FIG. 2.1 – La structure d'une base de données selon le modèle ANSI.

Le niveau physique : Il est chargé de la gestion physique de la base de données au niveau du matériel et du système d'exploitation. Il assure la gestion des disques, pistes et secteurs

ainsi que la gestion des tampons de lecture/écriture en mémoire. C'est le plus bas niveau de l'architecture, la réalité binaire de la base.

Le niveau interne : Le schéma interne fournit une perception plus technique de la base de données. On décrit à ce niveau un ensemble d'objets informatiques (fichiers, index, listes...) dont l'organisation et les caractéristiques visent à optimiser les ressources (disques, mémoires, microprocesseur) lors de l'exploitation de la base de données.

Le niveau conceptuel : Le niveau conceptuel décrit les concepts utilisés dans la base. Il est totalement indépendant de la technologie utilisée (matérielle et logicielle) pour gérer la base. Il se compose de deux parties : la structure de données qui comprend l'ensemble des données et des liens pour les applications et les contraintes d'intégrité qui garantissent la cohérence et la vraisemblance des données.

Le niveau externe : Ce niveau représente le niveau application dans lequel les utilisateurs n'ont accès qu'à une vue partielle de la base de données (celle qui les intéresse). Il y a un nombre quelconque de schémas externes pour une base de données qui dépend du nombre d'utilisateurs et de leurs droits d'accès en fonction des applications.

Ce découpage en couches permet d'assurer une indépendance de chacune des parties de la base de données. Ainsi, on pourra aisément mettre le même niveau conceptuel ou le même niveau externe sur plusieurs systèmes d'exploitation différents sans avoir à tout refaire. On sépare ainsi le contenu du contenant (analogie avec XML [8]).

2.1.3 Le modèle relationnel

Le modèle relationnel [Cod70] est le modèle le plus utilisé dans les bases de données actuelles. Son succès vient du fait qu'il est basé sur la notion mathématique d'algèbre et qu'il possède donc un comportement régi par des règles rigoureuses. Il contient des règles de création, de manipulation et d'interrogation des données assurant un état fiable des données à n'importe quel instant de la vie de la base de données.

Le langage de manipulation et d'interrogation des données est *SQL* (*Structured Query Language*). Il offre une syntaxe très intuitive pour les requêtes et permet de réaliser toutes les opérations de l'algèbre relationnelle (sélection, projection, jointure...). Il existe d'autres modèles de bases de données qui ne seront pas abordés dans cette thèse.

Les bases de données relationnelles contiennent principalement des données alphanumériques (lettres et chiffres). Il existe pour ces types de données des techniques d'indexation et de recherche très rapides basées sur la structuration des données sous forme d'arbres (B-arbre). Ces techniques bénéficient des nombreuses années de recherche déjà effectuées dans le domaine de la recherche de texte, les principaux résultats sont accessibles dans les actes de la conférence

TREC [31] (Text REtrieval Conference) qui est la référence dans cette discipline. C'est grâce à toutes ces techniques que la recherche sur Internet par moteurs de recherche est possible.

Pour les bases de données textuelles, il existe des méthodes de construction permettant de concevoir la base de données dans son intégralité en partant du problème du monde réel pour arriver à la modélisation informatique de ce problème. La méthode MERISE [NEC+01], pour ne citer qu'elle, offre toutes les étapes de la conception d'un système d'information. Malheureusement, il n'existe pas, à l'heure actuelle, une méthode similaire pour les bases d'images en raison de la complexité intrinsèque des images.

La structure des données textuelles permet de les classer rapidement selon un ordre alphanumérique simple (ordre alphabétique par exemple). Ceci facilite l'indexation et la recherche. Les moteurs de recherche Internet sont basés sur l'indexation des mots contenus dans les pages web et des techniques d'association par synonymes et par relation de proximité entre les mots. Le fonctionnement des moteurs de recherche [Lel98] permet de comprendre les techniques d'indexation mises en œuvre pour l'indexation de textes. Il n'existe pas d'ordre aussi simple sur les images, le problème est de définir les critères de classement des images.

2.1.4 Les bases de données d'images

Les bases de données gèrent de façon efficace les données de type texte mais sont mal adaptées aux données multimédia. Toutefois et afin de pouvoir inclure des images dans les bases de données, un type spécial a été ajouté dans les bases de données relationnelles. Il s'agit du type objet binaire (*BLOB*) dans lequel on peut mettre une image, un code exécutable ou n'importe quel objet informatique, quel que soit sa taille. Ce type ne permet pas de résoudre les problèmes de la recherche d'images, mais il permet de stocker les données multimédia : images, vidéos et sons dans la base. La plupart des bases de données ont donc la capacité de stocker des données multimédia, mais sans moyen pour les interroger dans le modèle lui-même.

Une base d'images est donc une base de données contenant des images et/ou leur représentant construite selon un certain modèle dans le but de la stocker, de l'interroger, de l'enrichir, de la partager.

On classe les bases d'images en deux grandes catégories pour la recherche et l'indexation.

- Les **bases généralistes** sont des bases d'images de sujets très variés comprenant des familles d'images très différentes (par exemple couchers de soleil, montagne, plage, personnages, véhicules, bâtiments...).
- Les **bases spécialisées** sont des bases dans lesquelles on va trouver des images d'un domaine particulier (images médicales, images satellites, images architecturales, photos de visages ou tableaux d'un musée par exemple).

Bien qu'il y ait une distinction entre bases généralistes et bases spécialisées, les bases spécialisées ne sont pas plus faciles à interpréter que les bases généralistes. Dans le cas typique de

Trans'tyfipal, on se trouve dans le domaine de la paléontologie, c'est-à-dire une base spécialisée dont les images n'ont pourtant parfois que très peu de choses en commun.

2.1.5 La recherche d'images

Les premiers systèmes de recherche d'images utilisaient des mots-clés associés aux images pour les caractériser. Grâce à cette association de mots-clés, il suffit d'utiliser les méthodes basées sur le texte pour retrouver les images contenant les mots-clés. Plusieurs moteurs de recherche [22, 23] proposent ces recherches d'images basées sur le texte. Ils s'appuient sur le principe simple que dans une page web, il y a une forte corrélation entre le texte et les images présentes. Le principal problème de ces recherches par mots-clés est que le résultat peut être complètement hors sujet. Les figures 2.2 et 2.3 donnent un exemple de recherche sur Google pour des images de «Jules César».

Comme on peut le constater, pour la recherche «Jules César», la première page de la recherche donne des représentations de Jules César sous forme de portraits, de statues, d'images de bandes dessinées, les résultats sont bons. Mais la seconde page fournit également des images de deux charmants chatons Jules et César, une photographie d'un hotel (intérieur et extérieur) qui ne sont pas forcément des éléments intéressants pour un exposé sur Jules César par exemple.

On voit dans cet exemple que la recherche par mot clé dans les bases d'images peut donner de bons résultats mais révèle aussi quelques inconvénients :

- Un utilisateur peut tout à fait créer un site sur Jules César et mettre des photos de paysages en les renommant julescesar.gif, cesar.jpg et ainsi tromper le système d'indexation d'images (basé sur le texte) de google.
- L'association de textes à l'image est une démarche réaliste pour de petites bases de données (taille inférieure à 10 000 images), mais est complètement impensable pour de grandes bases de données (nombre d'images supérieur à 10 000). En effet, le temps passé à l'association de mots-clés et la pertinence des mots-clés restent très subjectifs et très dépendants des personnes qui effectuent l'association.

Une solution consiste à ne pas utiliser les mots-clés et donc à considérer l'image et uniquement l'image pour effectuer les recherches. Cette méthode est la **recherche d'images par le contenu**. En règle générale, les systèmes de recherche d'images par le contenu fonctionnent par comparaison d'un vecteur descripteur d'une image requête avec les vecteurs descripteurs des images de la base selon une métrique donnée.

2.1.6 La recherche par le contenu

La recherche par le contenu consiste à rechercher les images en n'utilisant que l'image elle-même sans aucune autre information. On ne considère que l'image numérique (voir figure 3.3),

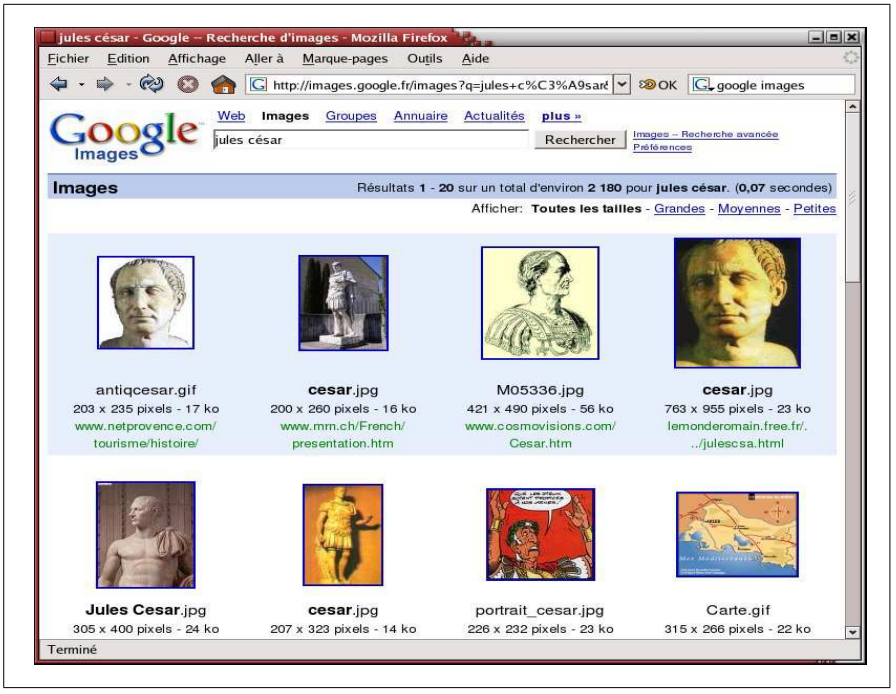


FIG. 2.2 – Une recherche d'images par le texte «Jules César» sur Google.

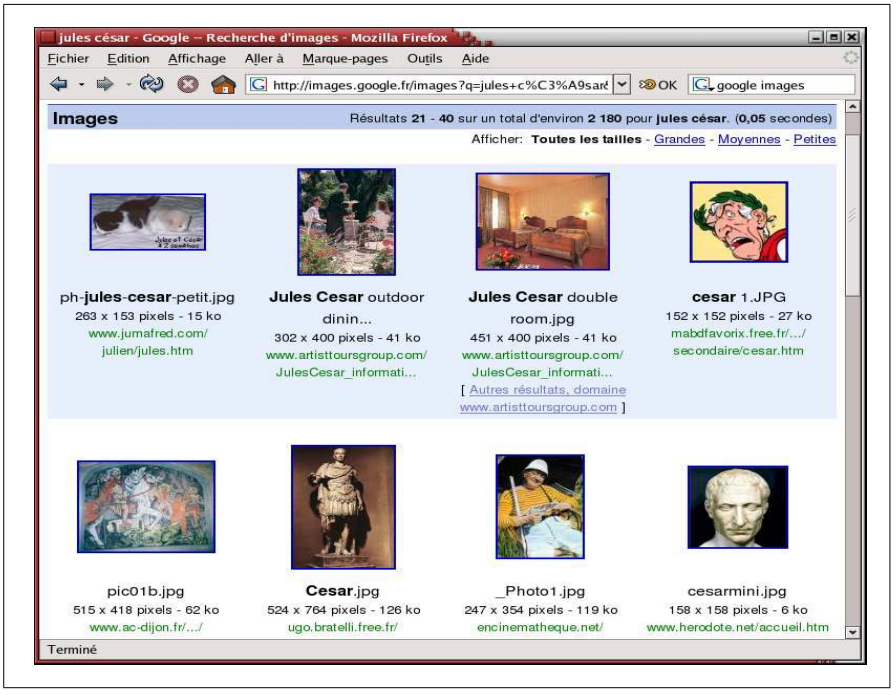


FIG. 2.3 – Une recherche d'images par le texte «Jules César» sur Google.

c'est-à-dire un tableau de pixels (pour *picture elements*, éléments d'images) à deux dimensions (largeur et hauteur). Une image couleur RVB possède trois composantes tandis qu'une image en niveaux de gris n'en possède qu'une seule.

Un problème posé par cette approche est la quantité très élevée de points contenus dans une image. Par exemple une image couleur de $1600 * 1200$ pixels contient $1600 * 1200 * 3 = 5\ 760\ 000$ points. Il est donc nécessaire, d'une part pour réduire le temps de calcul et d'autre part pour comparer des images de taille différentes, de travailler avec un ensemble réduit d'attributs d'images. Ces **attributs** sont des informations spatiales, colorimétriques, géométriques ou statistiques, extraites de l'image, qui synthétisent au mieux l'information contenue dans celle-ci.

Ces attributs sont regroupés dans un vecteur V_i appelé **vecteur descripteur** de l'image i. V_i possède n composantes réelles (en général) qui sont les attributs extraits de l'image, $V_i \in \mathbb{R}^n$. Pour décrire au mieux une image, il faut tenir compte des transformations géométriques qu'elle peut subir. Il faut donc trouver des descripteurs d'images invariants par rotation, translation et changement d'échelle pour assurer une indépendance de V_i vis-à-vis de ces transformations.

Au lieu de stocker les images elles-mêmes dans la base (ce qui peut poser des problèmes de taille de données dans les grandes bases d'images), on peut choisir de ne stocker dans la base que le descripteur de l'image et son URI (*Uniform Resource Identifier*), c'est-à-dire le chemin absolu pour la retrouver sur Internet ou sur un disque dur local. Ainsi, on dispose des descripteurs d'images pour comparer les images et on va les chercher seulement si on a besoin de les afficher par exemple. Cela diminue sensiblement la taille de la base de données et donc la vitesse d'accès aux informations stockées dans celle-ci.

Le problème de recherche d'information a amené la création de la norme spécifique MPEG-7 [24] qui définit des descripteurs d'images dans les contenus audio-visuels et permet une recherche par similarité de scènes dans les vidéos. Cette norme fournit aussi un langage et des outils de création et de gestion de schéma de description de contenu audio-visuel. MPEG-7 n'est pas une norme de codage de vidéo comme MPEG-2 et MPEG-4 mais un standard de description de contenu qui permet de décrire des vidéos mais aussi du texte, des sons, des images fixes. MPEG-7 permet par exemple de décrire le contenu de vidéos MPEG-2 ou MPEG-4 pour une recherche plus facile. MPEG-7 est basée sur XML, la norme est aussi extensible, elle permet d'utiliser ses propres descripteurs.

2.2 L'interprétation des images

Les méthodes de recherche existantes sont basées sur les informations contenues dans l'image. On peut classer ces informations selon un modèle à plusieurs niveaux sémantiques. La figure 2.4 montre les différents niveaux, de la valeur la plus basse : les pixels de l'image à la valeur la plus haute : la description de la scène. Les pixels de l'image participent à l'interpré-

tation **bas-niveau** de l'image alors que la description de la scène correspond à l'interprétation **haut-niveau** de l'image.

La **segmentation d'images** permet de trouver les différentes régions qui composent une image. De nombreuses techniques de segmentation [BCC+95] ont été mises au point et offrent un début de solution au problème de reconnaissance de scènes.

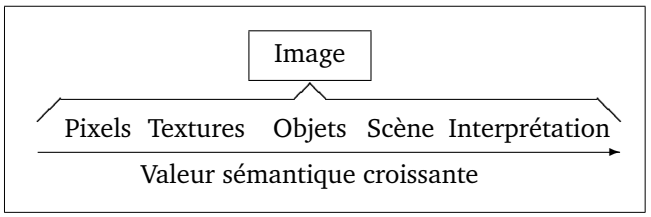


FIG. 2.4 – Les niveaux sémantiques dans une image.

Lorsque nous regardons une image, nous pouvons analyser son contenu grâce à nos connaissances. En effet, si on nous présente une image de paysage de montagne, nous allons reconnaître immédiatement la montagne, les arbres, le chalet. Maintenant, si on nous présente la photo de l'oursin 2.5, nous verrons une forme sphérique (car nous percevrons le relief) sans pour autant savoir de quoi il s'agit. On pourrait penser à une vue grossie d'un virus ou d'une bactérie ou encore à la vue de dessus d'un champignon...

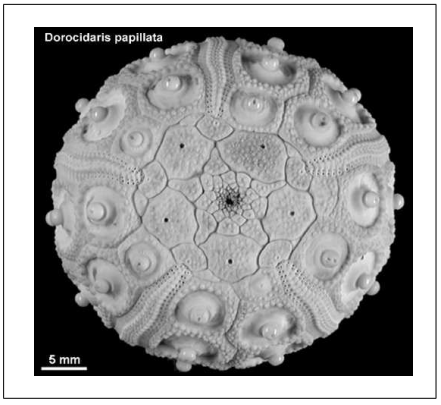


FIG. 2.5 – Oursin : sans autre indication, on pourrait reconnaître une vue grossie d'un virus ou d'une bactérie... Car nous utilisons notre connaissance pour identifier le contenu d'une image.

En revanche, personne n'y verra jamais un porte-avions, une maison ou une paire de lunettes. Nous utilisons toujours nos connaissances pour décoder une image. Le problème est d'arriver à simuler le comportement humain à l'aide d'un ordinateur et de trouver des algorithmes capables de reconnaître des objets dans une image. Il existe de nombreuses méthodes de reconnaissance de formes mais aucune technique actuelle ne permet de reconnaître complètement une scène et les objets qui la composent.

2.2.1 La sémantique

La difficulté dans les systèmes de recherche d'images par le contenu est d'associer une valeur sémantique à une image. À partir des pixels qui représentent une information **bas-niveau**, il est très difficile d'arriver à l'interprétation **haut-niveau** de l'image. La figure 2.4 montre à quel point ce pas est difficile à franchir puisqu'à l'heure actuelle, reconnaître si un certain animal est présent ou non dans une image est encore un problème difficile à résoudre.

Dans l'étape de segmentation, les pixels sont associés dans des régions de différentes textures. Ces textures définissent les objets qui composent la scène conduisant à l'interprétation de l'image. Ainsi, donner un sens à une image signifie qu'à partir d'une suite de pixels, on va être capable de définir les objets présents dans la scène. Or il n'existe pas de technique de reconnaissance capable de recréer ce processus d'analyse qu'un enfant de quatre ans arrive à faire au premier coup d'œil.

2.2.2 Le fossé sémantique

Les figures 2.6 et 2.7 donnent le résultat d'une recherche par le contenu respectivement dans une base d'images généraliste et dans une base d'images spécialisée. Les images représentées sont associées à la mesure de leur distance à l'image requête selon une métrique donnée. La première image est l'image requête elle-même (la distance associée est nulle). Dans ces exemples, la métrique utilisée est la distance euclidienne.

On s'aperçoit que dans le cas d'une base généraliste comme dans celui d'une base spécialisée, les premiers résultats de la recherche correspondent bien à des images visuellement similaires mais très rapidement, on obtient des images qui ne sont plus visuellement similaires et qui n'ont plus rien à voir avec la requête. Ce phénomène est connu sous le nom de **fossé (ou vide) sémantique (*semantic gap*)**.

Il est très difficile de trouver des descripteurs qui permettent de prendre en compte d'une part toutes les représentations possibles d'une même scène et d'autre part la subjectivité des requêtes formulées par les utilisateurs. Dans le domaine de la recherche d'images, on doit tenir compte du fossé sémantique et trouver des techniques permettant de le combler au moins en partie.

2.2.3 La requête par l'exemple

L'architecture générale des systèmes de recherche d'images par le contenu est basée sur un calcul de similarité entre une image exemple et les images de la base d'images. Plus précisément, on classe les images résultat présentées à l'utilisateur selon la distance entre le vecteur descripteur de l'image exemple et les vecteurs descripteurs des images de la base. La figure 2.8 montre ce principe.

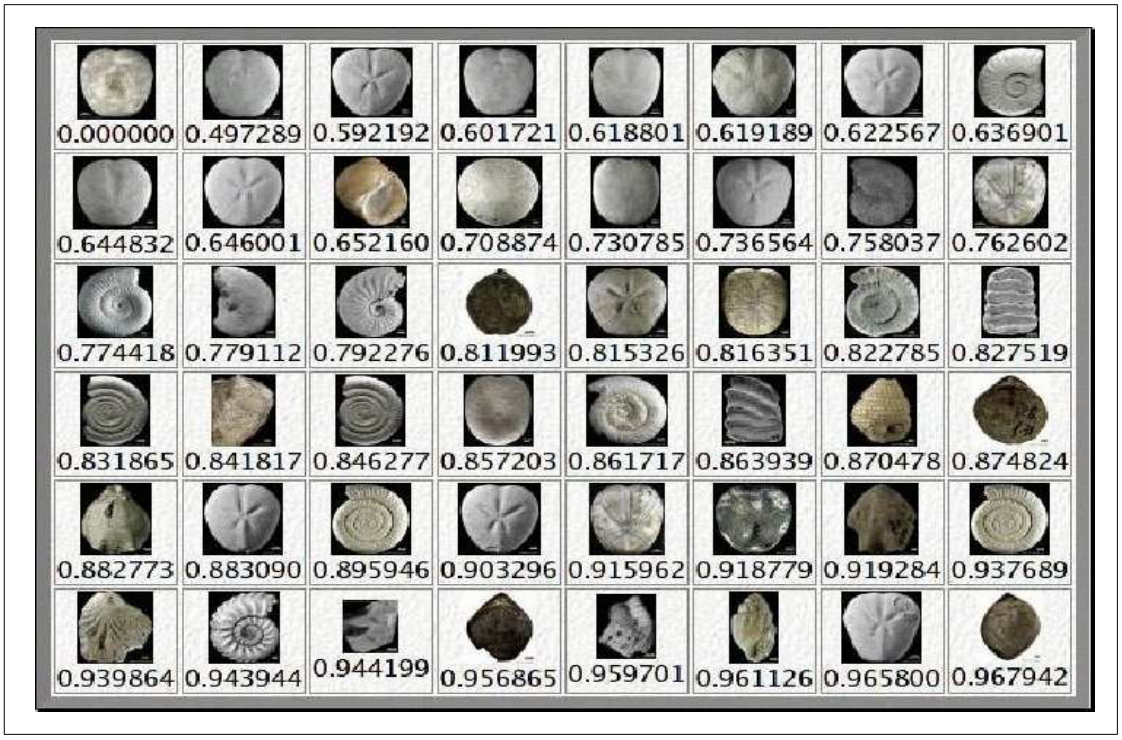


FIG. 2.6 – Fossé sémantique entre les données et leur interprétation dans le cas d'une base d'image généraliste.

FIG. 2.7 – Fossé sémantique entre les données et leur interprétation dans le cas de Trans'Typipal.

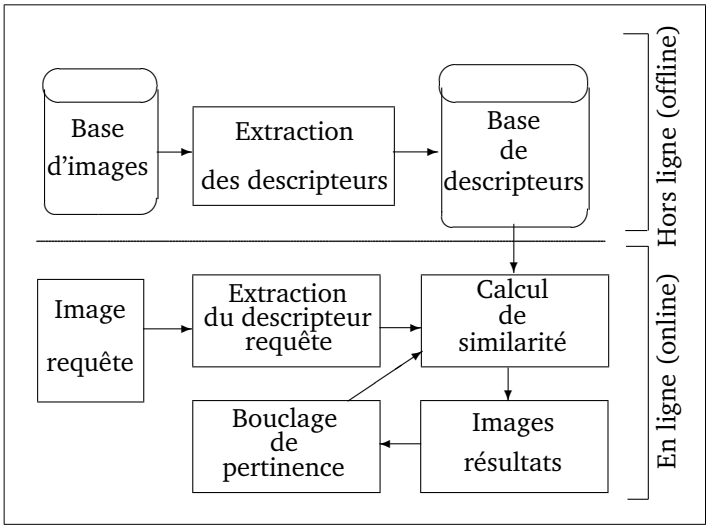


FIG. 2.8 – Architecture générale des systèmes de recherche d'images par l'exemple.

On distingue deux phases indépendantes :

– Une phase **hors ligne** (*Offline*) dans laquelle sont réalisés l'extraction et le stockage des descripteurs des images de la base. Durant cette phase, aucun utilisateur n'est connecté à la base d'images. Cette phase peut donc prendre le temps nécessaire à l'évaluation des descripteurs.
– Une phase **en ligne** (*Online*) où l'utilisateur interroge la base à l'aide de son image exemple. Durant cette seconde phase, le temps de réponse du système est crucial, il faut le réduire au maximum.

Cette architecture générale présente cependant deux inconvénients majeurs :

– L'utilisateur ne dispose pas toujours d'une image exemple pour formuler sa requête,
– Lorsque le nombre d'images de la base augmente, le temps de recherche devient rapidement prohibitif.

L'idée de notre méthode est d'effectuer un pré-classement des images durant la phase hors ligne afin de proposer à l'utilisateur une base d'images pré-classée lors de la phase en ligne. Ainsi, il a la possibilité de naviguer dans la base à la recherche de l'image qui lui convient. La recherche devient **visuelle**.

2.2.4 Du côté de l'utilisateur

L'aspect utilisateur est très important dans la création d'un système de recherche d'images. L'utilisateur doit disposer d'un programme de recherche d'images simple d'utilisation, rapide et efficace. Ces contraintes doivent être rigoureusement respectées si on veut fournir un bon système de recherche d'images.

On distingue deux types d'utilisateurs d'une base d'images :

– Le **non-spécialiste** qui cherche une image sans avoir de connaissance particulière du domaine de la base,
– L'**expert** qui connaît parfaitement le domaine de la base d'images et qui connaît (ou qui veut tester) les attributs à utiliser pour classer la base selon ses critères.

Il faut donc adapter le système à des utilisateurs experts ou non du domaine de la collection. C'est-à-dire qu'un utilisateur néophyte demandera simplement à rechercher une image dans la base alors qu'un expert pourra modifier les paramètres afin de prendre en compte sa connaissance du domaine.

Lors de l'utilisation d'un système de recherche d'images, l'utilisateur dispose de plusieurs moyens pour définir sa requête.

– **requête par esquisse** (*sketch*) : L'utilisateur dessine une esquisse très simple de ce qu'il cherche (contours et couleurs).
– **requête par image exemple** : l'utilisateur dispose d'une image de référence de ce qu'il cherche et la donne en entrée du système. Le principal inconvénient de cette méthode est qu'il faut disposer d'une image approchée de ce que l'on cherche.
– **requête par description** : l'utilisateur décrit ce qu'il veut en terme de couleurs et de relations spatiales : 20% de bleu, du vert en haut et du rouge en bas par exemple.

Dans ces trois cas de formalisation de la requête, l'utilisateur doit avoir une idée de ce qu'il cherche. La plupart du temps, l'utilisateur n'a qu'une idée vague de l'image qu'il souhaite obtenir et c'est en visualisant les images de la base qu'il cible sa requête. Il existe donc une quatrième manière d'effectuer la requête : la **navigation visuelle** dans la base. C'est le principe que nous avons retenu dans notre solution.

Par exemple, une personne trouve un objet fossile par hasard lors d'une promenade. Elle n'est pas spécialiste en paléontologie. Notre système peut lui proposer une navigation dans une base d'objets paléontologiques en partant par exemple de caractéristiques de formes, puis de texture pour arriver à un ensemble restreint d'images qui correspondra à la famille paléontologique de l'objet. La difficulté est de construire une représentation pour guider l'utilisateur lors de sa recherche visuelle et l'amener vers la bonne solution.

2.2.5 La navigation visuelle

L'idée de la navigation visuelle est de présenter à l'utilisateur du système de recherche d'images une vue générale de la base de données dans laquelle il navigue simplement à l'aide de la souris. Cette technique nécessite bien sûr d'organiser la base d'images en «familles» d'images qui seront visuellement similaires.

Le principal problème de ce type de méthode est de choisir les images que l'on va proposer à l'utilisateur lors de sa recherche. Doit-on proposer des images réelles de la base ou

bien construire des images représentatives du contenu de la base (et dans ce cas, comment les construire) ?

Il se pose ensuite un autre problème, quelle liste d'images doit-on proposer à l'utilisateur pour démarrer sa recherche. Ce problème de démarrage de la navigation est connu sous le nom de problème de la **page zéro**. Il faut donc un moyen de «synthétiser» le contenu de la base d'images afin de proposer à l'utilisateur du système une série d'images de départ à l'exploration de la base.

Un autre problème de la recherche d'images par le contenu est la définition de la notion de similarité des images.

2.2.6 La similarité

La notion de similarité entre deux images est une notion difficile à définir. Il faut en effet préciser sur quel(s) critère(s) cette notion de similarité se base. La figure 2.9 présente un exemple où on demande à l'utilisateur de classer les quatre images proposées en familles d'images similaires.

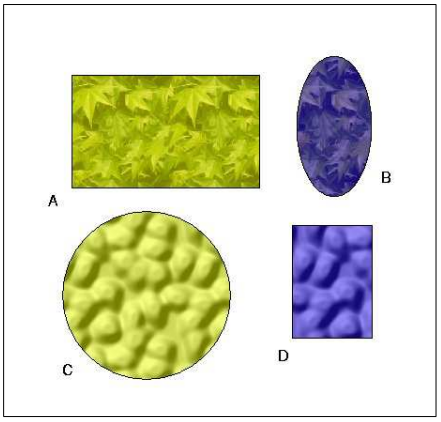


FIG. 2.9 – Exemple de base à classer en familles d'images similaires.

Il y a plusieurs réponses acceptables à cette question selon le **critère de similarité** retenu. Si on décide de retenir la couleur comme critère, les deux familles sont (A,C) et (B,D). Si c'est la texture que nous retenons, on aura (A,B) et (C,D). Si c'est la forme que nous choisissons comme critère, les deux familles deviennent (A,D) et (B,C). Enfin, si nous choisissons la taille, le regroupement devient (A,C) et (B,D) comme dans le premier cas. Il est aussi possible de définir une composition de ces critères avec un résultat qui dépend de leur importance relative.

Cet exemple est volontairement caricatural mais il indique qu'il faut tenir compte du critère de similarité lors de la recherche. Autrement dit, il faut créer un vecteur descripteur qui contient les informations selon un ou plusieurs critère(s) choisi(s) en fonction des besoins de l'utilisateur.

La similarité de deux images est un problème mal posé dans lequel des contraintes additionnelles sont nécessaires pour la régularisation. Il y a des choix à faire a priori pour résoudre ce problème. Il faut donc définir un critère de similarité avant de comparer deux images.

Dans la définition de la similarité et dans l'extraction des attributs des images, l'expert du domaine joue un rôle très important. Il choisit les caractéristiques des objets et propose un modèle de comparaison des images. Il apporte son expérience du domaine comme connaissance a priori de la similarité entre les images.

2.3 L'aspect psycho-visuel

La perception visuelle est l'un des mécanismes les plus complexes du corps humain. Notre cerveau est capable d'analyser des scènes complexes en une fraction de seconde. De nombreux scientifiques travaillent sur la perception visuelle. L'étude complète des mécanismes mis en œuvre lors de l'interprétation d'une scène nécessite un large champ de compétences. La physique (l'optique) donne les lois qui régissent la lumière (et les couleurs) reçue par l'œil, la biologie permet d'étudier l'œil en tant que récepteur de lumière, la médecine (neurobiologie) s'intéresse à la transmission et au décodage de la lumière perçue par la rétine au niveau du cerveau. Enfin la psychologie de la vision étudie le comportement et les réactions de notre système visuel.

L'œil est un récepteur photo-sensible qui est capable de transmettre au cerveau des informations à travers le nerf optique. Schématiquement, l'œil fonctionne comme une lentille convergente qui dirige le faisceau de lumière observé vers la rétine. La figure 4.3 présente la structure interne de l'œil. La rétine transmet l'information lumineuse au cerveau pour son décodage et son interprétation.

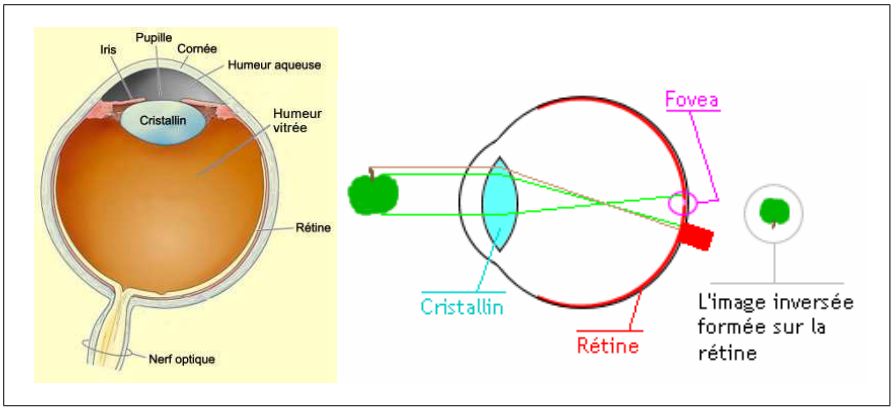


FIG. 2.10 – Structure interne de l'œil.

La médecine permet de connaître avec précision le fonctionnement de l'œil. La transmission de l'information entre l'œil et le cerveau est un processus connu. La rétine est composée de

plusieurs couches successives, elle possède des cellules neuronales photo-réceptrices. Ces neurones, dont le rôle est la transformation de l'image perçue en image neuronale, sont reliés au nerf optique. La rétine effectue donc un pré-traitement de l'image reçue avant d'envoyer des informations au cerveau [Wan95]. L'interprétation des informations reçues par le cerveau est un problème complexe. Comment interprète-t-on une image ?

Les études menées en psychologie de la vision sont nombreuses [TB98]. Une théorie ancienne (années 1920) effectue son retour au premier plan, c'est la théorie des *gestalt* [RYM99]. Dans cette théorie, le regroupement est le processus principal de la perception visuelle.

Quand des points (ou des objets précédemment définis) possèdent une ou plusieurs caractéristiques communes, ils sont regroupés pour former un objet visuel plus grand, un *gestalt*. Ce regroupement est effectué selon plusieurs lois de regroupement. Les lois utilisées par le cerveau pour les regroupements sont des lois géométriques et perceptuelles simples : proximité, voisinage, continuité de direction, fermeture, parallélisme, couleur, expérience... La figure 2.11 donne des exemples de lois de regroupement utilisées par le cerveau pour l'association de primitives visuelles en *gestalts*. Cette figure est issue d'un article qui illustre l'utilisation des propriétés de la théorie des *gestalts* en vision artificielle [DMM04].

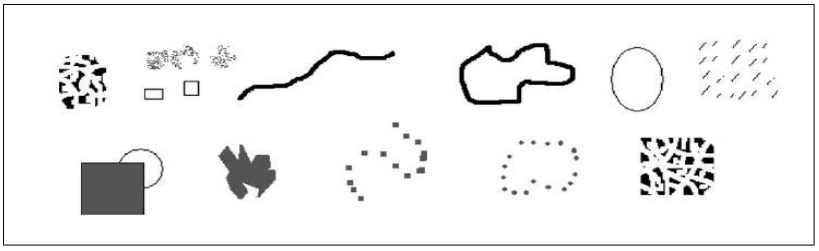


FIG. 2.11 – Exemples de lois de regroupement : proximité, similarité de forme et de texture, continuité de direction, fermeture, convexité, parallélisme, complétion amodale, couleur constante, continuité, fermeture par continuité, complétion de forme (d'après Desolneux et al.).

Ces primitives définissent des lois gestalt utilisées dans notre système de vision. Notre vision fonctionne donc sur le principe du regroupement de primitives visuelles (*gestalt*) entre elles, partant d'objets fins pour arriver à des objets de plus en plus gros. C'est une approche fine à grossière (*fine-to-coarse*) qui part des atomes de vision pour arriver à des objets graphiques complexes. L'analyse en ondelettes est donc sans doute un excellent outil pour approcher ce mode de fonctionnement grâce à une décomposition multi-échelle des images traitées.

Ce parallèle entre le fonctionnement de la vision chez les mammifères et l'analyse en ondelettes a déjà été formulée par Stéphane Mallat [Mal96]. Les imagettes d'approximation et les trois imagettes de détails donnent respectivement une approximation de l'image de départ (vue plus éloignée de l'image de départ) et les détails horizontaux, diagonaux et verticaux perdus

lors du passage de l'image de départ à l'image transformée. Les trois imagettes de détails offrent la possibilité de détecter des primitives visuelles (*gestalts*) en raison de leur séparation en trois directions principales : horizontale, verticale et diagonale.

Il y a deux niveaux de traitement dans l'interprétation visuelle des images [AB91] : un bas niveau (*early vision*) utilise le regroupement de *gestalts* et un haut-niveau d'interprétation (high-level vision) de la scène. Le principe fondamental de l'interprétation haut-niveau est d'interpréter des scènes de la façon la plus simple possible. La figure 2.12 montre quelques exemples d'illusions d'optique qui mettent en défaut notre interprétation haut-niveau en raison de regroupements erronés de primitives bas-niveau.

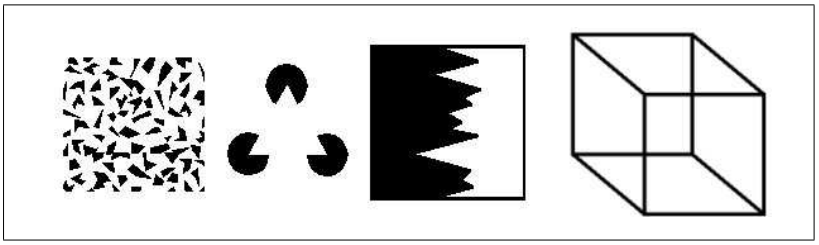


FIG. 2.12 – Illusions d'optique. Dans les deux premiers dessins, notre perception détecte un triangle alors qu'il n'y en a pas. Dans les deux dessins suivants, on voit des images bimodales qui peuvent être interprétées de deux façon différentes.

Si on demande à quelqu'un de dessiner une clé ou une tête de bonhomme, on obtiendra toujours une clé vue de côté et une tête vue de face car nous possédons des prototypes d'objets dans notre mémoire. Nous utilisons notre connaissance pour dessiner comme pour interpréter des scènes.

Il existe deux modèles principaux de la perception visuelle haut-niveau [TB98]. Le premier modèle considère que le cerveau reconstruit les scènes 3D à partir de la vue 2D imprimée sur la rétine, c'est la **description structurelle** (*structural description*). Le second modèle est **l'approche basée image** (*image-based*) qui considère que la reconnaissance de scène est effectuée non pas par reconstruction 3D, mais par repérage de caractéristiques locales dans l'image envoyée par la rétine.

Les résultats récents en neurophysiologie et en vision artificielle tendent à unifier ces deux modèles en considérant que l'approche basée image possède des lacunes qui sont palliées par l'introduction de données structurelles. La vision fonctionne donc par reconnaissance de caractéristiques locales puis par regroupement de ces informations en structures globales.

Les ondelettes permettent la décomposition des images en plusieurs niveaux de résolution, avec une échelle qui va d'une image grossière à une image très détaillée. Le cerveau humain utilise également cette approche pour décomposer une scène. Le système de recherche d'images

développé possède la propriété de travailler tout comme le cerveau par raffinements successifs en partant d'une approximation de la scène observée pour arriver à son interprétation en ajoutant des détails.

2.4 État de l'art

2.4.1 Méthodes générales

Dans les systèmes de recherche d'images par le contenu, la méthode générale consiste à calculer pour chaque image un vecteur descripteur V_i de dimension n, stocké dans une base de données, qui représentera l'image i lors de la recherche. Tous les calculs sont effectués hors-ligne (*offline*) afin de ne pas pénaliser l'utilisateur avec des calculs souvent longs.

Une image exemple est donnée au système de recherche qui calcule le vecteur descripteur associé V_r et évalue la distance du vecteur requête avec chacun des vecteurs de la base. Ensuite, les images correspondant aux vecteurs les plus proches du vecteur requête sont présentées à l'utilisateur comme étant les plus proches au sens de la distance considérée.

En raison des nombreuses publications dédiées au domaine de la recherche d'images, un état de l'art est une entreprise ambitieuse. Toutefois, quelques auteurs [RHC99, VD00, San01, VBK01] s'y sont risqués et présentent l'étendue des recherches dans ce domaine.

L'article de Forsyth et al. [FMW97] présente les problèmes que soulève cette thématique. Les utilisateurs cherchent des images dans des contextes variés avec des objectifs différents. C'est cette diversité des contextes d'utilisation qui rend la tâche de recherche d'images difficile. La recherche d'information visuelle de Gupta et al. [GJ97] répertorie toutes les difficultés que doit surmonter un système de recherche d'images.

Les techniques utilisées donnent en général des résultats assez différents et difficilement comparables en raison de l'utilisation de bases d'images différentes. L'unique moyen de tester les méthodes de recherche d'images est de créer une base d'images libre de droits téléchargeable par tous les acteurs du domaine pour tester leurs résultats. C'est l'objectif que c'est fixé le projet benchathlon [25] qui propose en ligne une base d'images de test. Ce projet manque encore de maturité mais peut devenir une référence de base d'images avec vérité terrain (*ground truth*) dans les années à venir.

De nombreuses définitions de distances ont été proposées dans la littérature, chacune donnant bien entendu des résultats différents, le choix d'une fonction distance est primordial dans la recherche d'images. Les distances métriques sont présentées au paragraphe 5.2.

Les techniques d'indexation de données multidimensionnelles que sont les vecteurs descripteurs d'images sont décrites de façon exhaustive par Sid-Ahmed Berrani [Ber04] dans sa thèse. Il utilise une recherche approximative des plus proches voisins afin de déterminer les images les plus proches de l'image requête de l'utilisateur.

Dans sa thèse, Scott Cohen [Coh99] aborde le problème de la recherche d'images par le contenu avec une approche de distribution couleur et de reconnaissance de contours d'objets en utilisant la distance EMD (*Earth Mover Distance*). Il étudie également les requêtes partielles dans lesquelles on cherche des images par comparaison d'attributs de régions similaires. Une technique originale offrant la possibilité d'avoir plusieurs images requêtes au lieu d'une seule — *Query by Example Sets (QBES)* — a été développée par John Zachary dans sa thèse [Zac00]. L'indexation des images se fait par codage de l'entropie de l'image et calcul de distance *Maximum Distance Entropy* (MDE).

Les méthodes basées sur les points d'intérêt ont fait l'objet de nombreuses recherches. Cordelia Schmid propose une étude de cette technique dans sa thèse [Sch96]. Des améliorations ont été apportées par Patrick Gros [Gro98] et Etienne Loupias [Lou00] propose une version multirésolution des points d'intérêts. Une contribution importante qui permet de trouver des points d'intérêts invariants par transformation affine du plan est apportée par Krystian Mikolajczyk [Mik02].

L'étude d'une description invariante de contours par les transformations affines du plan est réalisée par Stanislaw Matusiak [Mat99]. Des méthodes utilisant la logique floue permettent d'améliorer les résultats en ajoutant la notion de flou dans la requête. Les travaux de Patrick Lambert [LG01] pour la société GoodShoot [6] détaillent ces méthodes. La thèse de Julien Fauqueur [Fau03] étudie la recherche d'images par composition de catégories de régions issues de la segmentation couleur des images.

Il existe de nombreuses techniques d'extraction d'attributs d'images basées sur les ondelettes.

2.4.2 Méthodes basées sur les ondelettes

Notre technique est basée sur une décomposition multirésolution des images en utilisant l'analyse en ondelettes à l'aide de l'algorithme *lifting scheme*. Mallat [Mal96] a démontré que les ondelettes étaient un excellent outil pour détecter les singularités à plusieurs échelles dans un signal. De plus, Mallat a souligné l'aspect multirésolution et sa ressemblance avec le mode de fonctionnement du cerveau humain pour la perception visuelle et auditive. Notre méthode est fondée sur cette remarque : nous utilisons les ondelettes pour caractériser les images et en extraire des attributs.

D'autres approches de systèmes de recherche d'images utilisant les ondelettes ont déjà été publiées. Dans leur article, Jacobs et al. [JFS95] utilisent l'analyse multirésolution pour créer un index généré à partir des valeurs les plus élevées (en valeur absolue) des coefficients d'ondelettes. Ils utilisent les ondelettes de Haar et une fonction distance adaptée à leur représentation de coefficients d'ondelettes. Mandar et al. [MAP98] ont proposé une autre technique basée sur le calcul de moments à partir des coefficients d'ondelettes et la construction d'histogrammes. Idris

et al. [IP95] donnent une méthode de quantification vectorielle dans laquelle la comparaison est effectuée sur des vecteurs quantifiés issus des coefficients d'ondelette.

L'idée de recherche progressive dans laquelle on utilise les différents niveaux de résolution lors de la recherche a été proposée par Liang et al. [LK97]. Stark [Sta95] décrit un algorithme qui utilise des réseaux de neurones pour classer différentes textures à l'aide d'attributs issus des ondelettes. Pečenović [Peč98] détaille un algorithme de paquets d'ondelettes qui approxime la transformée de Karhunen-Loève pour indexer et retrouver les images. Do [Do98] propose une méthode de moments maxima d'ondelettes où la décomposition des images est suivie d'une indexation locale selon les moments des coefficients d'ondelettes. Enfin, Chen et al. [CC00] travaillent avec la segmentation d'images couleur à plusieurs niveaux de résolution pour une recherche progressive.

Notre approche, quant à elle, utilise une classification automatique des images afin de construire un arbre visuel de recherche. Elle se rapproche de la technique de Bouman et al. [CB00] qui utilise un arbre de recherche quaternaire pour la navigation. Notre technique de recherche favorise une navigation visuelle facile et rapide à travers la base d'images par un simple clic de souris sur l'image la plus proche de ce que l'utilisateur désire, avec la possibilité de revenir en arrière si l'utilisateur le souhaite.

2.4.3 Les sytèmes de recherche d'images

Une liste de systèmes de recherche d'images a été proposée par Pečenović [PDAV98]. Historiquement, le premier système de recherche est QBIC [FSN+95, FBF+94] d'IBM dans lequel la recherche est basée sur l'indexation des textures des régions des images dans l'espace couleur de Munsell amélioré.

Photobook [PPS96], du MIT Media Lab (*Massachussets Institute of Technology*), propose une recherche possible sur trois critères différents : l'apparence, le contour et la texture. Les deux premiers critères utilisent une décomposition de Karhunen-Loève des régions des images de la base. À partir de cette transformée, les vecteurs propres des images sont utilisés pour la recherche et l'indexation. En ce qui concerne la recherche de texture, elle est basée sur une localisation des pics de fréquence de la transformée de Fourier des images. Ce système est très efficace pour les bases d'images spécialisées.

Dans le système Virage [BFG96], la localisation spatiale des couleurs associée avec la détection de texture des régions peut être pondérée pour affiner les résultats d'une recherche. Le système BlobWorld [CTB+99] de l'université de Berkeley en Californie est disponible en ligne [26] et travaille sur des régions homogènes issues de l'image en procédant à une recherche par région exemple.

Cortina [QMTM04] utilise des descripteurs issus de la norme MPEG-7 et des mots issus du texte autour des images dans les pages web pour construire son index d'images. Le regroupement

d'images est réalisé avec l'algorithme des k plus proches voisins. La requête est soit une requête par mot-clé, soit une requête par image exemple.

Ikona est un système de recherche développé à l'INRIA [30]. Il est basé sur une recherche par image exemple utilisant couleur, texture et forme et un système de bouclage de pertinence qui adapte la recherche suivant les modifications apportées aux résultats par l'utilisateur au cours de la recherche.

Kiwi [14] est un système développé à l'INSA de Lyon. Il est basé sur une analyse des images et l'extraction de points d'intérêts multirésolution des images.

Le système RETIN [FCPF01, Fou02] a été développé à l'ENSEA de Cergy-Pontoise [27], il utilise des attributs de couleur dans l'espace Lab et des attributs de texture (filtres de Gabor). Sa principale force est son système très efficace de bouclage de pertinence.

WINDSURF [ABP99] est basé sur la décomposition en ondelettes des images, suivie par une segmentation des régions à l'aide des nuées dynamiques et par l'extraction d'attributs colorimétriques et de texture. Les régions de l'image requête sont ensuite comparées selon la distance de Mahalanobis pour donner les images les plus proches de la requête.

Tous ces systèmes sont malheureusement difficilement comparables car ils travaillent tous avec une base d'images différente. Ils offrent tous une interface conviviale à l'utilisateur qui doit néanmoins entrer quelques paramètres avant de lancer la recherche.

2.4.4 Les limites

Les résultats présentés dans les articles de recherche du domaine sont difficiles à interpréter pour deux raisons. Premièrement, ni les bases de données utilisées pour tester les techniques d'indexation et de recherche, ni les critères de mesure de similarité ne sont les mêmes d'une méthode à une autre. Par conséquent, le résultat final d'une méthode est difficile à comparer à celui donné par d'autres méthodes. Deuxièmement, les méthodes sont testées sur des bases d'images d'une dizaine de milliers d'images qui sont loin, très loin des soixante cinq millions d'images contenues dans la base Corbis [3] ou bien des 880 millions d'images de l'Internet recensées par Google [22] (parmi plus de 8 milliards de pages). Le travail de recherche dans des bases d'images de plus d'un million d'images est une problématique différente avec des contraintes très fortes en matière de temps de calcul et de pertinence des résultats.

Il y a une autre difficulté à surmonter lorsqu'on travaille avec des descripteurs d'images, c'est celle de la dimension de ces vecteurs. En effet, dans la littérature, les descripteurs extraits des images sont la plupart du temps des vecteurs de taille $n > 100$. Ceci pose le problème de la **malédiction de la dimension (*dimensionality curse*)**.

En effet, les espaces de grandes dimensions possèdent des propriétés mathématiques particulières qui affectent le comportement des méthodes manipulant des données dans ces espaces. Ainsi, la notion de distance en dimension deux ou trois n'a rien à voir avec la notion de dis-

tance dans un espace de dimension 100. Dans sa thèse, Sid-Ahmed Berrani [Ber04] donne des exemples surprenants de propriétés (liées en particulier à leur vacuité) des espaces de grandes dimensions.

Ces différentes raisons entraînent qu'il faut tester les méthodes employées sur de grandes bases d'images et qu'il faut réduire la dimension des vecteurs descripteurs pour pouvoir donner un sens à la notion de distance qu'on souhaite utiliser.

2.4.5 Vers un classement des images

Une solution au problème de la taille de la base est d'appliquer la technique «diviser pour mieux régner» (*divide and conquer*) qui consiste dans ce cas précis à diviser la base constituée de millions d'images en regroupements d'images plus petits organisés en familles d'images visuellement similaires.

Ce découpage de la base d'images permet de définir une structure en couches dans lesquelles les images sont regroupées en familles. En terme informatique, cela consiste à construire un arbre (figure 2.13) dont les feuilles sont des images et dont les nœuds représentent des familles d'images.

L'organisation d'une base de descripteurs en familles relève d'une technique de classification, le *clustering* ou regroupement dans laquelle les données sont regroupées en familles selon une certaine distance. Les techniques de regroupement sont détaillées dans la thèse de Sid-Ahmed Berrani [Ber04].

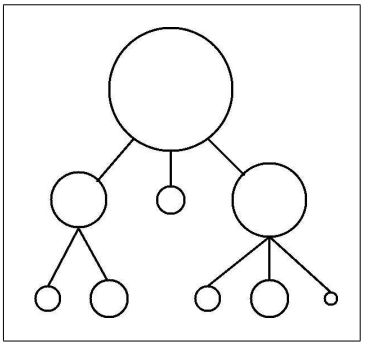


FIG. 2.13 – Arbre de recherche qui découpe la base de données.

Le découpage de la base d'images dans la phase hors-ligne permet de ne travailler que sur un sous-ensemble de la base de données et donc d'appliquer avec une grande chance de succès les méthodes de recherche d'images sur ces «petites» bases.

On peut parcourir l'arbre toujours de la même façon, c'est-à-dire utiliser toujours le même critère de recherche au cours de l'exploration. Toutefois une meilleure approche est de définir en plus de la structure arborescente de la base de données une hiérarchie des signatures.

2.4.6 Méthodes basées sur la navigation

La navigation dans les bases d'images semble être une bonne solution pour simplifier au maximum le travail de l'utilisateur de la base.

Le travail présenté dans la thèse de Yossi Rubner [Rub99] porte sur la navigation dans une base d'images organisée à l'aide de la technique du *multidimensional scaling* (MDS) en réponse à une requête par description de l'utilisateur. Il utilise la distance EMD (*Earth Mover Distance*) pour évaluer les similarités des distributions spatiales de couleur dans la base d'images et regroupe les images selon ce critère. Le résultat de la requête est présenté à l'utilisateur en trois dimensions avec possibilité de naviguer dans les réponses proposées. Le principal inconvénient est que l'utilisateur doit formuler sa requête, il n'y a pas d'organisation automatique de la base.

Une méthode de classification des images en utilisant les nuées dynamiques est utilisée par Pečenović et al. dans leur système CIRCUS [PDVP00]. Ce système autorise la navigation ainsi que la recherche par image exemple. Bertrand Le Saux [LS03] utilise une version améliorée de l'algorithme d'agglomération compétitive afin de classer les image en familles pour la navigation.

Une autre approche consiste à travailler sur l'interface utilisateur pour accéder aux images. Chen et al. [CB00] ont proposé un schéma de décomposition de la base d'images en familles et sous-familles afin d'autoriser la navigation visuelle. Ils utilisent l'algorithme des nuées dynamiques pour classer une base d'images en quatre familles à chaque niveau de l'arbre. Ils construisent un arbre quaternaire ou *quadtree* qui sert de base pour la recherche d'images. C'est leur technique de décomposition qui se rapproche le plus de la méthode que nous allons proposer si ce n'est que leur technique n'utilise pas les ondelettes, est limitée au *quadtree* et ne prend pas en compte le contexte de recherche de l'utilisateur.

Notre contribution principale se décompose en plusieurs étapes :

1. la création et l'utilisation d'une **hiérarchie de signatures multirésolution** de taille **croissante**,
2. la génération d'un **arbre de recherche flou** par le **regroupement automatique d'images en familles d'images visuellement similaires**,
3. la conception d'une **interface de recherche d'images** simple et efficace.

Dans le chapitre suivant, nous allons étudier la notion de signal et d'image et décrire les outils mathématiques dont nous disposons pour extraire des attributs représentatifs à partir des images.

Chapitre 3

Signaux et images

Comme on l'a vu précédemment, il est trop coûteux en temps de calcul et en occupation mémoire de travailler directement sur les images. Il est donc nécessaire d'extraire de l'image des attributs caractéristiques en utilisant des techniques de transformation d'images. Ce chapitre décrit ce qu'est une image numérique et les principales transformations utilisées en traitement d'images. L'analyse d'images [BCC+95, SHB99] est un domaine très vaste que nous allons présenter de façon succinte.

3.1 Signaux, images, statistiques

3.1.1 Signaux

La communication permet l'échange de l'information entre une source (l'émetteur) et une destination (le récepteur) par un canal de communication. Cette information est codée par un signal. Un signal est donc un phénomène physique mesurable contenant de l'information. La théorie du traitement du signal [Fla93, Tru98b, Rei95a, Rei95b, DQ96d, DQ96c, DQ96b, DQ96a] a beaucoup évolué ces dernières années en raison des besoins de transmission et de compression des données audios et vidéos. Cependant, cette théorie est assez ancienne car les bases de la théorie de l'information ont été posées par le mathématicien Claude Elwood Shannon à partir de 1948 [Sha48]. Selon cette théorie, l'information est une baisse de l'incertitude sur une mesure [Sch95].

Dans la majorité des cas, les signaux réels sont des fonctions du temps et sont issus de

capteurs physiques (capteur de température, de vitesse, de pression...). de nombreux signaux dépendent également de variables d'espace. Ainsi, les signaux peuvent être mono ou multi-dimensionnels en fonction de la grandeur qu'ils représentent.

Les signaux portent de l'information, qu'ils soient naturels ou créés artificiellement. Il existe deux grandes familles de signaux.

La première famille est constituée par les signaux **déterministes**, c'est-à-dire les signaux reproductibles lors d'expériences dont les propriétés sont parfaitement connues.

La seconde famille est celle des signaux **aléatoires**, ces signaux ne sont pas reproductibles, on ne peut qu'obtenir des réalisations d'un processus lors d'expériences. Ces réalisations ne permettent pas de caractériser le signal, il faudrait procéder à une infinité de mesures pour définir parfaitement le signal. Le comportement du signal aléatoire est décrit par des données statistiques estimées à partir des réalisations.

En traitement du signal, on distingue les signaux **continus** définis en chaque point de l'espace de mesure des signaux **discrets** définis seulement sur une grille discrète de valeurs. Les signaux numériques sont des signaux échantillonnés et quantifiés. La figure 3.1 montre un signal continu et le signal numérique correspondant. Dans les images, les valeurs des niveaux de gris sont définies seulement sur une grille discrète et sont quantifiées. Les images sont des signaux bi-dimensionnels numériques. Les méthodes d'analyse des images sont des techniques numériques [Tru98b].

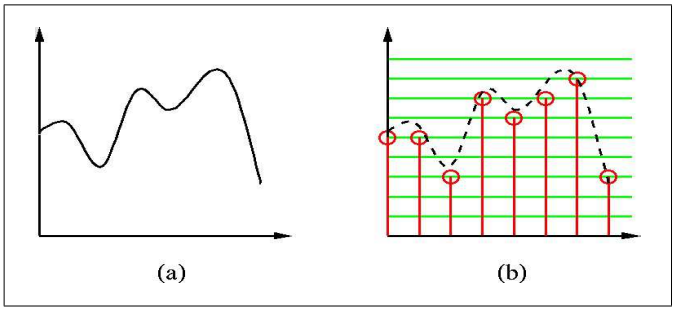


FIG. 3.1 – (a) signal continu, (b) signal numérique.

Les signaux qui apportent de l'information sont les signaux qui varient au cours du temps. En traitement du signal, on va donc chercher à détecter les transitions (singularités) qui vont apporter des informations sur le signal étudié.

En plus de la représentation spatiale ou temporelle du signal, il existe une représentation fréquentielle. Tout signal peut être décomposé à l'aide de la transformée de Fourier en composantes sinusoïdales caractérisées par leur fréquence et leur amplitude. Les **basses fréquences** correspondent à l'allure générale du signal (variations lentes), tandis que les **hautes fréquences** sont caractéristiques des détails du signal (variations rapides).

3.1.2 Images

3.1.2.1 Représentation des images numériques

Les images informatiques sont des signaux numériques dépendant de deux variables (les deux dimensions du plan). Une image peut être interprétée comme la variation dans un espace à deux dimensions d'une information lumineuse. La figure 3.2 donne deux représentations possibles de la même image, la représentation à deux dimensions (2D) classique et la représentation spatiale en trois dimensions : la largeur, la hauteur et la variation de la luminosité (ou niveau de gris).

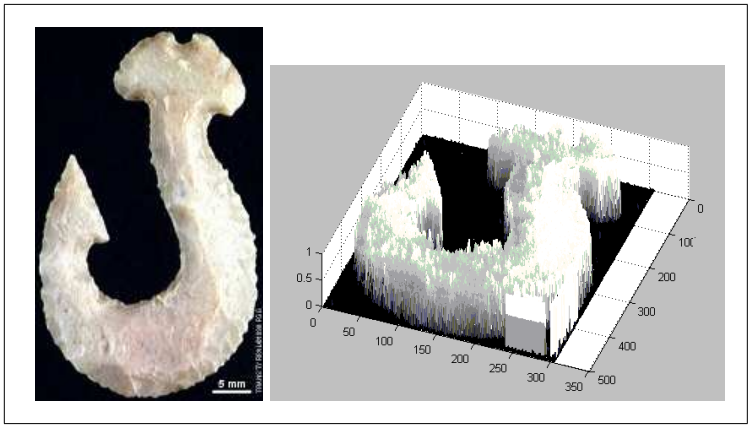


FIG. 3.2 – Une image et sa représentation spatiale en terme de variation de luminosité.

Pour coder les images couleur, on utilise en général trois plans en niveaux de gris : rouge, vert et bleu dans lesquels les couleurs sont des quantités numériques. Ainsi, il faut trois fois plus d'informations pour coder une image en couleurs qu'une image en niveaux de gris. La figure 3.3 donne la représentation numérique d'une image en niveaux de gris et d'une image couleur. L'image en niveaux de gris n'utilise qu'un plan pour stocker l'information de luminosité.

L'image couleur, quant à elle, utilise trois plans, un rouge (R), un vert (V) et un bleu (B) pour stocker l'information couleur de l'image. Pour des contraintes informatiques de stockage mémoire, on code souvent un niveau de gris sur un octet (soit huit bits), ce qui équivaut donc à une dynamique de $2^8 = 256$ niveaux de gris possibles. Pour une image couleur, on code souvent chaque plan couleur avec un octet, ce qui équivaut $(2^8)^3 = 16\,777\,216$ couleurs possibles.

3.1.2.2 Régions des images

Afin d'analyser la scène contenue dans une image, une méthode classique consiste en la recherche des différentes régions qui composent l'image. Dans la plupart des cas, ces régions

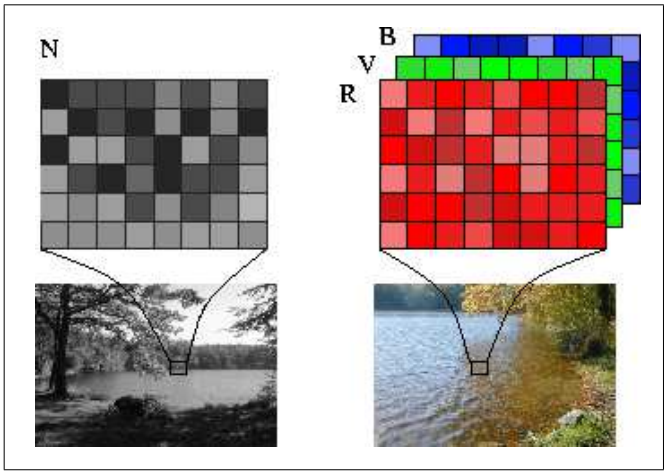


FIG. 3.3 – Exemple d'image en niveaux de gris et d'image couleur.

correspondent aux objets présents dans la scène. La recherche des régions qui composent une image est la **segmentation** [BCC+95].

On distingue la segmentation **approche contour** de la segmentation **approche région**. Dans la première, on cherche à déterminer les frontières des régions de l'image, alors que dans la seconde, on cherche directement les régions présentes dans l'image. Ces deux approches ne donnent pas forcément la même partition de l'image et les résultats dépendent très largement de la méthode utilisée.

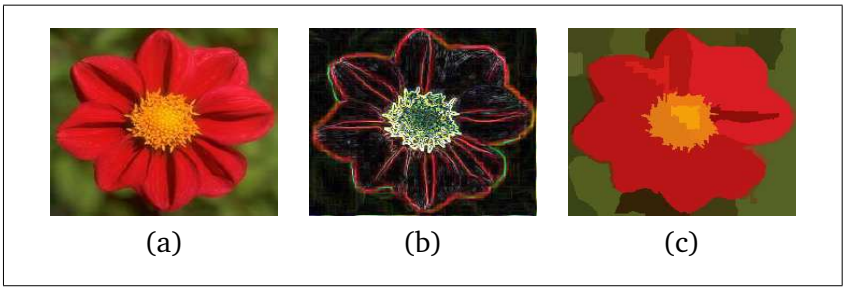


FIG. 3.4 – Illustration de la segmentation d'une image. (a) image de départ, (b) détection des contours, (c) segmentation en régions.

Dans l'approche contour sont utilisées des méthodes dérivatives, surfaciques, morphologiques... Tandis que dans l'approche région, on utilise des méthodes structurelles telles que la croissance de régions ou les algorithmes de type ligne de partage des eaux (*watershed*). La figure 3.4 donne un exemple de segmentation d'une image.

La segmentation en régions d'une image permet de décomposer l'image en zones homogènes afin de calculer les attributs sur des régions et non sur l'image tout entière. Grâce à la

segmentation, on peut essayer de retrouver des objets dans une scène.

3.1.3 Notions de statistiques

L'analyse statistique des données est très importante dans le traitement des signaux car la famille des signaux aléatoires ne peut être étudiée que par rapport aux statistiques extraites de ces signaux. Nous allons rappeler les quelques définitions statistiques de base que nous utiliserons par la suite.

La **moyenne** d'un signal X de taille p représente la moyenne des valeurs prises par les échantillons du signal.

$$\bar{X} = \frac{1}{p}\sum_{i=0}^{p-1} X_i$$

La **variance** du signal donne la moyenne des carrés des variations des échantillons du signal par rapport à sa moyenne.

$$\sigma_X^2 = \frac{1}{p-1}\sum_{i=0}^{p-1}(X_i - \bar{X})^2$$

L'**écart-type** est la racine carrée de la variance. L'écart type indique comment, en moyenne, les valeurs de la variable sont groupées autour de la tendance centrale (moyenne arithmétique). Un faible écart type signifie que les valeurs sont peu dispersées autour de la moyenne (série homogène), et inversement (série hétérogène).

La **covariance** de deux variables réelles permet de mesurer la dépendance linéaire des variables, c'est-à-dire la façon dont deux variables X et Y varient simultanément. Globalement, si X croît (respectivement décroît) quand Y croît (respectivement décroît), la covariance est positive.

$$cov_{XY} = \frac{1}{p-1}\sum_{i=0}^{p-1}(X_i - \bar{X})(Y_i - \bar{Y})$$

Toutefois, la covariance entre deux variables est une grandeur qui possède une unité, on lui préfère donc une grandeur sans unité, le coefficient de corrélation. Le **coefficient de corrélation** entre deux signaux X et Y définit le niveau de corrélation/dépendance entre deux variables. Il est compris entre 0 et 1. S'il est proche de 1, les variables sont corrélées, s'il est nul, les variables sont décorrélées.

$$corr_{XY} = \frac{cov_{XY}}{\sigma_X \sigma_Y}$$

3.2 Temps, fréquence, espace, échelle

Un outil mathématique utilisé depuis très longtemps en traitement du signal est la transformée de Fourier que nous allons présenter en en précisant quelques avantages et quelques inconvénients. Puis l'analyse en ondelettes que nous utilisons dans ce travail sera présentée en détails.

3.2.1 La transformée de Fourier

Le mathématicien Joseph Fourier a introduit en 1822 l'idée que toute fonction périodique peut être décomposée en une somme de sinus et de cosinus. Cet énoncé qui semble simple au premier abord a bouleversé les mathématiques.

La transformée de Fourier est un outil mathématique qui permet de projeter un signal sur une base de fonctions sinus et cosinus de différentes fréquences. Le signal transformé est la représentation fréquentielle du signal de départ.

La notation exponentielle complexe permet grâce à la formule de De Moivre $e^{i\theta} = cos\,\theta + i\,sin\,\theta$ d'exprimer plus simplement la transformée de Fourier. Par définition, la transformée de Fourier d'une fonction continue $f(t) \in L^1(\mathbb{R})$ est :

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)\, e^{-i\omega t} dt \tag{3.1}$$

Où $\omega = 2\pi\nu$ est la pulsation et ν la fréquence.

La caractéristique primordiale de cette transformée est sa réversibilité (sous certaines conditions), ainsi on a :

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega)\, e^{i\omega t} d\omega \tag{3.2}$$

Une autre propriété très intéressante est l'équivalence entre convolution dans l'espace de départ (temporel ou fréquentiel) et multiplication dans l'espace d'arrivée (théorème de Plancherel). Ainsi, on a :

$$f(t) \star g(t) = \hat{f}(\omega).\hat{g}(\omega) \text{ et } f(t).g(t) = \frac{1}{2\pi}\hat{f}(\omega) \star \hat{g}(\omega)$$

Le passage à une transformée bi-dimensionnelle est donné par l'équation 3.3 ci-dessous :

$$\hat{f}(\omega_1, \omega_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(t_1, t_2)\, e^{-i(\omega_1 t_1 + \omega_2 t_2)} dt_1 dt_2 \tag{3.3}$$

La transformée de Fourier discrète permet d'appliquer la transformée de Fourier aux signaux numériques selon l'équation 3.4. Pour toute série numérique $s(n)$ à N éléments, sa transformée de Fourier discrète $S(k)$ est définie par :

$$S(k) = \frac{1}{N} \sum_{n=0}^{N-1} s(n).e^{-2i\pi k \frac{n}{N}} \tag{3.4}$$

En deux dimensions, cette équation devient :

$$S(k,l) = \frac{1}{M}\frac{1}{N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} s(m,n).e^{-2i\pi k \frac{m}{M}}.e^{-2i\pi l \frac{n}{N}} \tag{3.5}$$

La figure 3.5 illustre la transformée de Fourier d'une image (signal bi-dimensionnel). L'image originale (a) a été transformée en l'image (b) par un filtre de lissage (moyenneur). Ainsi l'image (b) est floue, elle contient donc moins de détails. Ces détails perdus apparaissent clairement sur l'image (c) qui est la différence entre l'image (a) et l'image (b). On traduit cela dans l'espace de Fourier. L'image (d) contient des informations fréquentielles réparties sur tout l'espace (ou presque). Les basses fréquences de l'image se trouvent au milieu alors que les hautes fréquences se trouvent en périphérie. L'image (e) contient clairement moins de hautes fréquences, l'essentiel des points se situent au centre. L'image (f), différence de (d) et (e) confirme que les hautes fréquences sont perdues lors d'un lissage moyenneur.

On peut utiliser la transformée de Fourier pour extraire des informations fréquentielles d'une image, toutefois le principal problème de la transformée de Fourier est son manque de résolution temporelle. Cela signifie simplement que si on est effectivement capable de détecter toutes les fréquences qui apparaissent dans un signal, on est en revanche incapable de déterminer à quel moment elles se produisent dans le signal. Car si on est en mesure d'interpréter le module de la transformée, il en est tout autrement de la phase qui est difficile à analyser. Il existe une transformée de Fourier plus «locale» donnant des informations mieux localisées. Nous allons l'étudier ci-dessous.

3.2.2 Transformée de Fourier à fenêtre glissante

Pour pallier ce problème de manque de résolution, Gabor [Gab46] a introduit la transformée de Fourier à fenêtre glissante. Dans cette méthode, on considère le signal à analyser comme appartenant à une fenêtre de longueur fixe qui glisse sur le signal pendant la transformation.

$$\hat{\hat{f}}(s,\omega) = \int_{-\infty}^{+\infty} f(t)g(t-s)e^{-i\omega t}dt \tag{3.6}$$

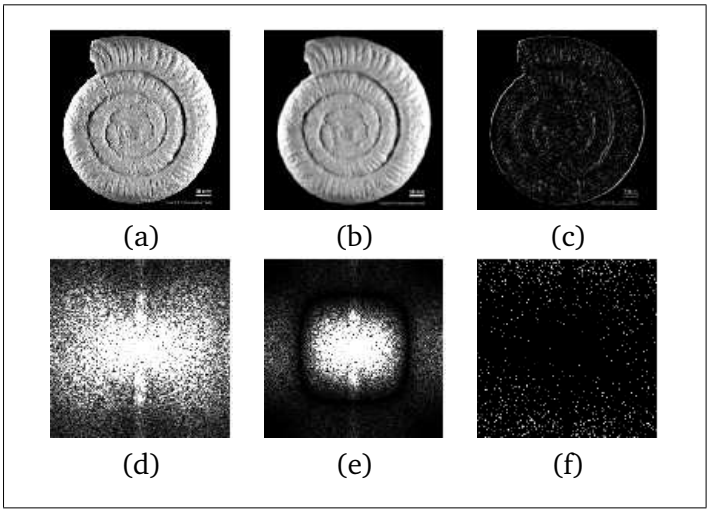


FIG. 3.5 – Illustration de la transformée de Fourier. La tranformée de Fourier d'une image laisse apparaître une concentration de basses fréquences autour du centre de l'image et des hautes fréquences dans la périphérie de l'image. (a) image de départ ; (d) transformée de Fourier de cette image ; (b) image lissée (moyennée) ; (e) transformée de Fourier de cette image moyennée ; (c) image différence entre l'image de base et l'image lissée et (f) transformée de Fourier de l'image différence.

$g(t)$ est la fonction fenêtre (Gabor avait choisi une fenêtre gaussienne) qui permet une localisation spatiale du spectre fréquentiel fourni par la transformée de Fourier. Cette approche permet d'avoir une meilleure résolution spatiale que la transformée de Fourier, mais la taille fixe de la fenêtre est un gros inconvénient. L'outil idéal serait une fenêtre qui s'adapte aux variations de fréquence dans le signal à analyser. Cet outil existe, il s'agit de la récente (années 1990) analyse en ondelettes [Tru98a].

3.2.3 Les ondelettes

La transformée en ondelettes [BH95, Mal99] est un outil mathématique récent qui décompose un signal en fréquences en conservant une localisation spatiale. Le signal de départ est projeté sur un ensemble de fonctions de base qui varient en fréquence et en espace. Ces fonctions de base s'adaptent aux fréquences du signal à analyser. Cette transformation permet donc d'avoir une localisation en temps et en fréquence du signal analysé.

La figure 3.6 donne le pavage de l'espace temps-fréquence pour la transformée de Fourier à fenêtre et pour l'analyse en ondelettes. L'inégalité d'Heisenberg assure que $\sigma_\omega.\sigma_t \geq 1/2$. Le produit $\sigma_\omega.\sigma_t$ définit l'aire de la fenêtre d'analyse du signal. Pour l'analyse de Fourier, la fenêtre est de taille constante, pour l'analyse en ondelettes, la fenêtre est de surface constante mais sa taille varie en fonction de la fréquence à analyser.

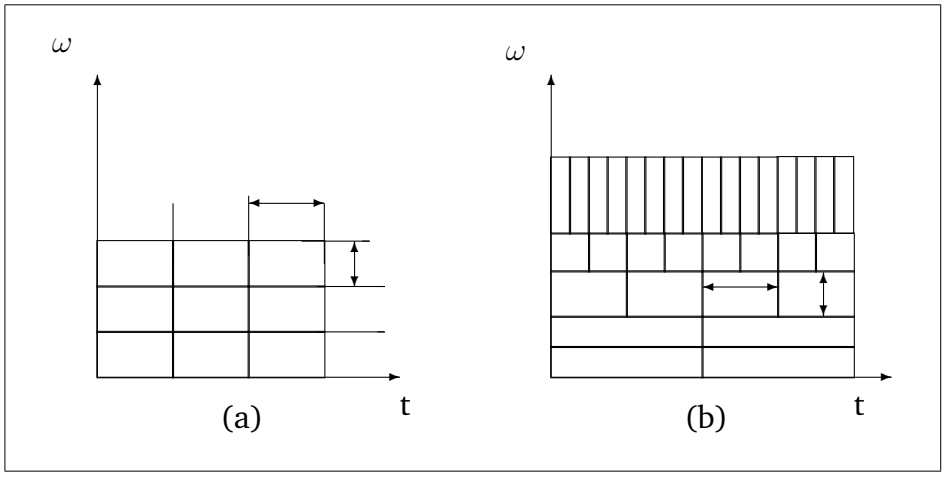


FIG. 3.6 – Pavage du plan temps-fréquence. (a) Transformée de Fourier, (b) transformée en ondelettes.

Les ondelettes ont d'abord été introduites par Grossman et Morlet [GM84] comme un outil mathématique d'analyse de signaux sismiques. Ensuite la théorie s'est développée sous la conduite de nombreux contributeurs [LM86, Bat87, Mal89, Dau90, Mey90, RV91, CDF92]. Aujourd'hui, les applications des ondelettes sont nombreuses [20] : compression [Gou02, Par03], filtrage, débruitage, résolution d'équations aux dérivées partielles...

La transformation en ondelettes continue est définie par :

$$\tilde{f}(a,b) = \int_{-\infty}^{+\infty} f(t)\psi^*_{a,b}(t)dt \tag{3.7}$$

Dans cette équation, $\psi^*(t)$ est l'ensemble des fonctions de base appelées ondelettes. L'étoile «*» représente le complexe conjugué. La famille d'ondelettes est générée à partir de la dilatation et de la translation d'une ondelette mère $\psi(t)$. Les ondelettes sont donc définies par :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}) \tag{3.8}$$

Où b est le **facteur de translation** et a le **facteur de dilatation** de l'ondelette mère. Les équations 3.7 et 3.8 donnent :

$$\tilde{f}(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} f(t)\psi^*(\frac{t-b}{a})dt \tag{3.9}$$

Où ψ désigne la fonction ondelette mère dont les dilatées et translatées sont les bases d'un espace d'analyse en ondelettes sur lequel est projetée la fonction $f(t)$.

La transformée en ondelettes est réversible, on peut donc passer de l'analyse d'une fonction à sa reconstruction, à un éventuel facteur de normalisation près, par :

$$f(t) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \tilde{f}(a,b)\psi_{a,b}(t)\frac{da}{a^2}db$$

Remarquons que contrairement à la transformée de Fourier, la fonction ψ n'est pas imposée. Par contre, ψ doit posséder certaines caractéristiques pour que la transformée en ondelettes soit inversible. Toute fonction qui vérifie la condition d'admissibilité ci-dessous est une ondelette possible :

$$\int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|}d\omega < +\infty\,, \quad \psi \in L^2(\mathbb{R})$$

C'est-à-dire que l'ondelette est à énergie finie. De plus, on a $|\hat{\psi}(0)|^2 = 0$, autrement dit la fonction ψ est un filtre passe-bande. Cela implique que $\int_{-\infty}^{+\infty} \psi(t)dt = 0$ donc ψ est de moyenne nulle, elle doit osciller, ψ est une onde, d'où le terme d'ondelette. En d'autres termes, l'ondelette doit être localisée en temps et en fréquence. La localisation fréquentielle de l'ondelette étant parfois approximative, on parle souvent plutôt d'**analyse temps-échelle**.

Il existe une infinité de fonctions d'ondelettes ψ qui répondent aux critères énoncés ci-dessus. Il existe donc de nombreuses familles d'ondelettes ayant chacune leurs propriétés. L'ondelette la plus simple est l'ondelette de Haar qui va servir à illustrer l'exemple ci-dessous.

En figure 3.7, on voit un signal représentant trois notes de musique (do, ré et mi) jouées à la suite les unes des autres. La courbe en haut représente le signal temporel où on retrouve nos trois notes séparées par un court silence. Si on a bien la représentation temporelle, on ne sait pas quelles notes sont jouées.

À gauche on trouve la représentation fréquentielle du signal. On se rend compte que les fréquences sont bien localisées, on peut connaître les notes jouées (do, ré et mi). Par contre, on ne sait pas à quelle moment elles sont jouées. Seule la représentation temps/fréquence centrale permet de savoir quelle note est jouée et à quel moment. On obtient avec la représentation temps/fréquence des informations dans les deux espaces en même temps ce qui présente un énorme intérêt en traitement d'images.

3.2.4 La transformée en ondelettes discrète

La transformée en ondelettes continue est très redondante. Afin d'appliquer efficacement la transformée en ondelettes aux signaux discrets, il convient de discrétiser les coefficients de dilatation a et de translation b.

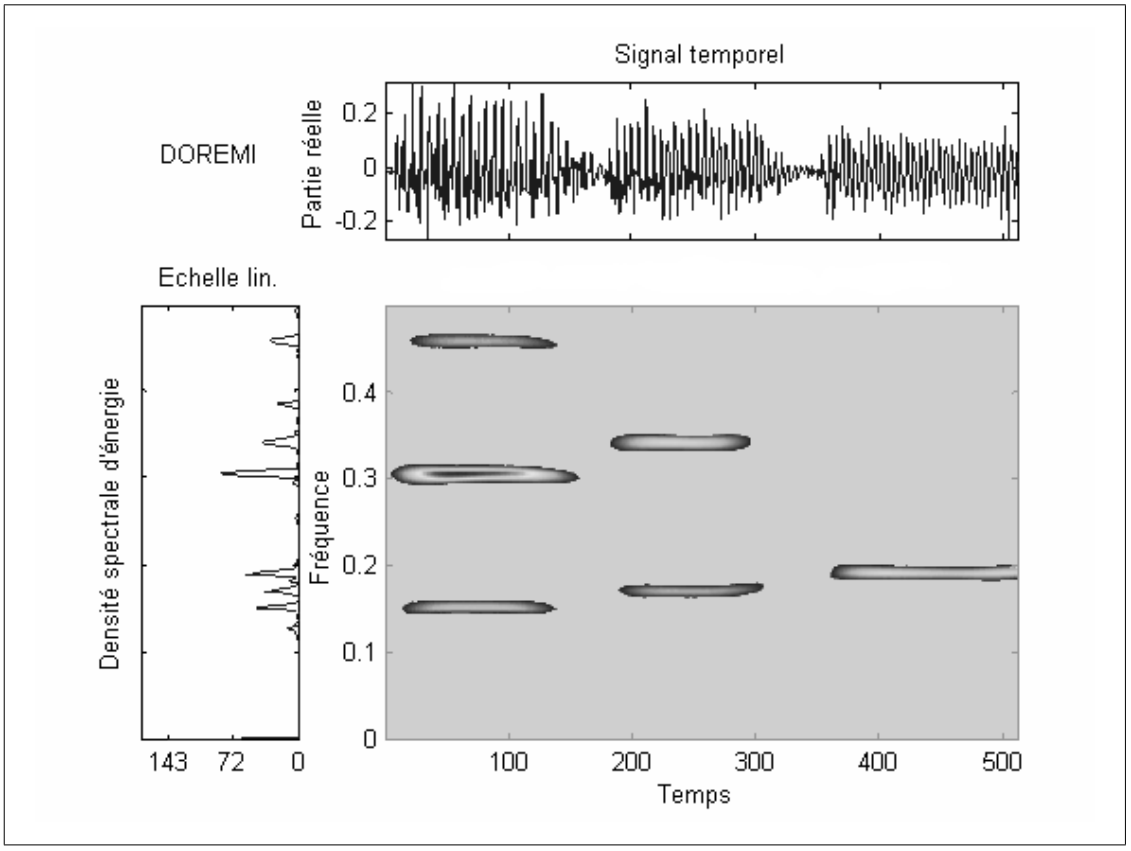


FIG. 3.7 – Exemple d'utilisation de la représentation temps/fréquence (d'après une présentation de Frédéric Truchetet).

On impose donc une grille de valeurs discrètes pour a et b. On pose $a = a_0^m$ et $b = nb_0a_0^m$ avec $a_0 \in \mathbb{Z}$ et $b_0 \in \mathbb{Z}$.

La transformée en ondelettes discrète est donnée par :

$$\tilde{f}(m,n) = a_0^{-\frac{m}{2}} \int_{-\infty}^{+\infty} f(t)\,\psi(a_0^{-m}t - nb_0)dt \tag{3.10}$$

Si on choisit $a_0 = 2$ et $b_0 = 1$, on se place dans le cas **dyadique**. On a alors :

$$\tilde{f}(m,n) = 2^{-\frac{m}{2}} \int_{-\infty}^{+\infty} f(t)\,\psi(2^{-m}t - n)dt \tag{3.11}$$

La figure 3.8 donne un exemple de décomposition du signal sinusoïdal bruité sur une base d'ondelettes avec une échelle dyadique. Le signal de base possède 1000 échantillons. Le nombre d'échantillons est divisé par deux à chaque échelle. On retrouve dans le niveau d'échelle le plus

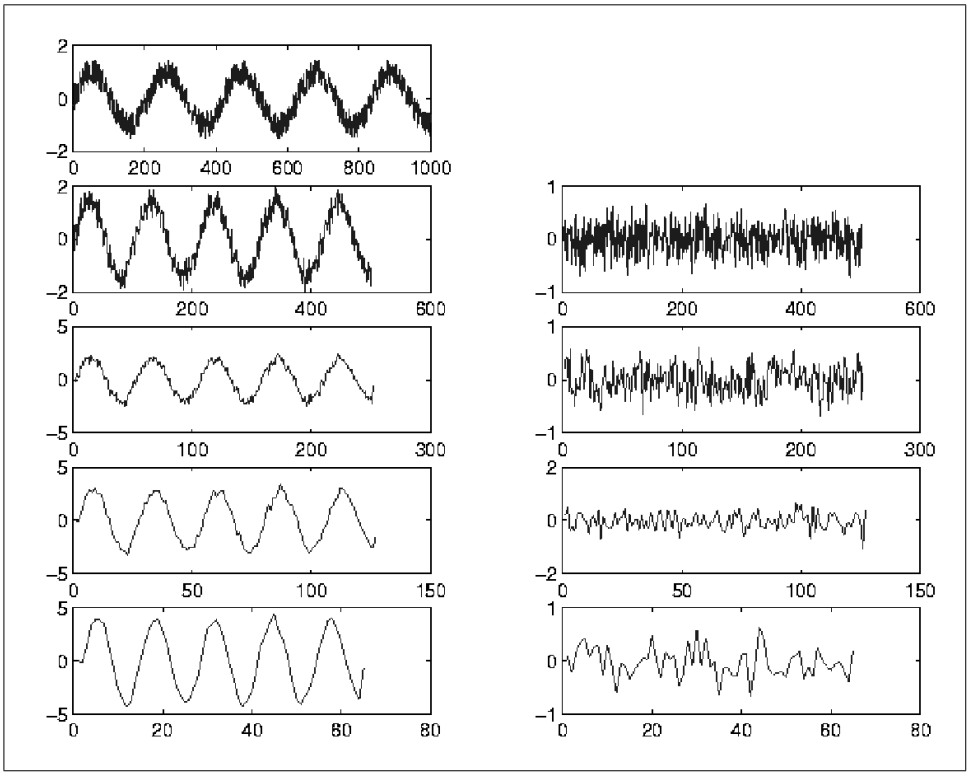


FIG. 3.8 – Transformée en ondelettes d'un signal 1D à plusieurs niveaux de résolution. La colonne de gauche donne le signal d'approximation, celle de droite le signal des détails à chaque niveau de résolution.

bas l'aspect du signal débruité (sans les hautes fréquences qui se trouvent dans le signal de détails). Dans la suite de cette thèse, on considère toujours qu'on se trouve dans le cas dyadique à deux dimensions.

3.2.5 Les familles d'ondelettes

Il existe une infinité de fonctions d'ondelettes parce que toute fonction oscillante localisée est une ondelette mère possible. Toutefois, elles ne possèdent pas toutes des propriétés intéressantes. Aussi, de nombreux spécialistes des ondelettes ont construit des familles d'ondelettes possédant certaines propriétés remarquables.

D'après le principe d'inégalité d'Heisenberg $\Delta t \, \Delta\omega \geq \frac{1}{2}$, on ne peut pas à la fois localiser un signal en temps et en fréquence. Quand on améliore la localisation dans un des deux espaces, c'est au détriment de l'autre.

Parmi les familles d'ondelettes, les ondelettes de Haar sont les plus simples, mais elles ne sont pas bien localisées. Ingrid Daubechies a construit des ondelettes à support compact qui permettent d'utiliser des filtres de taille finie. Une autre famille d'ondelettes est la famille des

ondelettes splines dont la réponse fréquentielle est bien localisée. Les différentes familles d'ondelettes sont utilisées selon leurs propriétés en fonction du problème à résoudre. Dans notre démonstrateur logiciel, nous avons utilisé des ondelettes en nombres entiers décrites au paragraphe 3.3.4.

3.2.6 L'analyse multirésolution

L'analyse multirésolution, introduite par Meyer et Mallat [Mal89], est un outil de traitement du signal qui permet de décomposer un signal à plusieurs échelles (résolutions) et de le reconstruire à partir des éléments de cette décomposition.

Une analyse multirésolution est un partitionnement de l'espace des fonctions d'énergie finie $L^2(\mathbb{R})$ par une famille de sous-espaces vectoriels V_j emboîtés les uns dans les autres tels que le passage de l'un à l'autre soit le résultat d'un changement d'échelle. Ces sous-espaces sont appelés des espaces d'approximation à l'échelle j ($j \in \mathbb{Z}$) et vérifient les propriétés suivantes :

Soit un ensemble de sous-espaces de $L^2(\mathbb{R})$ (l'ensemble des signaux à énergie finie) tels que :

$\ldots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset \ldots \subset V_{j+1} \subset V_j \subset \ldots$

$\overline{\cup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$

$\cap_{j\in\mathbb{Z}} V_j = 0$

$\forall j \in \mathbb{Z}, f(x) \in V_j \Longleftrightarrow f(2^{-1}x) \in V_{j+1}$

$\forall k \in \mathbb{Z}, f(x) \in V_0 \Longleftrightarrow f(x-k) \in V_0$

Ces propriétés définissent une analyse multirésolution dyadique sur $L^2(\mathbb{R})$.

L'analyse multirésolution a été définie par Mallat [Mal89]. L'idée est de projeter un signal $f(t) \in L^2(\mathbb{R})$ appartenant à un espace V_j sur un sous-espace V_{j+1} et un sous-espace W_{j+1} dans le but de réduire la résolution de moitié. Le schéma est donné en figure 3.9. Il existe donc un opérateur de projection A_j et un opérateur de projection D_j qui projettent respectivement le signal $f(t)$ sur V_{j+1} et W_{j+1}. V_{j+1} est le sous-espace d'approximation et W_{j+1} le sous-espace de détails. On peut démontrer qu'il existe une fonction d'échelle $\phi(t) \in L^2(\mathbb{R})$ qui engendre par dilatation et translation une base orthonormée de V_{j+1} et une fonction d'ondelettes $\psi(t) \in L^2(\mathbb{R})$ qui engendre par dilatation et translation une base orthonormée de W_{j+1}. Les espaces obtenus ne sont pas quelconques, ils possèdent des propriétés intéressantes. Par construction, les espaces d'approximation V_{j+1} et de détails W_{j+1} sont complémentaires : $V_j = V_{j+1} \bigoplus W_{j+1}$. De plus, si les bases sont orthogonales, ils sont orthogonaux : $V_{j+1} \perp W_{j+1}$.

Les fonctions de bases dilatées sont données par les relations :

$\phi_{j,n}(t) = 2^{-j/2}\phi(2^{-j}t - n)$ avec $n \in \mathbb{Z}$ et $\psi_{j,n}(t) = 2^{-j/2}\psi(2^{-j}t - n)$ avec $n \in \mathbb{Z}$

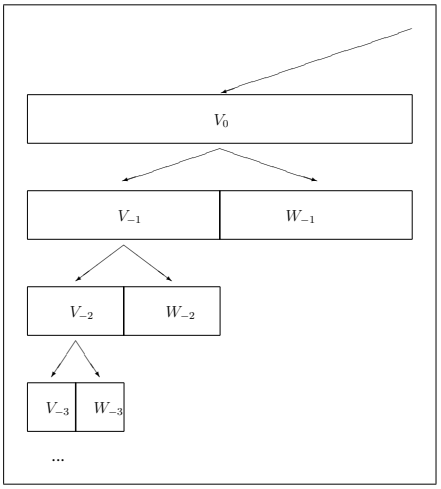


FIG. 3.9 – Principe de l'analyse multirésolution.

On a donc $A_j f = \sum_n < f, \phi_{j,n} > \phi_{j,n}$ et $D_j f = \sum_n < f, \psi_{j,n} > \psi_{j,n}$ où $< f(t),\ g(t) >$ désigne le produit scalaire de $f(t)$ par $g(t)$: $< f(t),\ g(t) >= \int_{-\infty}^{+\infty} f(t)\ g(t)^* dt$

Puisque les signaux analysés sont réels, on a $g(t)^* = g(t)$. On pose $a_{j,n} =< f, \phi_{j,n} >$ et $d_{j,n} =< f, \psi_{j,n} >$. $a_{j,n}$ et $d_{j,n}$ sont respectivement les coefficients d'approximation et de détails de la transformée en ondelettes de la fonction f.

Le résultat de l'analyse multirésolution d'une image est donné en figure 3.11. On voit la diminution de la résolution, l'image d'approximation et les images de détails horizontaux, verticaux et diagonaux.

3.2.7 Algorithmes d'analyse et de synthèse de Mallat

Stéphane Mallat a donné un algorithme d'analyse (ou décomposition) en ondelettes qui permet d'obtenir une analyse multirésolution du signal. Cet algorithme travaille par filtrage de l'image suivant les lignes puis les colonnes par deux filtres, $\tilde{g}$ passe-haut et $\tilde{h}$ passe-bas. $\tilde{h}$ va permettre de repérer les basses fréquences dans l'image (l'approximation) et $\tilde{g}$ les hautes fréquences (les détails). $\tilde{h}$ et $\tilde{g}$ sont construits à partir des fonctions ψ et ϕ. La figure 3.10 donne le schéma d'analyse (de décomposition) de Mallat.

La reconstruction des signaux analysés est effectuée à l'aide d'un banc de filtres h et g qui sont les filtres conjugués de $\tilde{h}$ et $\tilde{g}$. En fonction de l'ondelette et du type de base (orthogonale ou bi-orthogonale) choisis pour l'analyse, les filtres d'analyse et de synthèse peuvent être de même taille, symétriques ou bien de taille différente, non symétriques. La figure 3.12 donne le schéma de synthèse (de reconstruction) de Mallat.

Dans la reconstruction, on travaille alternativement sur les colonnes puis sur les lignes lorsque les ondelettes sont séparables. Il existe un autre algorithme de transformation en onde-

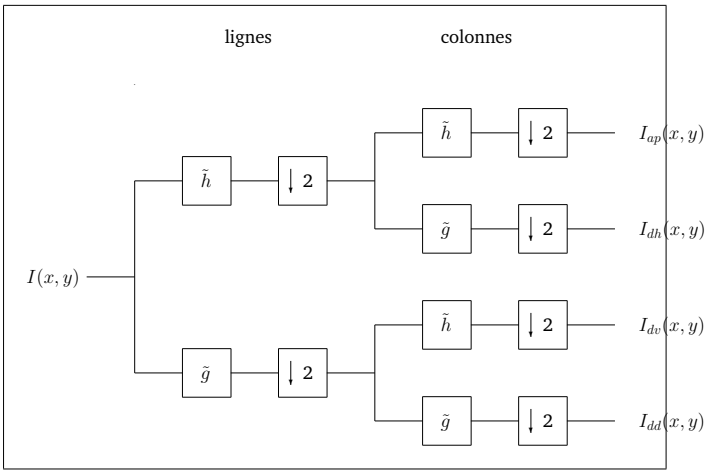


FIG. 3.10 – Algorithme d'analyse de Mallat.

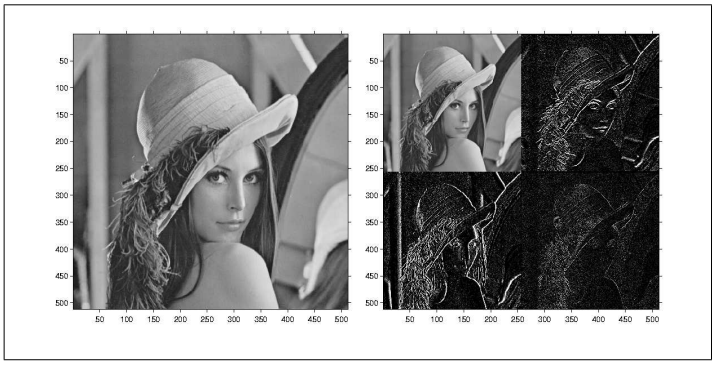


FIG. 3.11 – Un exemple de décomposition en ondelettes d'une image au premier niveau de résolution.

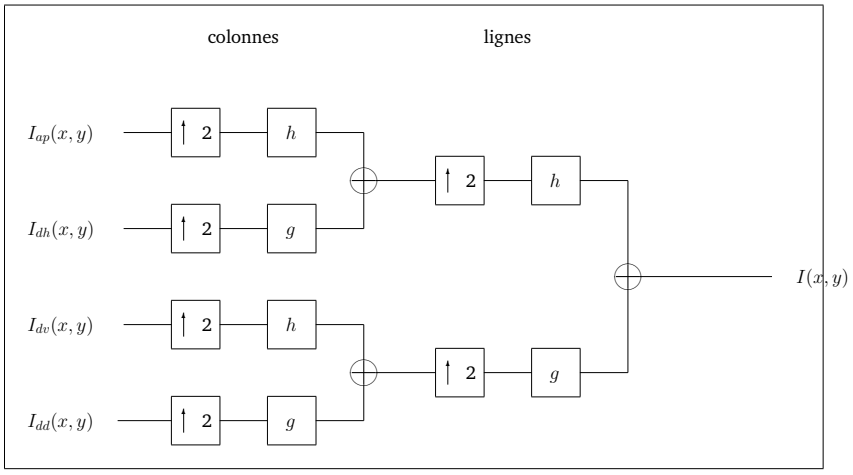


FIG. 3.12 – Algorithme de reconstruction de Mallat.

lettes : le *lifting scheme*.

3.3 *Le lifting scheme*

Le *lifting scheme* est une construction de transformées en ondelettes non basée sur l'analyse de Fourier. Elle a été présentée par Wim Sweldens [Swe95] et est parfois qualifiée d'ondelettes de seconde génération.

Le *lifting scheme* fonctionne comme l'algorithme de décomposition de Mallat vu précédemment : à partir d'un signal de longueur $2n$, on veut obtenir un signal d'approximation et un signal de détails, tous deux de longueur n.

3.3.1 Algorithme d'analyse

L'algorithme d'analyse du *lifting scheme* est donné en figure 3.13. Il se décompose en trois étapes :

– Séparation (*Split*) : Cette étape consiste à découper le signal à analyser s_j en deux signaux, d'une part un signal $even_{j-1}$ (ou s_{2l}) contenant les échantillons pairs et d'autre part un signal odd_{j-1} (ou s_{2l+1}) contenant les échantillons impairs. La taille de chacun des signaux obtenus est égale à la moitié de la taille d'origine. Cette étape est appelée transformation en ondelettes paresseuses (Lazy wavelet transform) :

$$(odd_{j-1}, even_{j-1}) := Split(s_j) \tag{3.12}$$

– Prédiction (*Predict*) : Considérant que les signaux pairs et impairs sont corrélés, on peut estimer l'un d'eux en connaissant l'autre. On définit donc un opérateur de prédiction noté P permettant de prédire les échantillons impairs à partir des échantillons pairs, on a :

$$d_{j-1} = odd_{j-1} - P(even_{j-1}) \tag{3.13}$$

– Mise à jour (*Update*) : Le calcul de la prédiction précédente introduit des erreurs dans le signal (car la prédiction n'est pas exacte). Il convient donc de mettre à jour le résultat précédent grâce à un opérateur de mise à jour U.

$$s_{j-1} = even_{j-1} + U(odd_{j-1}) \tag{3.14}$$

À l'issue des itérations sur le signal de départ, s_{j-1} est le signal d'approximation de s_j à l'échelle $j-1$ et d_{j-1} est le signal de détails à l'échelle $j-1$.

En résumé, on a l'algorithme d'analyse suivant :

$$\begin{array}{rcl} (odd_{j-1}, even_{j-1}) & := & Split(s_j) \\ odd_{j-1} & -= & P(even_{j-1}) \\ even_{j-1} & += & U(odd_{j-1}) \end{array}$$

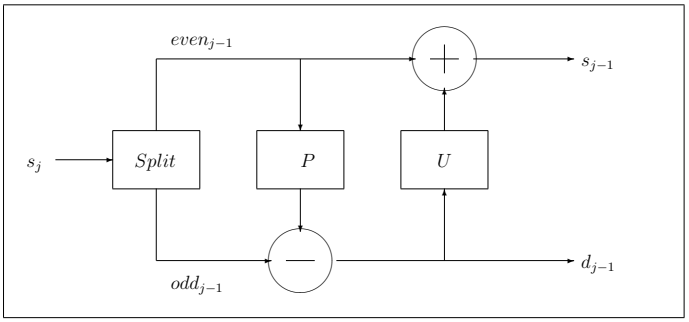


FIG. 3.13 – L'algorithme lifting scheme d'analyse

3.3.2 Algorithme de synthèse

Tout comme les ondelettes classiques, le *lifting scheme* est réversible et son inverse est aussi composé de trois étapes.

– Mise à jour inverse (*Undo update*) : On retrouve le signal pair de départ simplement en inversant le signe de l'opérateur de mise à jour U.

$$even_{j-1} = s_{j-1} - U(odd_{j-1})$$

– Prédiction inverse (*Undo predict*) : Connaissant les échantillons pairs et l'opérateur de prédiction P, il suffit d'inverser le signe de l'opération dans l'analyse :

$$odd_{j-1} = odd_{j-1} + P(even_{j-1})$$

– Regroupement (*Merge*) : On regroupe ensemble les signaux pairs et impairs en les intercalant. Cette étape est la transformation en ondelettes paresseuses inverse (*inverse lazy wavelet transform*).

$$s_j := Merge(odd_{j-1}, even_{j-1})$$

En résumé, la transformée inverse est donnée par le tableau ci-dessous :

$$\begin{array}{rcl} even_{j-1} & -= & U(odd_{j-1}) \\ odd_{j-1} & += & P(even_{j-1}) \\ s_j & := & Merge(odd_{j-1}, even_{j-1}) \end{array}$$

Le schéma de synthèse lifting scheme est donné sur la figure 3.14.

Les caractéristiques de la base d'ondelettes sont fixées par U et P, tout comme $\tilde{h}$ et $\tilde{g}$ fixaient les propriétés des ondelettes dans l'algorithme de Mallat. Les bases d'ondelettes utilisées dans le *lifting scheme* sont en général bi-orthogonales.

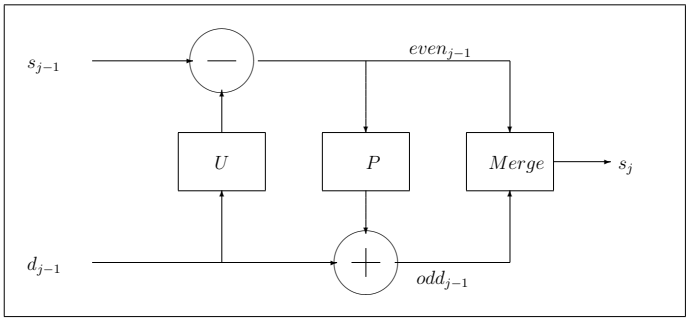


FIG. 3.14 – L'algorithme lifting scheme de synthèse

3.3.3 Les images

Afin d'appliquer la transformée *lifting scheme* aux images, on procède simplement au découpage de l'image $I(i,j)$ en deux images. La première image I_{even} contient les éléments d'ordre pair ($i+j$ est pair) de l'image de départ, la seconde I_{odd} contient les éléments d'ordre impair ($i+j$ est impair).

Ensuite, on applique la transformée sur les lignes (j) puis sur les colonnes (i) des images à l'aide des équations 3.13 et 3.14. Le résultat contient donc l'image d'approximation et les images de détails (horizontaux, verticaux et diagonaux) mélangées qu'il convient de remettre en ordre avant de les exploiter.

		i						
		0	1	2	3	4	5	6
j	0	e	o	e	o	e	o	e
	1	o	e	o	e	o	e	o
	2	e	o	e	o	e	o	e
	3	o	e	o	e	o	e	o
	4	e	o	e	o	e	o	e

FIG. 3.15 – L'algorithme lifting scheme sur une image

3.3.4 Avantages

Le *lifting scheme* d'ondelettes à plusieurs avantages algorithmiques par rapport à la construction classique des ondelettes dans le domaine de Fourier.

– **Mémoire** : Les calculs du *lifting scheme* peuvent être réalisés en place, c'est-à-dire qu'un signal peut être tranformé selon les trois étapes à la place qu'il occupe en mémoire de

l'ordinateur. En effet, on travaille d'abord sur les coefficients pairs, puis sur les coefficients impairs du signal, on peut donc stocker le résultat intermédiaire aux positions impaires et le résultat final aux positions paires du signal.

– **Efficacité** : La transformation *lifting scheme* est une opération de calcul en $O(N)$ où N est la taille du signal de départ. Pour comparer, la tranformée de Fourier rapide est en $O(NlogN)$ et les transformations linéaires classiques sont en $O(N^2)$. La figure 3.16 donne le comparatif de gain de temps entre le *lifting scheme* et les autres transformées linéaires lorsque la taille de l'image augmente.
– **Parallélisation** : Les schémas d'analyse et de synthèse montrent immédiatement la possibilité de paralléliser la méthode sur des architecture SIMD (Single Instruction Multiple Data).

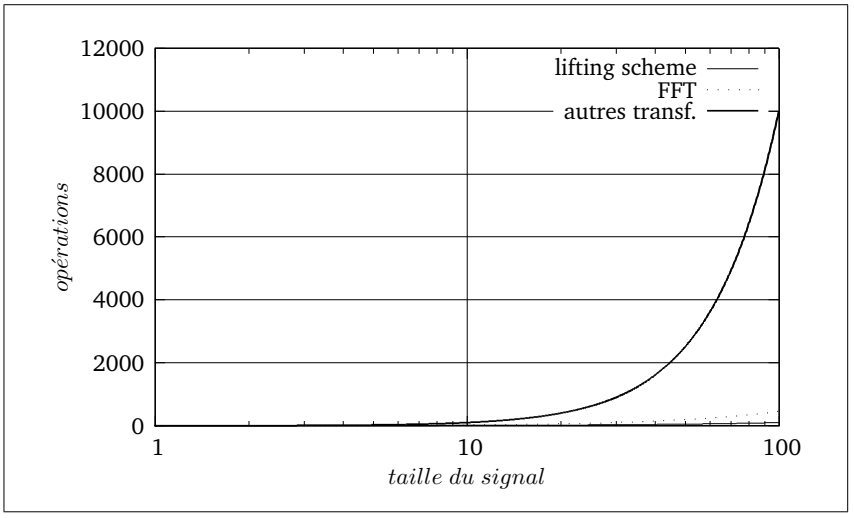


FIG. 3.16 – Comparaison du nombre d'opérations entre le lifting sheme et les autres transformations linéaires

Mais les deux principaux avantages de la méthode restent :

– **Transformée inverse simple** : Contrairement aux méthodes basées sur l'espace de Fourier, l'inverse de l'analyse est obtenue très simplement en réalisant les opérations en ordre inverse en changeant le signe des opérateurs.
– **Généralité** : La méthode peut facilement être étendue aux domaines où l'analyse de Fourier n'est pas facile à mettre en œuvre : Les courbes, les surfaces, les volumes, les domaines non bornés...

Ce sont tous ces avantages qui sont à l'origine de l'utilisation de la transformée en ondelettes *lifting scheme* pour la compression des images dans la norme JPEG2000 [13] destinée à remplacer le format JPEG basé sur la transformée en cosinus discret.

Dans la suite de ce document, nous utiliserons une propriété intéressante du *lifting scheme*. Calderbank et al. [CDSY98] ont défini une transformée *lifting scheme* en nombres entiers. Ainsi, les transformées effectuées sont uniquement basées sur les nombres entiers, ce qui accélère beaucoup les calculs.

Les ondelettes utilisées forment une version en nombres entiers des ondelettes de Deslauriers-Dubuc [Swe96]. Ces ondelettes disposent d'une bonne localisation en temps et en fréquence. La figure 3.17 montre l'allure de la fonction d'échelle et de l'ondelette pour cette famille d'ondelettes en nombres entiers.

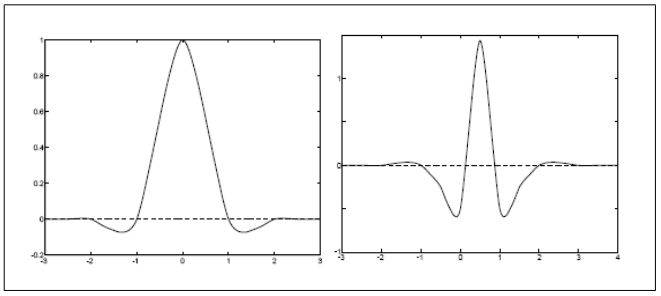


FIG. 3.17 – Exemple de fonction d'échelle et de fonction d'ondelettes de Deslauriers-Dubuc en nombres entiers.

Après avoir introduit les outils de traitement d'images nécessaires à la réduction des informations contenues dans l'image, nous allons nous intéresser aux différents attributs que nous allons extraire des images pour les représenter.

Chapitre 4

Attributs et descripteurs d'images

Les images sont des objets numériques à deux dimensions (quelconques) très riches en terme d'information. Leur manipulation directe ne permet pas d'obtenir des temps de réponse réalistes pour un système de recherche d'images. Il convient donc d'utiliser une représentation de dimension réduite pour caractériser une image. Ainsi, on va extraire des **attributs** caractéristiques de l'image à l'aide de fonctions mathématiques et on va les regrouper sous la forme d'un représentant de l'image : le **vecteur descripteur** de l'image.

4.1 De l'importance des attributs

Les attributs vont représenter l'image, leur choix est déterminant pour la suite de la méthode. Si les attributs sont mal choisis, la méthode de classification donnera de mauvais résultats. Comment choisir de bons attributs ? Il n'y a pas de réponse générale à cette question car le choix des attributs va dépendre de ce que l'on souhaite classer.

Les attributs sont en général choisis par un expert du domaine des images de la base. L'expert justifie le choix des attributs par son expérience, sur les caractéristiques qui lui semblent importantes et sur le champ applicatif de la méthode de recherche. Suivant l'application développée, l'expert pourra choisir différents attributs.

Le choix des attributs est fortement dépendant des images de la base. Ainsi, les attributs qui donnent d'excellents résultats sur une base d'images peuvent donner des résultats médiocres sur une autre base. Il n'y a pas d'attributs universel donnant de bons résultats sur n'importe quelle base d'images. Les attributs dépendent du contexte (*context-sensitive*). Le principe général de

notre système est indépendant du choix des attributs. Nous proposerons un choix afin de valider et d'illustrer notre méthode, mais elle peut évidemment faire appel à d'autres familles d'attributs que celles que nous utilisons.

Il existe deux familles d'attributs, les attributs **globaux** qui sont calculés à partir de l'image entière et les attributs **locaux** qui sont calculés sur une région de l'image considérée.

Dans le cadre de ce travail de recherche, seuls des attributs globaux calculés sur toute l'image ont été utilisés. Ces attributs sont bien adaptés aux images manipulées car elles ne contiennent qu'un objet centré sur un fond uniforme. Il n'est pas nécessaire d'utiliser des attributs locaux dans ce cas.

4.2 Les principaux attributs

Des attributs de différents types sont utilisés pour représenter le contenu de l'image. Les attributs sont classés en trois familles principales : la couleur, la texture et la forme. Pour l'étude de la base d'images Trans'tyfipal, la couleur sera peu utilisée en raison de la coloration artificielle des images d'objets paléontologiques. Par contre, le classement des objets sera effectué à l'aide d'attributs issus de la forme et de la texture des objets.

4.2.1 La couleur

Depuis l'expérience de décomposition de la lumière blanche par un prisme réalisée par Isaac Newton vers 1665 et les travaux de James Clerk Maxwell vers 1865, on sait que la lumière est composée d'un ensemble d'ondes électromagnétiques de longueurs d'ondes différentes.

La lumière est une énergie électromagnétique comportant une partie ondulatoire (onde) et une partie corpusculaire (les photons). L'énergie lumineuse possède une longueur d'onde caractérisée par sa couleur. La figure 4.1 donne la correspondance longueur d'onde/couleur pour le spectre visible par l'œil humain, soit de 380 à 780 nm environ.

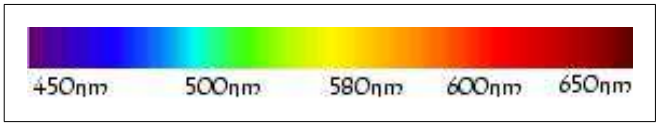


FIG. 4.1 – Correspondance longueur d'onde/couleur pour le spectre visible.

Les modèles de perception humaine des couleurs ont été étudiés [All99, Car95, Nad00] pour adapter au mieux les réactions des systèmes de recherche d'images à la perception humaine : «La couleur n'est pas une manifestation physique proprement dite. Elle résulte de la manière dont l'homme perçoit un spectre de longueurs d'ondes et de la manière dont les neurones du système visuel codent ce spectre.».

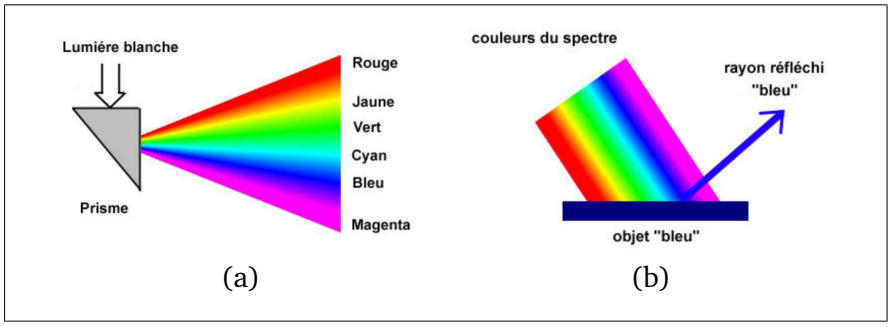


FIG. 4.2 – (a) la décomposition de la lumière blanche par un prisme (expérience d'Isaac Newton vers 1665) et (b) la perception du bleu renvoyé par un objet.

C'est le physicien James Clerk Maxwell qui a prouvé que toute couleur peut être obtenue par synthèse de trois couleurs seulement : le rouge, le vert et le bleu. À partir de cette découverte, la colorimétrie, la science qui étudie la couleur a vu le jour. Le principe des trois couleurs primaires est employé dans la télévision ou le moniteur de l'ordinateur.

La biologie nous indique que l'œil dispose de deux types de cellules réceptrices : les bâtonnets sensibles à la luminance (quantité de lumière) et les cônes sensibles à la chrominance (couleur). Il existe trois types de cônes : rouge, vert et bleu. Il y a beaucoup moins de cônes bleu que de rouge et de vert. La figure 4.3 présente la courbe de sensibilité de l'œil aux différents stimuli colorés.

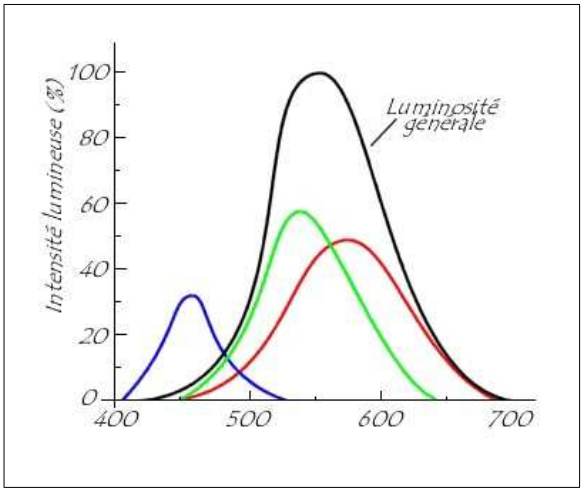


FIG. 4.3 – Sensibilité de l'œil aux différentes longueurs d'ondes.

L'espace **RGB** (Red-Green-Blue) défini par la compagnie internationale de l'éclairage (C.I.E.) en 1931 représente les couleurs par synthèse soustractive. Chaque couleur est représentée par trois composantes : le rouge (R), le vert (G) et le bleu (B) dont les longueurs d'onde normalisées

sont respectivement $\lambda_R = 700\,nm$, $\lambda_G = 546,1\,nm$ et $\lambda_B = 435,8\,nm$.

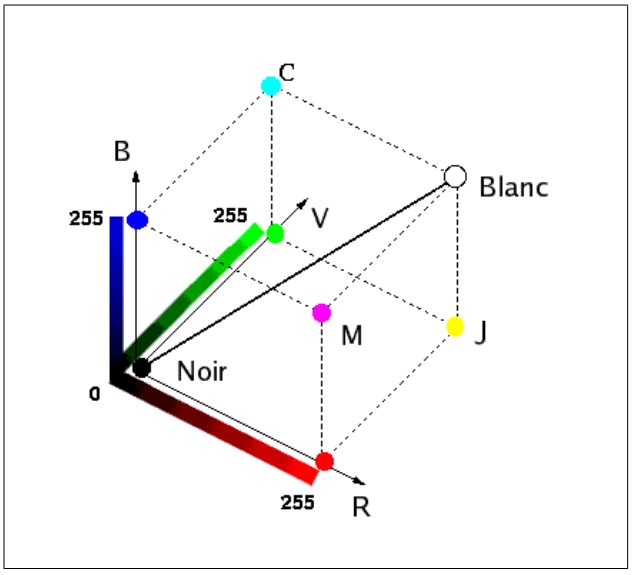


FIG. 4.4 – Cube de l'espace couleur RGB.

En général, les trois composantes sont représentées sur l'ordinateur par trois octets (24 bits). Avec ce choix pour le stockage des couleurs, on obtient une palette de 256^3 soit $16\,777\,216$ couleurs possibles. Le système RGB est représenté par un cube (figure 4.4) dont les sommets sont les couleurs de base, noir, blanc, rouge, vert, bleu, cyan, magenta, jaune ; le noir et le blanc étant diamétralement opposés.

$$C = [R, V, B]\ avec\ R \in [0, 255],\ G \in [0, 255]\ et\ B \in [0, 255] \tag{4.1}$$

Le principal inconvénient de cet espace de représentation vient du fait qu'il ne tient pas compte de la variation de la sensibilité de l'œil.

La C.I.E. a défini un espace de représentation qui prend en compte la sensibilité de l'œil. Il s'agit de l'espace **XYZ**. La figure 4.5 donne ce diagramme qui permet de tenir compte de la sensibilité de l'œil aux différents stimuli.

L'espace XYZ est rarement utilisé dans les recherches d'images car il n'est pas perceptuellement uniforme. C'est-à-dire que les différentes nuances d'une même teinte (par exemple rouge) peuvent être très éloignées dans l'espace. On lui préfère d'autres espaces qui ont des propriétés plus favorables au traitement informatique de la couleur.

L'espace **HSV** (Hue-Saturation-Value) — teinte, saturation et valeur (luminance) — possède une composante de luminance, une composante de teinte qui correspond à la couleur et une saturation de cette teinte (qui correspond à la pureté de la couleur). Il est représenté par un cône (figure 4.6) dont les sommets sont le noir et le blanc.

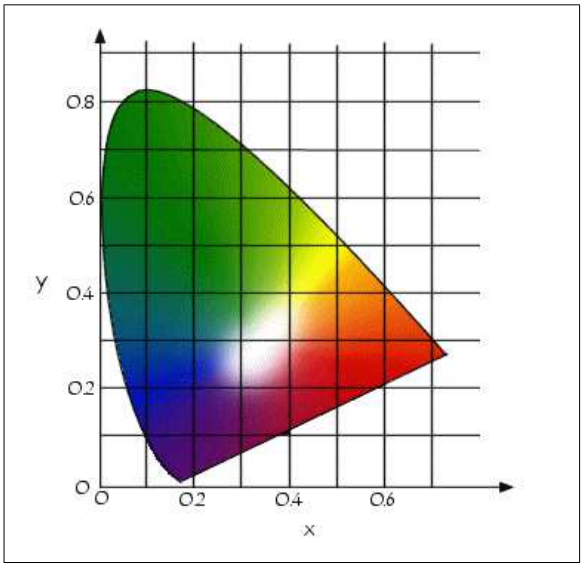


FIG. 4.5 – Représentation de l'espace couleur XYZ.

Cet espace présente l'avantage de simuler le comportement visuel humain dans lequel la couleur est décomposée en une image de luminance (la valeur) et une image de chrominance (teinte et saturation). La teinte fournit directement l'information de la couleur dominante dans les différentes régions de l'image. Dans sa thèse, Thierry Carron [Car95] utilise cet espace pour la segmentation d'images couleur.

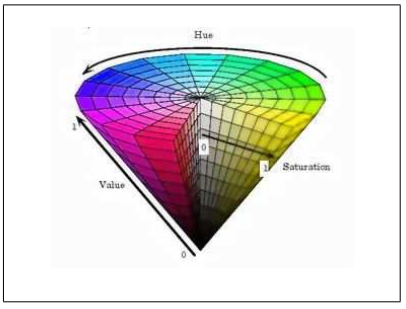


FIG. 4.6 – Représentation de l'espace couleur HSV.

L'espace HSV n'est malheureusement pas uniforme, c'est-à-dire qu'une distance entre couleurs visuellement proches calculée dans cet espace peut être très grande. Un autre espace a donc été crée par la C.I.E. pour résoudre cette difficulté.

L'espace **Lab** est un espace couleur perceptuellement uniforme, c'est-à-dire qu'il possède la bonne propriété de respecter les distances entre couleurs visuellement proches. Il est défini à partir de l'espace XYZ par des relations non-linéaires et par le choix d'un blanc de référence. Le blanc de référence est le D_{65}, c'est-à-dire le point de mesure de la lumière blanche émise par le

soleil dont les coordonnées sont $D_{65} = (X_0, Y_0, Z_0) = (0.3127, 0.3290, 0.3583)$.

Dans la suite de cette thèse, nous utiliserons les espaces HSV et Lab pour extraire des informations couleurs des images.

4.2.1.1 Les histogrammes

Les histogrammes [SB91] sont des indicateurs de répartition de niveaux de gris (ou de couleurs) dans une image. Ils sont très utilisés en recherche par le contenu car l'histogramme d'une image est presque invariant en rotation, translation et changement d'échelle de cette image.

À partir de l'histogramme, des attributs colorimétriques de l'image peuvent être extraits.

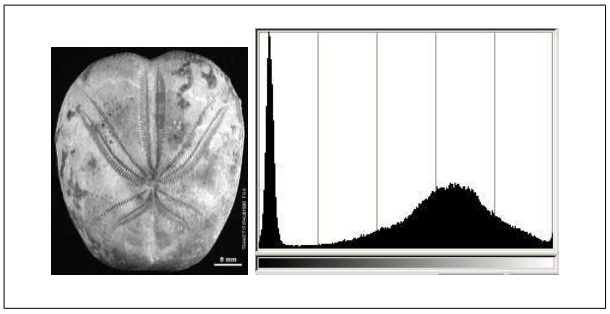


FIG. 4.7 – Exemple de l'histogramme d'une image en niveaux de gris.

Pour les images en couleurs, on utilise trois histogrammes, un par plan de bits de l'image (rouge, vert et bleu). De nombreux exemples d'attributs issus des histogrammes peuvent être calculés [**?**].

Une autre famille d'attributs d'images est la forme (*shape*).

4.2.2 La forme

La forme est un descripteur très important dans les bases d'images. La forme désigne l'aspect général d'un objet, son contour.

4.2.2.1 Les attributs géométriques de région

Les attributs géométriques de forme permettent de distinguer les différents types de forme que peuvent prendre les objets d'une scène. Ils nécessitent une segmentation en région préalable de l'image. Ils sont ensuite calculés sur les différentes régions de l'image.

La **surface** relative (ou normalisée) d'une région R_k de l'image $\mathcal{I}$ est le nombre de pixels contenus dans cette région par rapport au nombre total de pixels de l'image :

$$S_k = \frac{card(R_k)}{hauteur(\mathcal{I}) * largeur(\mathcal{I})}$$

Le **centre de masse** des pixels de la région est défini par :

$$P = (P_i, P_j) = (\frac{\sum_{i \in R_k} i/card(R_k)}{largeur(\mathcal{I})}, \frac{\sum_{j \in R_k} j/card(R_k)}{hauteur(\mathcal{I})})$$

La **longueur du contour** de la région est le nombre de pixels en bordure de la région :

$$l_k = card(contour(R_k))$$

La **compacité** traduit le regroupement des pixels de la région en zones homogènes et non trouées :

$$C_k = \frac{l_k^2}{S_k}$$

.

Ces attributs très simples permettent d'obtenir des informations sur la géométrie des régions de l'image. Il existe d'autres attributs de forme, basés sur des statistiques sur les pixels des régions de l'image.

4.2.2.2 Les moments géométriques

Les moments géométriques [SHB99] permettent de décrire une forme à l'aide de propriétés statistiques. Ils sont simples à manipuler mais leur temps de calcul est très long.

Formule générale des moments :

$$m_{p,q} = \sum_{p=0}^{m} \sum_{q=0}^{n} x^p y^q f(x, y) \tag{4.2}$$

L'ordre du moment est $p + q$. Le moment d'ordre 0 $m_{0,0}$ représente l'aire de la forme de l'objet.

Les deux moments d'ordre 1 $m_{0,1}$ et $m_{1,0}$, associés au moment d'ordre 0 $m_{0,0}$ permettent de calculer le centre de gravité de l'objet. Les coordonnées de ce centre sont :

$$x_c = \frac{m_{1,0}}{m_{0,0}} \tag{4.3}$$

$$y_c = \frac{m_{0,1}}{m_{0,0}} \tag{4.4}$$

Il est possible de calculer à partir de ces moments l'ellipse équivalente à l'objet. Afin de calculer les axes de l'ellipse, il faut ramener les moments d'ordre 2 au centre de gravité :

$$m_{2,0}^g = m_{2,0} - m_{0,0}\, x_c^2 \tag{4.5}$$

$$m_{1,1}^g = m_{1,1} - m_{0,0}\, x_c\, y_c \tag{4.6}$$

$$m_{0,2}^g = m_{0,2} - m_{0,0}\, y_c^2 \tag{4.7}$$

Puis on détermine l'angle d'inclinaison de l'ellipse α.

$$\alpha = \frac{1}{2}\ arctan \frac{2m^g_{1,1}}{m^g_{2,0} - m^g_{0,2}} \tag{4.8}$$

L'angle α est défini à $\frac{\pi}{2}$ près. La table 4.8 donne la valeur de l'angle en fonction du numérateur et du dénominateur de l'équation 4.8.

$m_{2,0} - m_{0,2}$	$m_{1,1}$	valeur	α
0	0		0
0	>0		$\frac{\pi}{4}$
0	<0		$-\frac{\pi}{4}$
>0	0		0
<0	0		$-\frac{\pi}{2}$
>0	>0	$\frac{1}{2}\ arctan \frac{2m^g_{1,1}}{m^g_{2,0}-m^g_{0,2}}$	$0 < \alpha < \frac{\pi}{4}$
>0	<0	$\frac{1}{2}\ arctan \frac{2m^g_{1,1}}{m^g_{2,0}-m^g_{0,2}}$	$-\frac{\pi}{4} < \alpha < 0$
<0	>0	$\frac{1}{2}\ arctan \frac{2m^g_{1,1}}{m^g_{2,0}-m^g_{0,2}} + \frac{\pi}{2}$	$\frac{\pi}{4} < \alpha < \frac{\pi}{2}$
<0	<0	$\frac{1}{2}\ arctan \frac{2m^g_{1,1}}{m^g_{2,0}-m^g_{0,2}} - \frac{\pi}{2}$	$-\frac{\pi}{2} < \alpha < -\frac{\pi}{4}$

FIG. 4.8 – Table de calcul des angles de l'ellipse équivalente à une région.

À partir des moments géométriques, Hu [Hu62] a introduit sept invariants aux translations, rotations et changement d'échelle, appelés moments de Hu.

$$\begin{aligned}
M_1 &= \mu_{20} + \mu_{02} \\
M_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \\
M_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \\
M_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \\
M_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu 03)^2] \\
&\quad +(3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \\
M_6 &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \\
&\quad + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{03} + \mu_{21}) \\
M_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\
&\quad -(\mu_{30} - 3\mu_{21})(\mu_{12} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{12} + \mu_{03})^2]
\end{aligned} \tag{4.9}$$

Les moments de Hu offrent d'excellents attributs invariants en translation, rotation et changement d'échelle pour décrire une image. Cependant leur calcul est relativement long et ils sont très sensibles au bruit, ce qui peut s'avérer être un gros inconvénient dans un système de recherche d'images.

Le dernier type d'attributs extraits des images est la texture.

4.2.3 La texture

Il existe deux type de définition de la texture [BCC+95] :

– La première est **déterministe** et fait référence à une répétition spatiale d'un motif de base dans différentes directions. Cette approche structurelle correspond à une vision **macroscopique** des textures. La peau des reptiles est un exemple de texture macroscopique dans lequel la primitive est l'écaille.

– La seconde est **probabiliste** et cherche à caractériser l'aspect chaotique qui ne comprend ni motif localisable, ni fréquence de répétition principale. Cette approche est **microscopique** et l'herbe en est un excellent exemple.

Les méthodes d'extraction des attributs dépendent du modèle de texture recherché. Pour les textures macroscopiques, on va chercher des répétitions d'un motif dans une certaine direction à l'aide des matrices de co-occurrence ou des matrices de longueur de plage [BCC+95]. Pour les textures microscopiques, les techniques de recherche sont basées sur des calculs statistiques sur les niveaux de gris des images. L'analyse de Fourier et l'analyse en ondelettes permettent de calculer des attributs de texture mixtes.

4.2.3.1 La matrice de co-occurence

La texture d'une image peut être interprétée comme la régularité d'apparition de couples de niveaux de gris selon une distance donnée dans l'image. La matrice de co-occurence[Pos87] contient les fréquences spatiales relatives d'apparition des niveaux de gris selon quatre directions : $\theta = 0$, $\theta = \pi/2$, $\theta = \pi/4$ et $\theta = 3\pi/4$. La matrice de co-occurence est une matrice carrée $n * n$ où n est le nombre de niveaux de gris de l'image.

On définit la matrice des fréquences relatives F par :

$$F(d, \theta) = (f(i, j|d, \theta))$$

$f(i, j|d, \theta)$ représente le nombre de fois où un couple de points séparés par la distance d dans la direction θ a présenté les niveaux de gris g_i et g_j. Pour obtenir de véritables fréquences relatives, il faut normaliser les éléments de la matrice en les divisant par le nombre total de paires de points élémentaires séparés par la distance d dans la direction θ dans toute l'image.

À partir de la matrice de co-occurrence, on extrait différents attributs de texture.

L'homogénéité h : $h = \sum_{i=1}^{n} \sum_{j=1}^{n} f(i, j|d, \theta)^2$.

L'entropie e : $e = \sum_{i=1}^{n} \sum_{j=1}^{n} f(i, j|d, \theta) \log_2 f(i, j|d, \theta)$.

Le contraste c : $c = \sum_{k=0}^{n-1} n^2 \left(\sum_{i=1}^{n} \sum_{j=1}^{n} f(i, j|d, \theta) \right)$, $|i - j| = n$.

L'inertie i : $i = \sum_{i=1}^{n}\sum_{j=1}^{n}(i-j)^2\, f(i,j|d,\theta)$.

La corrélation r :

$$r = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(i-\mu_x)(j-\mu_y)\, f(i,j|d,\theta)}{\sigma_x\sigma_y}$$

$$\text{avec } \mu_x = \sum_{i=1}^{n} i \sum_{j=1}^{n} f(i,j|d,\theta)$$

$$\text{avec } \mu_y = \sum_{j=1}^{n} j \sum_{i=1}^{n} f(i,j|d,\theta)$$

$$\text{avec } \sigma_x = \sum_{i=1}^{n}(i-\mu_x)^2 \sum_{j=1}^{n} f(i,j|d,\theta)$$

$$\text{avec } \sigma_y = \sum_{j=1}^{n}(j-\mu_y)^2 \sum_{i=1}^{n} f(i,j|d,\theta)$$

Il existe d'autres attributs de textures issus de la matrice de co-occurence qui ne seront pas abordés dans ce travail.

4.2.3.2 La transformée de Fourier discrète

De nombreuses méthodes d'extraction d'attributs de texture sont basées sur la transformée de Fourier. Cette dernière permet de passer du domaine spatial de l'image (coordonnées m et n) au domaine fréquentiel de l'image (coordonnées u et v).

La transformée de Fourier discrète d'une séquence 2D correspondant au signal discret $s[m,n]$, avec m et n entiers, $0 \leq m \leq M-1$, $0 \leq n \leq N-1$ est donnée par l'équation 3.5 page 54.

Comme on travaille dans le plan complexe, pour éviter d'interpréter la phase, on utilise souvent le module de la transformée de Fourier ou spectre de Fourier. Le spectre est très riche en information, il permet de déterminer des attributs dans le plan fréquentiel de l'image. La figure 3.5 page 55 donne un exemple de transformée de Fourier d'une image.

4.2.3.3 Les ondelettes

À partir de la transformée en ondelettes on peut extraire des attributs de différents types et à différents niveaux de résolution. L'image d'approximation donne des informations sur les régions qui composent l'image, d'une résolution fine à une résolution grossière. Les images de détails donnent des informations horizontales, verticales et diagonales sur l'image.

L'énergie des coefficients d'ondelettes est directement disponible, on la calcule en prenant la somme des carrés des coefficients d'ondelettes. On a ainsi des mesures d'énergie à différents niveaux de résolution.

4.3 Le vecteur descripteur

À partir des attributs extraits de l'image, on construit un vecteur caractéristique de l'image, c'est le **vecteur descripteur** de l'image. Ce vecteur descripteur contient les attributs intéressants extraits de l'image. Il se présente en général sous la forme d'un vecteur à n composantes réelles.

Les attributs extraits des images sont de différents types et sont exprimés dans des unités différentes selon qu'ils appartiennent à la couleur, la texture, la forme. Une étape de **normalisation** est indispensable, elle va permettre de réajuster les valeurs des attributs pour les rendre commensurables.

Cette phase est indispensable car lors de la classification, un calcul de distance est effectué sur les vecteurs. Si la distance euclidienne est utilisée et si un attribut prend une valeur maximale de 10^6 alors qu'un autre prend la valeur 10, ils n'ont pas le même poids dans le calcul de distance. En effet, la distance euclidienne est calculée comme la racine carrée de la différence des coefficients au carré. Donc 10^2 sera négligeable devant $(10^6)^2 = 10^{12}$ et les coefficients n'auront pas la même contribution à la distance calculée.

Dans notre plateforme, les ondelettes utilisées [CDSY98] fournissent des résultats en nombre entiers. On impose que les attributs extraits des images transformées soient également entiers et que leurs valeurs soient comprises entre 0 et 100. La normalisation est effectuée en calculant pour chaque attribut (colonne), $mini_n$ et $maxi_n$. On opère ensuite une soustraction et une division (avec arrondi à l'entier inférieur si nécessaire) pour obtenir le résultat normalisé.

$$v'_k = \frac{v_k - mini_n}{maxi_n - mini_n} * 100$$

Voici un exemple :

$$\begin{array}{rl} v_1 = [-3; -6; 10; 120] & v'_1 = [33; 0; 72; 100] \\ v_2 = [-7; 4; 15; 0] & v'_2 = [0; 100; 100; 0] \\ v_3 = [5; 4; -3; -1] & v'_3 = [100; 100; 0; 0] \\ Avant\ normalisation. & Après\ normalisation. \end{array}$$

On est assuré de travailler avec des valeurs de vecteur descripteur comparables entre elles et aucune d'entre elles ne peut fausser le résultat. Bien que cette méthode permette de normaliser les attributs, il faut utiliser des attributs de même type si on veut que le calcul de distance ait un

sens mathématique. Il est toujours plus avantageux de calculer des distances sur des attributs ayant des valeurs comparables.

4.4 Les attributs utilisés

Les attributs utilisés dans ce travail sont de trois types : la forme, la texture et la couleur. En ce qui concerne la base trans'tyfipal, la couleur n'est pas importante car les objets photographiés sont décolorés ou colorés artificiellement avant photographie. Par contre, pour la base Columbia, l'information couleur est importante puisque les utilisateurs peuvent effectuer des requêtes sur la couleur.

Nous avons utilisé pour les tests un ensemble d'attributs organisés en signature. Ces attributs sont issus de la transformée en ondelettes des images à trois niveaux de résolution. Soient ap, dh, dv et dd les imagettes d'approximation, de détails horizontaux, verticaux et diagonaux. Soit k le niveau de résolution avec $k = 1$ à $k = 3$. Les attributs sont issus de deux espaces colorimétriques différents : HSV (dont les plans sont notés h, s et v) et Lab (dont les plans sont notés l, a, b).

Les attributs extraits sont décrits ci-dessous :

- **Forme**
  - Rapport petit axe sur grand axe de l'ellipse englobante des imagettes r_{lk} et r_{hk},
  - Moments des imagettes ap_k : $m_{k_{(0,0)}}$, $m_{k_{(0,1)}}$, $m_{k_{(1,0)}}$, $m_{k_{(1,1)}}$, $m_{k_{(2,1)}}$, $m_{k_{(1,2)}}$,
  - Moments de Hu des imagettes ap_k : M_{1_k},M_{2_k}, M_{3_k}, M_{4_k}, M_{5_k}, M_{6_k}, M_{7_k}.
- **Texture**
  - Energie des coefficients d'ondelettes pour les imagettes $E(dh_{lk})$ et $E(dh_{vk})$,
  - Energie des coefficients d'ondelettes pour les imagettes $E(dv_{lk})$ et $E(dv_{vk})$,
  - Energie des coefficients d'ondelettes pour les imagettes $E(dd_{lk})$ et $E(dd_{vk})$,
  - Moyenne (m) et écart-type (σ) des niveaux de gris des imagettes dh_{lk}, dh_{vk},
  - Moyenne (m) et écart-type (σ) des niveaux de gris des imagettes dv_{lk}, dv_{vk},
  - Moyenne (m) et écart-type (σ) des niveaux de gris des imagettes dd_{lk}, dd_{vk},
- **Couleur**
  - Energie des coefficients d'ondelettes pour les imagettes $E(dh_{ak})$, $E(dh_{bk})$, $E(dh_{hk})$ et $E(dh_{sk})$,
  - Energie des coefficients d'ondelettes pour les imagettes $E(dv_{ak})$, $E(dv_{bk})$, $E(dv_{hk})$ et $E(dv_{sk})$,
  - Energie des coefficients d'ondelettes pour les imagettes $E(dd_{ak})$, $E(dd_{bk})$, $E(dd_{hk})$ et $E(dd_{sk})$,
  - Moyenne (m) et écart-type (σ) des niveaux de gris des imagettes dh_{ak}, dh_{bk}, dh_{hk} et dh_{sk},

– Moyenne (m) et écart-type (σ) des niveaux de gris des imagettes dv_{ak}, dv_{bk}, dv_{hk} et dv_{sk},
– Moyenne (m) et écart-type (σ) des niveaux de gris des imagettes dd_{ak}, dd_{bk}, dd_{hk} et dd_{sk}.

À l'issue de l'extraction des paramètres, on obtient un vecteur descripteur v. Les vecteurs signatures v_{sk} sont générés à partir des attributs de v, réorganisés hiérarchiquement avec un expert en fonction de requêtes types des utilisateurs. Le tableau ci-dessous donne un exemple de vecteur signature utilisé lors des tests avec des attributs d'abord de forme, puis de texture, enfin de couleur. Cet exemple a été utilisé pour la base Columbia.

Niveau	Vecteur signature
1	$v_{s1} = r_{h1}, m1_{(0,0)}, m1_{(0,1)}, mk_{(1,0)}$
2	$v_{s2} = m1_{(0,0)}, m1_{(0,1)}, mk_{(1,0)}, mk_{(1,1)}, mk_{(2,1)}, mk_{(1,2)}$
3	$v_{s3} = E(dh_{lk}), E(dh_{vk}), E(dv_{lk}), E(dv_{vk}),$ $E(dd_{lk}), E(dd_{vk}), m(dd_{hk}), \sigma(dd_{hk})$
4	$v_{s4} = E(dh_{ak}), E(dh_{bk}), E(dv_{ak}), E(dv_{bk}),$ $E(dd_{ak}), E(dd_{bk}), E(dd_{hk}), E(dd_{sk}), m(dd_{hk}), \sigma(dd_{hk})$

Le chapitre suivant présente quelques méthodes de classification de données multidimensionnelles.

Chapitre 5

Classification en familles d'images

5.1 La classification

La classification consiste à organiser un ensemble de données multidimensionnelles en un **ensemble fini de classes** selon un ou plusieurs **critère(s) de classification** à l'aide d'un **classifieur**. Il existe de nombreuses méthodes de classification de données multidimensionnelles. La figure 5.1 illustre le principe de la classification. Les données sont représentées par des points (vecteurs) dans un espace à n dimensions (dans notre exemple n=2). En sortie de la classification, on obtient m classes (ou familles) de points selon les critères donnés et le classifieur choisi (dans notre exemple, m=3).

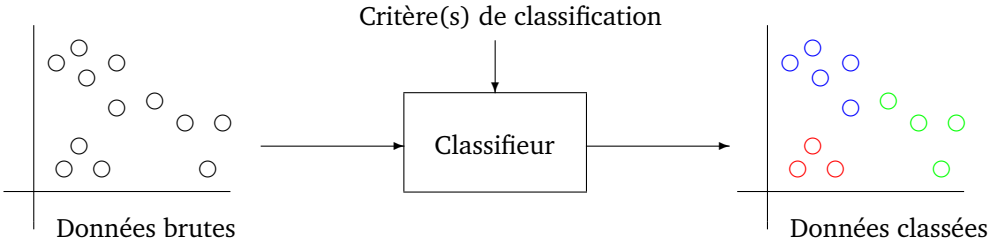


FIG. 5.1 – Principe de la classification.

5.1.1 Les différentes classifications

Parmi toutes les techniques de classification, on distingue :

– La classification **supervisée** dans laquelle un expert a fourni le modèle exact des classes à obtenir,
– La classification **non supervisée** dans laquelle le nombre de classes, inconnu a priori, est déduit directement des données.

On distingue ensuite :

– La classification **avec apprentissage** dans laquelle on entraîne le classifieur à l'aide d'un ensemble de données connues a priori. Cet entraînement a pour but d'adapter les sorties du classifieur en fonction des entrées qu'on lui soumet.
– La classification **sans apprentissage** où le classifieur travaille directement sur les données sans aucune connaissance préalable.

On trouve enfin des méthodes de classification :

– **paramétriques** dans lesquelles on fournit un certain nombre de paramètres au classifieur qui vont influencer le résultat de la classification.
– **non-paramétriques** dans lesquelles le classifieur doit se débrouiller seul pour classer les données sans aide extérieure.

Dans le cas de la classification supervisée, un expert identifie les classes de données et on classe les données existantes parmi elles. Dans le cas non supervisé, les classes sont construites en fonction des données, selon l'algorithme de classification employé. Dans cette étude, on considère qu'on n'a aucune connaissance a priori sur les classes, on se trouve dans le cas non supervisé.

5.1.2 La classification non-supervisée

Nous souhaitons obtenir une classification automatique des images en familles visuellement similaires. On s'oriente donc vers une méthode non-supervisée : le regroupement (*clustering*) en familles d'images visuellement similaires.

5.1.3 Les techniques de regroupement (*clustering*)

Les méthodes de regroupement (*clustering*) permettent de regrouper des valeurs multidimensionnelles en familles selon un critère de classification. Ces méthodes sont non-supervisées, c'est-à-dire que les familles obtenues sont déduites directement des données elles-mêmes, sans aucune connaissance particulière.

On distingue de nombreuses méthodes de regroupement, les trois principales familles de méthodes sont décrites ci-dessous, il s'agit :

– Les méthodes basées sur la **densité** des informations,
– Les techniques **hiérarchiques** de classification (ascendantes et descendantes),
– Les approches basées sur les **centres mobiles**.

5.1.3.1 L'approche densité des informations

Dans les méthodes basées sur la densité, on cherche à regrouper ensemble des points situés dans un voisinage d'un point non classé selon une certaine distance. Pour atteindre ce but, on définit une distance maximale entre deux points (voisinage) ainsi qu'un nombre de voisins minimal (densité). Ensuite, on procède au classement des données selon l'algorithme 5.2.

```
1- Tant qu'il y a des points non classés
{
   1.1- Choisir au hasard un point non classé qui a un voisinage dense, ce point et ses voisins formant un groupe
   1.2- Tant que le groupe n'est pas stable
      { 1.2.1- Chercher le voisinage dense de chaque point du groupe et l'ajouter au groupe }
}
```

FIG. 5.2 – Algorithme du voisinage dense.

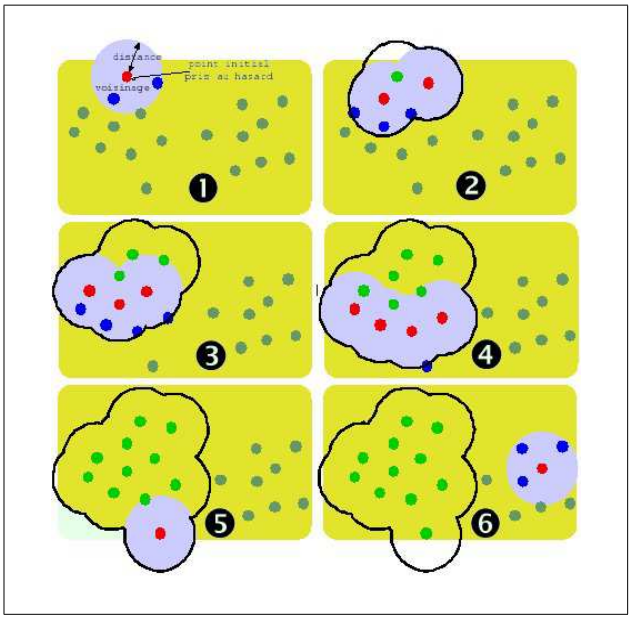


FIG. 5.3 – Un exemple de classification à l'aide des voisinages denses.

L'exemple donné en figure 5.3 illustre la méthode. On choisit au hasard un point non classé. Ce point et son voisinage sont classés dans le même groupe (1). On prend le voisinage de chacun des points du groupe et on le classe dans le même groupe (2) et ainsi de suite jusqu'à la stabilité du groupe obtenu (3), (4), (5). On recommence ensuite avec un autre point de départ non classé (6) et on réitère le processus jusqu'à ce que tous les points soient classés.

Cette méthode est la plus précise pour trouver le nombre de classes présentes dans les don-

nées sans indication préalable. Cependant, elle souffre de certains inconvénients : elle nécessite de nombreux calculs, elle est très sensible au choix de la distance et du nombre minimum de voisins.

5.1.3.2 L'approche hiérarchique

L'approche hiérarchique consiste à construire les familles par agglomération acendante de classes. On construit les classes par agglomération successive de deux groupes de données dont les centres de gravité sont plus proches qu'une distance d_{max} donnée. L'algorithme 5.4 est simple à mettre en œuvre.

```
1- Regrouper deux par deux les points les plus proches
2- Tant que les groupes ne sont pas stables
{
    2.1- Calculer le centre de gravité des groupes ainsi crées
    2.2- Regrouper deux par deux les groupes dont les centres ont une distance inférieure à d_max
}
```

FIG. 5.4 – Algorithme de classification hiérarchique (ou agglomération).

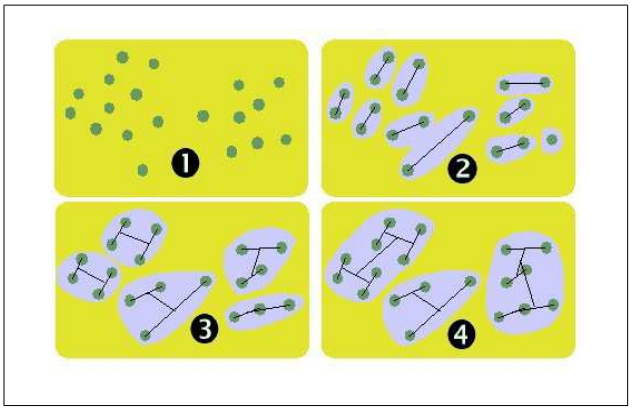


FIG. 5.5 – Un exemple de classification hiérarchique.

La figure 5.5 présente un exemple d'agglomération de classes. L'ensemble des points (1) est traité pour regrouper les points les plus proches par classes de deux (2). Les centres des classes sont calculés et les classes regroupées par deux (3). Enfin, quand les groupes obtenus sont trop éloignés pour être fusionnés, l'algorithme s'arrête.

Cette approche ne nécessite pas beaucoup de calculs et converge vite (par dichotomie). Par contre, elle ne sait traiter que des ensembles de données de cardinalité proche de 2^n, ce qui est un inconvénient.

5.1.3.3 L'approche centres mobiles

Cette technique itérative est basée sur le mouvement des centres de gravité des classes qui se déplacent vers une position de stabilité. Cette méthode nécessite un paramètre d'entrée qui est le nombre de classes à construire.

```
1- Choisir au hasard K centres de classe (noyaux)
2- Tant que non-stationnarité répéter
{
    2.1- Affecter chaque point à la classe du centre le plus proche
    2.2- Calculer le centre de gravité des classes obtenues
}
```

FIG. 5.6 – Algorithme des nuées dynamiques.

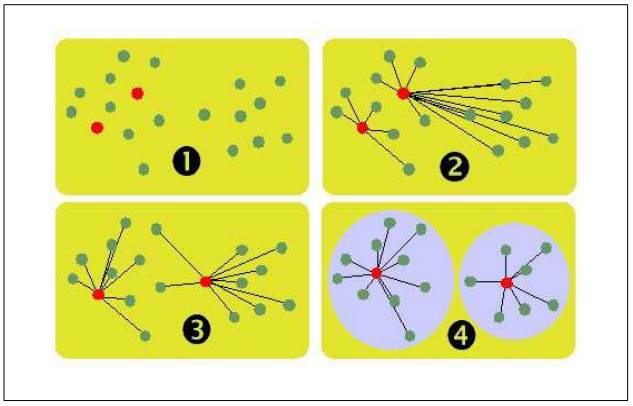


FIG. 5.7 – Un exemple de classification à l'aide des centres mobiles.

L'idée est de placer au hasard un nombre de centre de classes correspondant au nombre de classes souhaitées. Pour que l'algorithme donne de bons résultats, il est préférable d'avoir une connaissance a priori du nombre de classes présentes dans les données à regrouper. La classification est automatiquement obtenue par le déplacement itératif des centres de classes.

Cette technique n'implique pas de longs calculs et converge rapidement. Par contre, elle impose de donner le nombre de classes désirées au début de l'algorithme et elle est sensible aux conditions initiales.

5.2 Les distances métriques

Pour utiliser les méthodes de classification décrites ci-dessus, il est nécessaire de calculer des distances entre les données (vecteurs). Il faut donc définir précisément la notion de distance.

Par définition, une distance métrique sur un corps $\mathbb{K}$ entre deux vecteurs dans un espace à n dimensions est définie par :

Théorème 5.2.1 *Une fonction $d : \mathbb{K}^n \rightarrow \mathbb{K}$ est une distance métrique si*
1) $d(X,Y) = d(Y,X)\ \forall X \in \mathbb{K}^n \text{ et } Y \in \mathbb{K}^n$
2) $d(X,Y) \geq 0\ \forall X \in \mathbb{K}^n \text{ et } Y \in \mathbb{K}^n$
3) $d(X,Y) = 0 \Longleftrightarrow X = Y$
4) $d(X,Z) \leq d(X,Y) + d(Y,Z)\ \forall X \in \mathbb{K}^n,\ Y \in \mathbb{K}^n \text{ et } Z \in \mathbb{K}^n$

Toute fonction d répondant aux propriétés énoncées ci-dessus est donc une distance métrique. Il en existe de nombreuses dans la littérature. Les plus utilisées sont étudiées ci-dessous.

5.2.1 Distances de Minkowski

Une famille de métriques très utilisée est la famille des distances de Minkowski.

Dans un espace de dimension finie, la forme générale de la famille de distances de Minkowski s'écrit :

$$d_r(X,Y) = (\sum_{i=1}^{n} |x_i - y_i|^r)^{\frac{1}{r}} \ avec\ r \geq 1 \tag{5.1}$$

où n est la dimension du vecteur et r l'ordre de la distance.

Parmi cette famille de distances, trois d'entre elles possèdent des propriétés intéressantes :

– La **distance de Manhattan** (ou *city block*), $r = 1$, $d_C(X,Y) = (\sum_{i=1}^{n} |x_i - y_i|)$,

– La **distance euclidienne**, $r = 2$, $d_E(X,Y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$,

– La **distance du maximum**, $r \rightarrow +\infty$, $d_\infty(X,Y) = Max_{i=1}^{n} |x_i - y_i|$

Par définition, on a $\forall x, \forall y,\ d_C(x,y) \geq d_E(x,y) \geq d_\infty(x,y)$. Les boules unitaires des différentes distances (city block, euclidienne et du maximum) sont représentées en figure 5.8.

Les métriques de Minkowski sont simples d'utilisation, par contre leur calcul est réalisé en considèrant que chaque composante du vecteur apporte la même contribution à la distance. Afin de rendre compte de l'importance relative des composantes du vecteur les unes par rapport aux autres, les distances de Minkowski pondérées (équation 5.2) sont préférables.

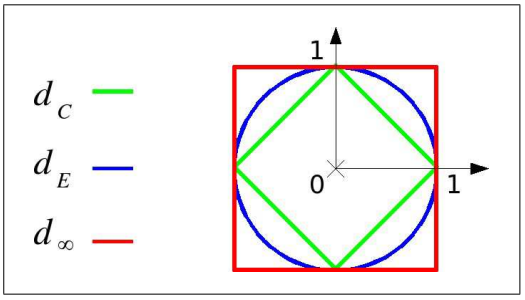


FIG. 5.8 – Boules unitaires pour les distances de Minkowski.

$$d_r^w(X,Y) = \left(\sum_{i=1}^{n} w_i \, |x_i - y_i|^r\right)^{\frac{1}{r}} \; avec \; r \geq 1 \tag{5.2}$$

où n est la dimension du vecteur, r est l'ordre de la distance et w est un vecteur de pondération à n composantes.

5.2.2 Distance de Mahalanobis

La *distance de Mahalanobis* permet de prendre en compte la distribution statistique des points dans l'espace car l'amplitude des différents attributs peut varier fortement, privilégiant les valeurs des attributs élevés dans le calcul de la distance.

Soient $x = (x_1, \ldots, x_p)^t$ et $y = (y_1, \ldots, y_p)^t$, deux points d'un même espace. La distance de Mahalanobis est définie par :

$$d(x, y) = \sqrt{(x - y)^t \, S^{-1} \, (x - y)}$$

S^{-1} est l'inverse de la matrice de covariance de x et y. La distance de Mahalanobis tient compte de la distribution statistique des données dans l'espace, c'est ce qui la différencie des autres distances. Il existe de nombreuses autres distances dans la littérature, elles ne seront pas abordées dans ce mémoire.

Dans la partie suivante, la méthode mise en œuvre est présentée de façon détaillée.

Troisième partie

Méthode de recherche

Chapitre 6

Méthodologie

Ce chapitre décrit le cahier des charges du système de recherche d'images par le contenu à concevoir en tenant compte des hypothèses décrites dans l'état de l'art. On souhaite disposer d'un système basé sur la navigation dans un arbre de familles d'images. Le problème est de définir précisément le fonctionnement optimal du système, c'est le rôle de ce chapitre.

6.1 Principe du système

Notre système de recherche d'images par le contenu [LT00, LT01] est basé sur la navigation visuelle dans la base d'images, il possède six étapes :

1. Transformation
2. Extraction
3. Organisation
4. Classification
5. Représentation
6. Navigation

L'étape 1 consiste en la transformation des images en ondelettes en nombres entiers (avec la méthode *lifting scheme*) afin d'obtenir des images multi-résolution dans des espaces couleur

intéressants. En 2, une extraction d'attributs multirésolution est effectuée sur les images transformées de l'étape 1. Ces attributs sont stockés sous forme de vecteurs descripteurs.

Dans l'étape 3, les attributs sont organisés en vecteurs signature multirésolution de taille croissante. Une fois organisés, dans l'étape 4, les vecteurs signature sont classés automatiquement en familles selon une distance.

La phase 5 sert à définir la représentation des familles d'images par une image modèle pour la navigation. Enfin l'étape 6 permet de naviguer visuellement dans la base d'images.

En résumé, on souhaite concevoir un système de recherche d'images basé sur des attributs multirésolution (en nombres entiers) dans lequel les images seront pré-classées en familles selon une hiérarchie d'attributs et dans lequel l'utilisateur pourra naviguer. Les choix retenus pour l'implémentation seront détaillés dans le chapitre 8.

Notre système doit posséder les propriétés suivantes, il se doit d'être :

- **extensible** : il est possible de rajouter dans l'architecture de base des descripteurs extraits par d'autres méthodes que celles que nous employons.
- **rapide** : l'extraction des attributs est effectuée sur des nombres entiers, ce qui permet de gagner de la vitesse par rapport à un traitement en nombres réels. La plupart des calculs sont effectués offline, ce qui ne pénalise pas l'utilisateur online.
- **efficace** : les images sont classées en familles visuellement similaires.
- **simple** : les images sont proposées à l'utilisateur qui navigue visuellement à travers la base à l'aide de la souris.
- **ouvert** : l'architecture permet aussi une recherche par image exemple classée dans l'arbre dont le résultat est le sous-arbre dans lequel l'image a été classée.
- **adaptable** : on distingue les utilisateurs non-spécialistes (recherche simple dans la base) des utilisateurs experts (possibilités étendues de paramétrage).
- **généraliste** : notre méthode est applicable à toutes les bases d'images avec quelques modifications sur les signatures.

6.1.1 Architecture de notre système

L'architecture de notre système [LT02, LT03b] est présentée en figure 6.1. Notre approche ajoute une étape importante par rapport à la recherche d'images «classique». En effet, à partir du vecteur descripteur (donc des attributs) multirésolution extrait lors de la phase hors-ligne (*offline*), on va construire plusieurs vecteurs signature de taille volontairement limitée dont les attributs seront un sous-ensemble du vecteur descripteur multirésolution. On va organiser les attributs descripteurs en attributs de signature.

La taille des vecteurs signature sera limitée à $n \leq 10$ dans le but de se défaire de la malédiction de la dimension. L'idée directrice de cette hiérarchisation est de n'utiliser qu'un vecteur

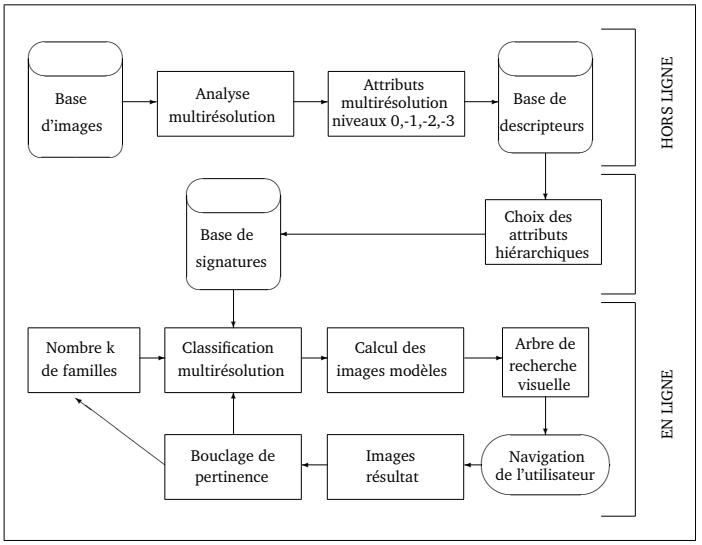


FIG. 6.1 – Notre approche de la recherche d'images.

signature à un instant donné afin de réduire le temps de calcul des diverses familles et de prendre en compte le contexte de la requête de l'utilisateur.

La figure 6.2 montre le schéma de la construction de la hiérarchie de vecteurs signature à partir des descripteurs. Ainsi, on part des descripteurs (vecteurs attributs) extraits des images et en les organisant, on arrive aux vecteurs signature de taille réduite.

Ces vecteurs signature ont deux fonctions principales :

– D'une part, ils doivent être de taille réduite croissante pour accélérer la recherche,
– D'autre part, ils synthétisent la stratégie de recherche de l'utilisateur (d'abord sur la forme, puis sur la texture, puis sur la couleur par exemple).

Ainsi, lors de la phase de classification, on utilise d'abord les vecteurs signature de faible longueur losqu'il y a beaucoup d'images à classer, puis on utilise des vecteurs signature de taille de plus en plus grande lorsque le nombre d'images à classer dans chaque famille diminue. Notre approche est une vue d'abord grossière puis de plus en plus fine (*coarse-to-fine*) de la base d'images.

De plus, les vecteurs signature sont organisés selon le choix de l'utilisateur. Par exemple, si l'utilisateur cherche les voitures rouges d'une base d'images, on classe d'abord les images selon leur forme (pour regrouper les voitures), puis leur couleur (pour regrouper les voitures rouges), puis leur texture (qui est peu importante dans ce cas de figure).

Le passage des vecteurs descripteurs aux vecteurs signature est effectué par un expert de la base d'image qui va décider des attributs à retenir pour répondre au mieux aux requêtes des utilisateurs. Toutefois, un utilisateur non-expert peut aussi utiliser le système car un certain nombre de vecteurs signature sont proposés par défaut.

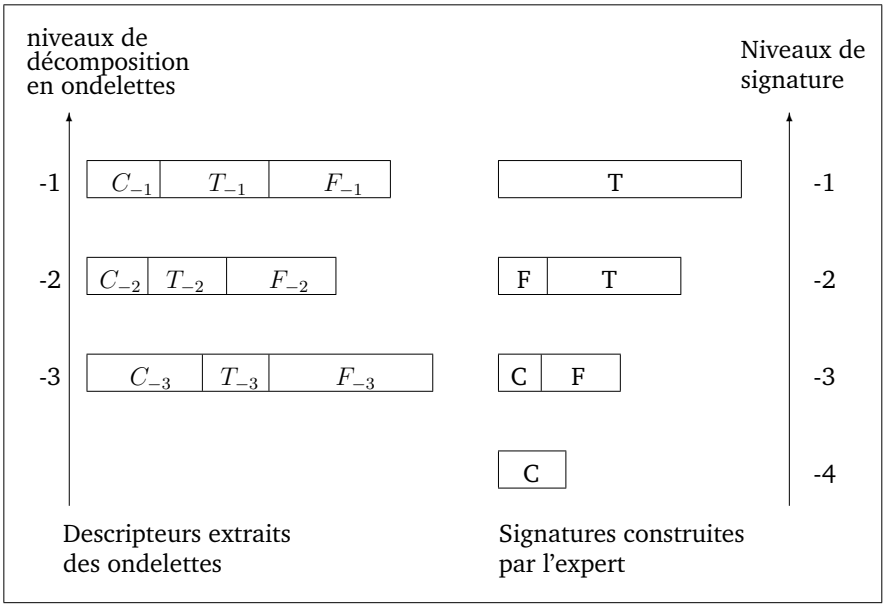


FIG. 6.2 – La construction des vecteurs signature.

6.1.2 Interface

L'interface de gestion de la base d'images et de la recherche doit être simple et compatible avec une utilisation à distance pour l'interrogation de la base. Elle doit permettre l'ajout d'une base d'images au système, l'extraction des attributs, leur organisation hiérarchique, la création de l'arbre de navigation visuelle et la phase de navigation.

L'interface ne doit pas demander un long temps d'apprentissage avant d'être opérationnelle. L'utilisateur doit pouvoir la prendre en main rapidement.

6.1.2.1 La méthode de regroupement retenue

Parmi les nombreuses méthodes de classification non-supervisées, nous avons choisi la méthode des nuées dynamiques. Pourquoi ?

Dans le but de réaliser un système de regroupement d'images en familles d'images visuellement similaires, nous devons choisir une méthode inspirée de la stratégie de regroupement de la vision humaine. Les nuées dynamiques fonctionnent sur le principe de la recherche par raffinements successifs. Cette méthode offre une certaine similitude avec la méthode humaine de recherche par tatonnement : on classe d'abord les images en paquets grossiers, puis on cherche à affiner le résultat du classement en tendant vers une solution idéale.

Les nuées dynamiques sont proches de la méthode *Self Organizing Map* (*SOM*), proposée par Kohonen [Koh90], qui est une méthode neuronale proche du comportement de déduction humain. Le temps de convergence des nuées dynamiques est relativement faible par rapport à

celui d'autres méthodes.

Ce sont ces différentes raisons qui ont guidé notre choix vers les nuées dynamiques. L'algorithme des nuées dynamiques est présenté ci-dessous.

6.1.3 Les nuées dynamiques (*k-means*)

Les vecteurs $v_{i,r}$ sont donc classés et regroupés en familles grâce à la méthode des nuées dynamiques (ou centres mobiles). Lorsque les classes ne sont plus suffisamment séparées (distance inter-centre très faible), on change de résolution en passant aux vecteurs $v_{i,r-1}$ et on relance l'algorithme avec une résolution plus fine. Après cette étape de classification, la construction de l'arbre proprement dite peut débuter.

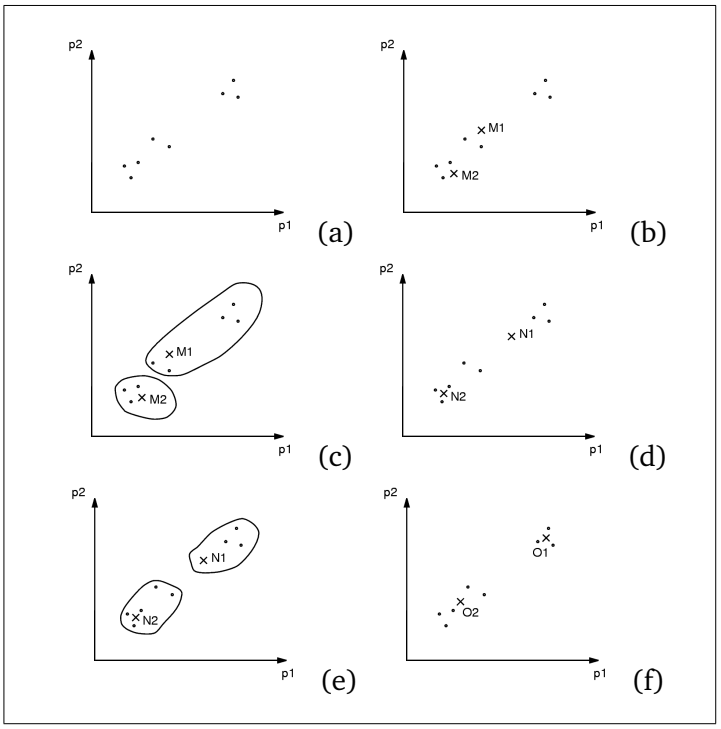


FIG. 6.3 – Un exemple d'utilisation des nuées dynamiques pour séparer automatiquement un espace de deux paramètres en deux familles.

La figure 6.3 illustre un exemple simple d'utilisation des nuées dynamiques. On souhaite séparer un ensemble de vecteurs descripteurs (à deux dimensions) en deux familles (*cluster*). (a) représente l'état initial du système. On a huit vecteurs à classer. Au début de l'algorithme, on choisit deux centres (car on veut obtenir deux familles) et on les place au hasard, ce sont M1 et M2. On affecte les vecteurs au centre le plus proche et on obtient une première séparation en deux familles en (c). En (d), on calcule les deux centres de gravité des familles obtenues, il s'agit de N1 et N2. On affecte à nouveau les vecteurs à la famille du centre le plus proche en (e).

On recalcule les centres de gravité des familles obtenues en (f), les nouveaux centres sont O1 et O2. Si on réitère le procesus, les centres ne bougent plus, ce sont O1 et O2, ils sont stables. On a donc trouvé la solution à notre problème.

L'algorithme des nuées dynamiques possède quelques inconvénients : il est sensible aux points isolés (*outliers*) qui ont tendance à attirer toute une famille vers eux, nous proposons d'ailleurs une méthode permettant de s'affranchir en partie de ce problème (voir le paragraphe 6.4.1.1) ; il est ensuite fortement dépendant des conditions initiales, c'est-à-dire qu'un autre choix de M1 et M2 aurait conduit à une autre classification. Il possède de nombreux avantages : sa convergence vers la solution est très rapide, même pour de grandes familles de vecteurs, il faut en fait peu d'itérations pour arriver à une solution ; il est enfin non-supervisé et nécessite seulement un paramètre : le nombre de classes à définir.

L'algorithme des nuées dynamiques a besoin d'évaluer la distance entre les vecteurs descripteurs, il faut donc choisir une distance métrique à utiliser lors de la phase de classification.

6.2 Algorithmes de tri des images

Pour classer les images, on classe leurs représentants c'est-à-dire les signatures. Cette section donne la démarche suivie pour la classification.

6.2.1 Algorithme d'extraction d'attributs et de création des descripteurs

Maintenant que les outils de traitement d'images et d'extraction d'attributs ont été présentés dans les chapitres précédents, nous allons présenter notre méthode de création de vecteurs descripteurs.

```
pour chaque image de la base I faire
 pour chaque espace couleur E faire
  pour chaque niveau de résolution n faire
   T = ondelette(I,E,n)
   vecteurAttributs[ I ] = extraireAttributs(T)
   finpour
  finpour
  stockerBD(vecteurAttributs[ I ])
 finpour
```

FIG. 6.4 – Algorithme d'extraction des attributs

Dans cette première partie, on lit les images de la base une par une, chacune d'entre elles est transformée en ondelettes, cette transformation fournit une image multirésolution (à n niveaux) dont on extrait un vecteur descripteur contenant tous les attributs extraits à toutes les

résolutions. Ce vecteur descripteur est stocké dans la base de données. Ensuite vient la phase de création des signatures.

6.2.2 Création des signatures

Les signatures sont créées par une organisation hiérarchique des attributs issus des vecteurs descripteurs de l'image sous la forme de quatre vecteurs signature. En fonction des connaissances de l'utilisateur sur une base d'images, il y aura deux possibilités :

- L'expert pourra créer sa propre signature en choisissant les attributs et la façon de les organiser en plusieurs vecteurs.
- l'utilisateur non-expert se verra proposer un certain nombre de signatures (pré-établies par un expert) selon la requête qu'il veut résoudre.

Il n'y a pas d'algorithme à proprement parler pour créer les signatures, il y a juste lecture des attributs dans les vecteurs descripteurs, sélection et organisation par l'expert et écriture dans la base de signature utilisée pour la création de l'arbre de recherche.

6.2.3 Création de l'arbre récursif

Par définition, un **arbre** est un élément récursif n-aire qui possède une racine qui a de 0 à $n-1$ fils, chacun des fils étant un arbre (ayant à son tour de 0 à $n-1$ fils)...

Cette définition récursive permet d'utiliser des algorithmes récursifs de création et de recherche (selon une stratégie : en largeur d'abord, en profondeur d'abord...). Un arbre possède une **racine** qui est la base de l'arbre. Il y a ensuite des **nœuds** non-terminaux (ayant 0 à $n-1$ fils) et des nœuds terminaux (les **feuilles**) n'ayant pas de fils. On définit la **cardinalité** de l'arbre comme le nombre de fils de chaque nœud non-terminal.

La création de l'arbre utilise par définition les propriétés de la récursivité. L'arbre est construit par séparation successives des images en classes à chaque niveau. Plus on descend en profondeur dans l'arbre, moins on trouve d'images dans chacune des classes. La figure 6.5 donne un exemple d'arbre binaire à quatre niveaux. Lors du parcours de l'arbre, on passe d'abord par la racine, puis par les nœuds pour arriver aux feuilles.

6.2.3.1 Choix du critère de changement de niveau de signature

Dans la procédure de classification, on utilise quatre vecteurs signature de taille croissante (4, 6, 8 et 10 attributs par exemple). On utilise d'abord le vecteur le plus court (4 attributs) lorsqu'il y a beaucoup d'images à classer (classement grossier mais rapide). Puis au fur et à mesure de la classification, il y a de moins en moins d'images à classer à chaque niveau de l'arbre. On peut donc affiner la recherche en passant à des vecteurs signature de plus grande taille.

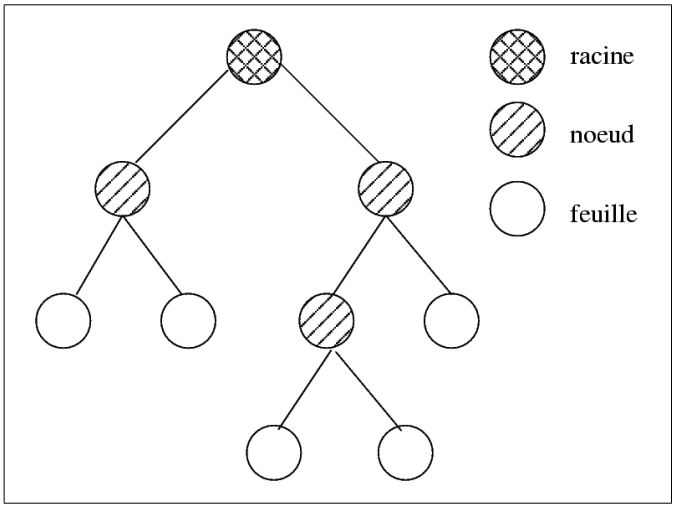


FIG. 6.5 – Un exemple d'arbre binaire de recherche.

La vitesse de création de l'arbre est peu importante puisque les calculs sont effectués hors-ligne. Cependant, pour des tailles raisonnables de bases de données (inférieur à 10 000 images), on peut envisager de réaliser la classification en ligne pour prendre en compte une technique de bouclage de pertinence avec apprentissage. Dans ce cas, il est très intéressant de disposer d'une méthode qui utilise des vecteurs de petite taille en nombres entiers afin d'obtenir des temps de calculs très faibles.

Dans la procédure de création de l'arbre de recherche, il faut définir le moment de changement de vecteur signature et donc le **critère de changement de niveau de signature**.

6.2.3.2 Choix du critère d'arrêt de la construction de l'arbre

Un autre choix important dans la construction récursive d'un arbre est le critère d'arrêt. Il faut définir dans l'appel de la procédure récursive une condition qui, si elle est vérifiée, va mettre un terme à la branche de l'arbre que l'on est en train de construire pour la transformer en feuille de l'arbre.

Le **critère d'arrêt** peut être une mesure qualitative et/ou quantitative sur la branche en cours de traitement.

Par exemple, un critère qualitatif peut être : on découpe le nœud actuel si la distance inter-classe est inférieure à un seuil fixé. Un critère quantitatif peut être : le nombre d'images en dessous duquel on ne découpe plus la famille obtenue. Ou bien on peut envisager un critère mixte : on ne découpe que si la famille contient un nombre donné d'images et que la distance inter-classes est inférieure à un seuil donné.

6.2.3.3 Algorithme de création de l'arbre récursif

L'algorithme de création de l'arbre récursif est donné ci-dessous :

```
Procédure arbre( liste-images, niveau-vecteur-signature, nombre-familles )
 début
  v = vecteur[niveau-vecteur-signature]
  list-images[j] = kmeans( liste-images, v ) // j = 1...nombre-familles
  si critère-changement-niveau est vérifié
  alors
  si niveau-vecteur-signature < 4
  niveau-vecteur-signature = niveau-vecteur-signature + 1
  finsi
 finsi
 si critère d'arrêt n'est pas vérifié
 alors
  pour i de 1 à nombre-familles faire
   arbre( list-images[i], niveau-vecteur-signature )
  finpour
 finsi
fin
```

Dans cette phase de classification, la définition de la cardinalité de l'arbre s'est inspirée de la perception psycho-visuelle. Afin de ne pas surcharger la mémoire à court terme de l'utilisateur du système, la cardinalité de l'arbre a été volontairement limitée. Cette cardinalité suit la règle bien connue des psychologues : 7 ± 2 qui signifie qu'un individu ne peut à un instant donné mémoriser dans sa mémoire à court terme que sept (plus ou moins deux selon les individus) concepts.

Dans notre cas, cela signifie simplement qu'on ne peut pas proposer aux utilisateurs du système plus de sept images pour la recherche car l'utilisateur serait incapable de les comparer correctement à l'image qu'il recherche et qu'il est le seul à visualiser.

Le critère de changement de niveau de signature et le critère d'arrêt seront définis dans la partie implémentation au chapitre 7.

6.3 Outils nécessaires

Afin de mettre en œuvre le principe de recherche décrit précédemment, un certain nombre d'outils vont être nécessaires.

6.3.1 Visualisation de la classification

Puisque le principe de notre approche est d'essayer de travailler avec des informations visuelles, il est nécessaire de disposer d'un outil de visualisation des classes obtenues pour pouvoir les juger. Il y a deux solutions pour visualiser les résultats, soit on utilise un graphique en deux dimensions, soit en trois dimensions. On peut difficilement représenter plus de trois dimensions

sur un écran standard. Une visualisation 2D est rapide à mettre en œuvre mais une visualisation 3D est préférable pour mieux se rendre compte des distances entre les images et entre les classes.

Les vecteurs à représenter possèdent des coordonnées dans un espace à 4, 6, 8 et 10 dimensions. Il est donc nécessaire de disposer d'un outil de projection de ces vecteurs sur un espace de dimension 2 ou 3. L'outil qui permet de réaliser la meilleure projection (au sens des moindres carrés) est l'analyse en composantes principales.

6.3.2 L'analyse en composantes principales (A.C.P.)

Afin de visualiser les résultats de la classification des familles obtenues à l'aide de l'algorithme des nuées dynamiques, il faut réduire la dimensionnalité des vecteurs signature pour un affichage 2D ou 3D. C'est dans un but de visualisation uniquement que nous utilisons l'analyse en composantes principales. Elle ne sert pas lors de la construction de l'arbre de recherche.

L'analyse en composantes principales est un outil mathématique permettant de synthétiser l'information contenue dans un ensemble de données multidimensionnelles. Ainsi, les vecteurs présents dans le tableau de valeurs sont projetés sur des axes principaux, ordonnés suivant l'énergie maximale qu'ils portent. Cette transformation produit une décorrélation maximale des informations portées par chacun des axes principaux. La méthode générale [GW02] est décrite ci-dessous.

Soit un tableau T de n variables (colonnes) et de m individus (lignes). À partir de T, on souhaite construire la matrice R contenant la projection de T sur l'espace dont les axes sont classés par énergie décroissante.

$$T = \begin{pmatrix} t_0(0) & \cdots & t_0(n) \\ t_1(0) & \cdots & t_1(n) \\ \vdots & \vdots & \vdots \\ t_m(0) & \cdots & t_m(n) \end{pmatrix} = \begin{pmatrix} t_0 \\ t_1 \\ \vdots \\ t_m \end{pmatrix} \tag{6.1}$$

Pour effectuer l'A.C.P., il faut utiliser quelques notions de statistiques (issues du cours [35]). La matrice de covariance de T représente la covariance de chaque couple de vecteurs (t_i, t_j).

$$Cov(T) = \begin{pmatrix} cov(t_0, t_0) & \cdots & cov(t_0, t_n) \\ cov(t_1, t_0) & \cdots & cov(t_1, t_n) \\ \vdots & \vdots & \vdots \\ cov(t_n, t_0) & \cdots & cov(t_n, t_n) \end{pmatrix} \tag{6.2}$$

Où $cov(t_i, t_j) = E[(t_i - \mu_i)(t_j - \mu_j)^T]$ est la covariance de t_i et t_j et $\mu_k = E[t_k]$ est l'espérance mathématique (ou moyenne) des composantes du vecteur t_k.

La matrice de corrélation est construite en calculant les corrélations de chaque couple de variables.

$$Corr(T) = \begin{pmatrix} corr(t_0, t_0) & \cdots & corr(t_0, t_n) \\ corr(t_1, t_0) & \cdots & corr(t_1, t_n) \\ \vdots & \vdots & \vdots \\ corr(t_n, t_0) & \cdots & corr(t_n, t_n) \end{pmatrix} \tag{6.3}$$

Où $corr(t_i, t_j) = E[t_i - t_j^T]$ est le coefficient de corrélation de t_i et t_j.

Les matrices $Cov(T)$ et $Corr(T)$ sont des matrices Toepliz (carrées à diagonales constantes).

On définit la matrice T_c comme la matrice T centrée. C'est-à-dire la matrice composée des valeurs de T auxquelles on ôte la moyenne de la variable (colonne).

$$T_c = \begin{pmatrix} t_0(0) - \mu_0 & \cdots & t_0(n) - \mu_n \\ t_1(0) - \mu_0 & \cdots & t_1(n) - \mu_n \\ \vdots & \vdots & \vdots \\ t_m(0) - \mu_0 & \cdots & t_m(n) - \mu_n \end{pmatrix} \tag{6.4}$$

On peut démontrer [GW02] que la projection donnant le meilleur résumé de l'information contenue dans T correspond à la multiplication de T_c par la matrice des vecteurs propres V — triée selon les valeurs propres croissantes — de la matrice de covariance $Cov(T_c)$.

$$R = V\, T_c \tag{6.5}$$

La matrice de covariance de R notée $Cov(R)$ peut être calculée aisément.

$$Cov(R) = V\, Cov(T)\, V^T \tag{6.6}$$

$Cov(R)$ est une matrice diagonale dont les éléments sont égaux aux valeurs propres de T_c classées par ordre décroissant $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_n$. Cette propriété est importante puisque comme les éléments non-diagonaux sont nuls, cette matrice indique que les éléments de R sont décorrélés.

Une autre propriété importante est la reconstruction. Comme V est une matrice réelle symétrique, on a $V^{-1} = V^T$ et la reconstruction est immédiate.

$$T = V^T\, R + \mu_t \tag{6.7}$$

Maintenant, supposons qu'au lieu de conserver toutes les valeurs propres de T_c, on décide de ne conserver que les k plus grandes, alors la reconstruction ne sera plus exacte.

$$\hat{T} = V_k^T\, R + \mu_t \tag{6.8}$$

On peut démontrer que l'erreur δ au sens des moindres carrés entre T et $\hat{T}$ est la somme des $N^2 - k$ dernières valeurs propres.

$$\begin{aligned} \delta &= \sum_{j=1}^{N^2} \lambda_j - \sum_{j=1}^{k} \lambda_j \\ &= \sum_{j=k+1}^{N^2} \lambda_j \end{aligned} \tag{6.9}$$

A partir d'une matrice T, on est capable de construire une matrice R qui synthétise l'énergie de T suivant la précision que l'on désire, précision réglée par la somme des k plus grandes valeurs propres de $Cov(T_c)$.

6.3.3 Les images représentatives des classes

Chaque famille issue de la classification doit être représentée par une (ou plusieurs) **image(s) modèle** qui sera (seront) proposée(s) à l'utilisateur lors de la navigation. Cette (ces) image(s) modèle ou image(s) représentative(s) de la classe doit (doivent) synthétiser au mieux le contenu de la famille qu'elle(s) représente(nt).

Pour construire l'image modèle, il faut donc tenir compte du contenu visuel de la classe. Comme les vecteurs sont regroupés autour du centre de gravité de la classe, les images les plus représentatives de la classe sont proches du centre de gravité (distance la plus faible). L'image modèle doit donc être une image ou une composition d'images proche du centre de gravité de la classe qu'elle représente.

Il existe plusieurs possibilités pour créer ces images modèle : une image modèle unique, plusieurs images modèles, des images réelles, des images artificielles. Comment choisir ? C'est à nouveau l'utilisateur qui va orienter notre décision. Il a été décidé de ne proposer qu'une seule image modèle par famille pour ne pas multiplier le nombre d'images affichées à l'écran lors de la navigation.

En ce qui concerne le débat image réelle ou image artificielle, le principe même de notre système est de travailler sur un regroupement des images en familles visuellement similaires. Il est donc très important que l'utilisateur n'ai pas une idée précise de la famille d'objets qu'on lui présente. L'utilisateur ne doit pas chercher un oursin, il doit chercher un objet de forme arrondie ayant certaines proportions et une certaine texture par exemple.

Le contexte d'utilisation de notre méthode nous oblige à choisir une image artificielle pour la génération des images modèle. Afin de respecter le contexte psycho-visuel des utilisateurs, nous avons finalement opté pour une image modèle composée de la moyenne des trois images les plus proches du centre de gravité de la classe considérée. Ce choix interdit à l'utilisateur d'associer un concept réel à l'image qu'on lui propose (sauf quand les trois images qu'on utilise sont de la même famille et ont les mêmes dimensions). L'utilisateur reste donc dans une recherche visuelle abstraite de la famille d'images qu'il souhaite retrouver.

6.4 Améliorations proposées

L'élaboration de l'architecture de notre système de recherche d'images par la navigation a soulevé un certain nombre de problèmes. Cette partie donne les solutions adoptées pour sinon les résoudre, au moins les atténuer.

6.4.1 L'algorithme des nuées dynamiques

Un effet de bord très connu des nuées dynamiques est que les points isolés (*outliers*) ont tendance à attirer vers eux les centres de classes et donc à dénaturer le résultat de la classification. Un autre problème connu est la séparation stricte des classes qui pénalise grandement les images en périphérie de classe.

6.4.1.1 Traitement des points isolés

Pour résoudre le problème des **points isolés** (*outliers*), il faut essayer de les détecter avant le début de la classification. Pour réaliser cette détection, on effectue un calcul assez simple. On calcule le centre de gravité des signatures. On calcule ensuite la distance entre chaque signature et ce centre de gravité pour obtenir un classement des points les plus éloignés du centre. Ces points situés loin du centre de gravité de la classe serviront comme noyaux lors du calcul des nuées dynamiques.

Ainsi, si ce sont des points isolés, ils resteront isolés dans les différentes itérations des nuées dynamiques et n'influenceront pas les autres classes. Sinon, ils participeront à la classe comme tous les autres points de celle-ci.

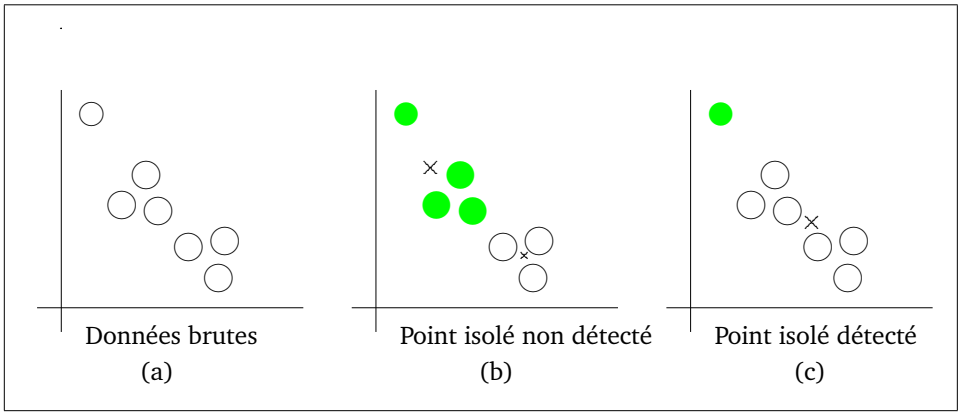


FIG. 6.6 – Traitement des points isolés.

La figure 6.6 illustre cet effet. Les données brutes sont représentées en (a). Si le point isolé n'est pas détecté, il va attirer à lui la classe la plus proche. Si il est détecté par notre technique,

l'un des noyaux de l'algorithme sera initialisé en ce point et il constituera une famille à lui tout seul. Ainsi, à l'itération suivante, la classification se poursuivra sans l'influence du point isolé.

La mise en œuvre de cette amélioration de la méthode des nuées dynamiques est illustrée en figure 6.7. Cette figure est issue de la classification d'une sous-famille de la base Columbia. La classification a nécessité dix itérations pour obtenir le résultat final. La figure montre la première et la dernière itération. Les centres ont été initialisés aux points les plus éloignés du centre de gravité de l'ensemble des données. La classe verte ne contient au départ qu'un seul point qui reste seul dans sa classe pendant tout le processus de classification.

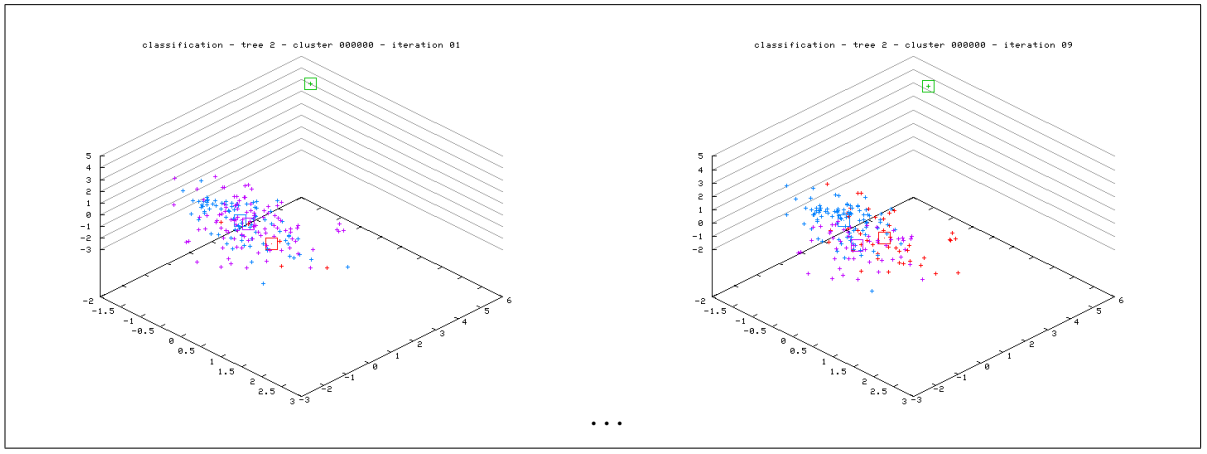


FIG. 6.7 – Exemple réel de l'amélioration par initialisation des centres aux points les plus éloignés du centre de gravité des données.

6.4.1.2 Classification floue

À la fin de la classification par les nuées dynamiques, on obtient des classes strictes, c'est-à-dire que chaque image n'appartient qu'à une seule classe. Les images situées en périphérie d'une classe, près de la frontière d'une autre classe, posent le problème de l'appartenance stricte à une classe unique.

Il serait intéressant d'introduire une notion plus floue d'appartenance à une classe pour permettre aux images placées entre deux classes d'appartenir un peu à chacune d'elle. Afin de rendre la **classification floue**, on va effectuer après la classification un calcul de **degré d'appartenance** d'une image à une classe. Le calcul du degré d'appartenance est décrit en détails en annexe B.

Soit un point M des données classées et soient d_i, $i = 1 \dots k$ les distances du point M à chacun des k centres des classes. Le degré d'appartenance du point M à la classe i est défini par :

$$deg_{M \in i} = \frac{100}{\sum_{p=1}^{k} \frac{d_i}{d_p}}$$

Grâce au calcul du degré d'appartenance (qui est compris entre 0 et 100), on a un moyen de rendre la classification floue. On peut par exemple décider que l'image appartient à toutes les classes pour lesquelles son degré d'appartenance est supérieur à 30 (**seuil d'appartenance**).

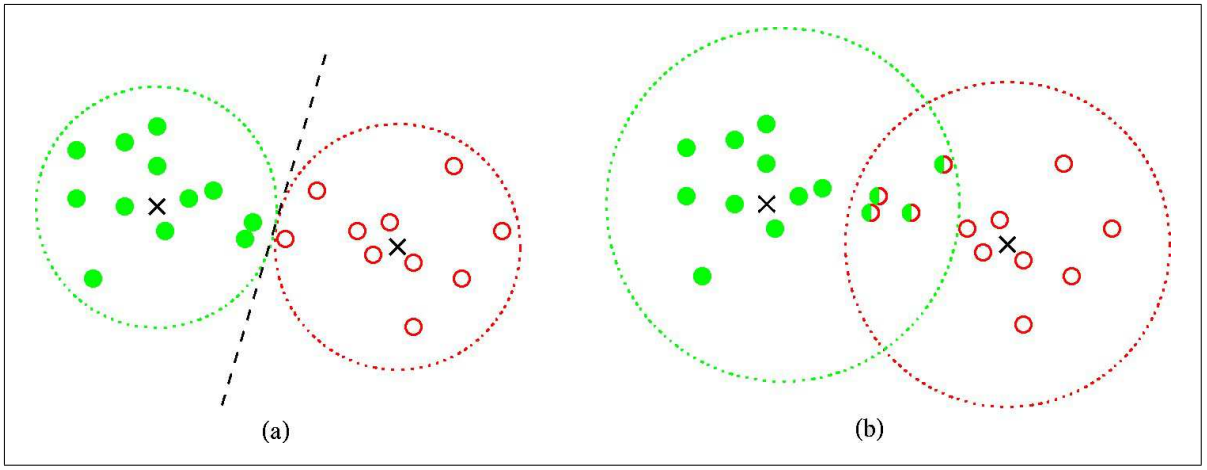


FIG. 6.8 – Introduction de flou dans la classification. (a) arbre strict, (b) arbre flou.

La figure 6.8 illustre le niveau de flou induit dans la classification. En (a), la classification est stricte et les points éloignés du centre de gravité de chacune des familles sont classés dans une seule famille. En (b), la classification est floue, on autorise certaines images à appartenir à plusieurs familles. On augmente ainsi le temps de calcul de l'arbre, mais on assouplit la rigidité de la classification stricte pour permettre à l'utilisateur de se tromper un peu plus dans le chemin qu'il suit dans l'arbre.

Il existe une version floue de l'algorithme des nuées dynamiques (*fuzzy C-means*) utilisée dans le contexte de la recherche d'image par Lambert et al. [LG01]. Cet algorithme revient à calculer un degré d'appartenance flou dont le flou est réglé par un paramètre. Cette méthode est semblable à la nôtre et nécessite de choisir un degré d'appartenance minimal (seuil d'appartenance) pour qu'une image appartienne à une classe.

Le choix du seuil d'appartenance à une classe dépend d'une part du nombre de classes et d'autre part de la répartition statistiques des données. Le choix du seuil nécessite donc d'effectuer des tests sur la base d'images pour trouver la valeur optimale pour assurer un niveau de flou efficace lors de la recherche.

6.4.2 Construction de l'image représentative

Lors de la construction de l'image modèle des classes, il serait souhaitable de prendre en compte le niveau de résolution d'ondelettes auquel on effectue la classification. Ainsi, plus on utilisera une résolution grossière, plus l'image modèle sera grossière et plus on ajoutera des détails dans la recherche, plus l'image modèle contiendra de détails.

Ce chapitre vient de poser clairement toutes les bases de l'approche retenue dans ce mémoire. Il faut maintenant passer aux choix des techniques numériques les plus appropriées pour répondre au cahier des charges qu'on vient de définir.

Le chapitre suivant est consacré à la mise en œuvre informatique de notre système de recherche d'images par la navigation visuelle.

Chapitre 7

Implémentation

La conception d'un système de recherche d'images nécessite l'utilisation de nombreux outils informatiques. Ce chapitre présente l'architecture et les choix d'implémentation retenus.

7.1 Description du système

7.1.1 Schéma du système

Notre système de recherche d'images par le contenu [LTMD01] est volontairement orienté réseau pour permettre d'effectuer une recherche sur une base d'images à partir de n'importe quel navigateur Internet. La volonté a été de créer des pages web dynamiques qui réagissent aux actions de l'utilisateur.

Le système se compose donc d'un serveur web (Apache) qui gère des pages web dynamiques (PHP) capables d'accéder à des données d'une base de données relationnelle (MySQL). Les clients se connectent au système par le protocole HTTP et interagissent avec lui à l'aide de formulaires HTML. Le système est capable de gérer plusieurs connexions simultanées et les accès concurrents à la base de données.

Le stockage des images est réalisé sur le serveur dans un répertoire spécial. On ne stocke pas les images dans la base de données mais seulement leur URI, c'est-à-dire l'adresse absolue Internet où on peut les télécharger. Seuls les descripteurs d'images et les URI des images sont stockés dans la base. Lorsque le système a besoin d'une image (pour l'affichage par exemple),

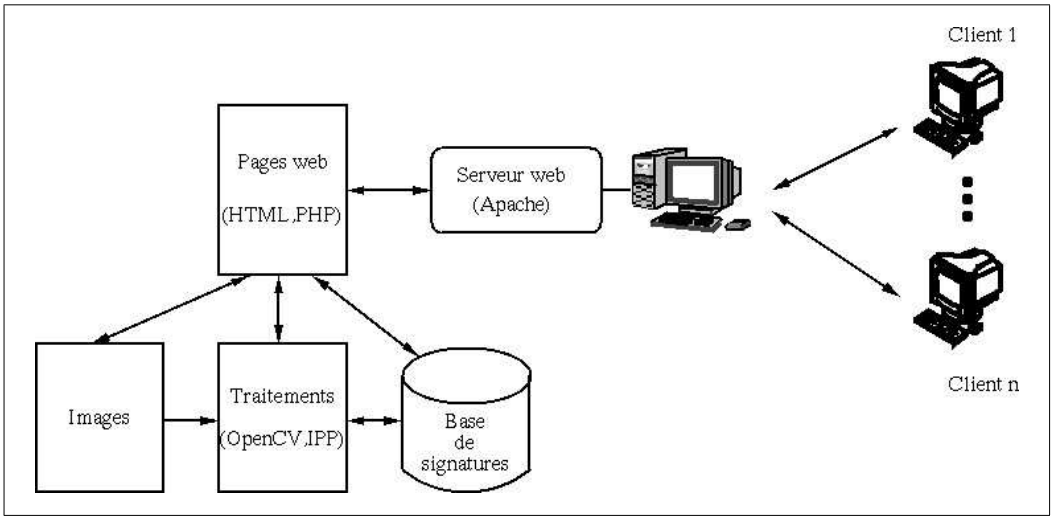


FIG. 7.1 – Architecture du système de recherche d'images.

PHP la charge et l'affiche dans la page web générée. La figure 7.1 présente l'architecture de notre système.

7.1.2 Serveur Web Apache

Le serveur Web Apache [28] est le serveur le plus utilisé au monde avec plus de 63% [29] de part de marché. C'est un logiciel libre dont les sources sont disponibles gratuitement. Le projet Apache prévoit de fournir un serveur efficace, sécurisé et extensible conforme aux dernières recommandations du protocole HTTP selon le consortium W3C [32].

Apache fournit des pages web à la demande des clients. Grâce à des modules d'extension, on peut lui ajouter des fonctionnalités supplémentaires et le coupler avec des langages de scripts côté serveur très performants.

7.1.3 Langage de scripts côté serveur PHP

Parmi les extensions proposées par Apache, on trouve le PHP [17] qui est un langage de génération de pages web dynamiques côté serveur. Ce langage définit des instructions exécutées par le serveur web lorsqu'une demande de page provient d'un client. Ainsi, l'interface de notre système sera écrite à l'aide de pages web qui réagiront en fonction des actions de l'utilisateur. Le langage PHP est inspiré du langage C dans sa syntaxe et possède deux fonctionnalités très intéressantes. D'une part il peut exécuter des programmes externes sur le serveur, d'autre part il permet d'interagir avec une base de données.

La première fonctionnalité va nous permettre d'effectuer tous les traitements sur les images à l'aide de programmes externes écrits en C. Et la seconde fonctionnalité va nous permettre de

stocker et d'interroger la base de données contenant nos signatures d'images.

7.1.4 Base de données relationnelle MySQL

Au début du développement du système, une solution basée sur XML a été testée, mais elle posait de nombreux problèmes d'interfaçage avec le code C (malgré la librairie XML2). Cette solution a été abandonnée au profit d'un système de gestion de bases de données relationnelles.

La base de données relationnelle MySQL [2] est une base de données *opensource* qui offre de nombreuses fonctionnalités conformes au standard SQL ANSI 1999 pour un système de gestion de bases de données relationnelles. Cette base de données est gratuite mais propose une maintenance payante pour les entreprises qui ont besoin d'un support technique.

Ses performances sont excellentes et les tests de rapidité donnés sur le site sont très avantageux pour cette base de données par rapport aux concurrents que sont Oracle, DB2 d'IBM, Access et SQL Server de Microsoft. Grâce au module d'accès à la base de données, PHP peut interagir avec MySQL au travers d'une page web.

7.1.5 Traitement d'images

Les premiers essais de la méthode ont été réalisés sous Matlab [21] en raison de sa facilité de développement et de ses possibilités de traitement d'images. Les temps de calculs étaient assez bons pour de petites bases d'images (moins de 1000 images), mais les performances se dégradaient rapidement avec l'augmentation de la taille de la base. Matlab a démontré la faisabilité de la méthode mais une solution plus rapide a été préférée pour le démonstrateur final.

Pour effectuer les divers traitement d'images associés à nos images, nous avons choisi des programmes écrits en C qui utilisent deux librairies de traitement d'images fournies par Intel Corporation. La première librairie nommée IPP [34] (Integrated Performance Primitives) est une librairie de traitement d'images bas niveau optimisée pour les processeurs Intel. La seconde librairie est OpenCV [33] (Open Computer Vision) qui offre des fonctions de traitement d'images haut niveau. Ces deux librairies sont décrites en annexe de ce document.

7.1.6 Détails de l'implémentation

7.1.6.1 Les espaces couleurs

Comme on l'a vu dans le chapitre 4 de création des descripteurs, l'information couleur est importante pour les recherches des utilisateurs dans une base d'images en couleurs.

Il est donc nécessaire d'utiliser des espaces couleurs bien adaptés au calcul de distance entre couleurs. Ce travail utilise les espaces couleurs HSV et Lab qui possèdent des propriétés très intéressantes pour extraire des attributs représentatifs.

7.1.6.2 Extraction d'attributs

Comme on l'a déjà vu, la comparaison directe des images entre elles n'est pas envisageable. Il est donc nécessaire d'en extraire au préalable des informations représentatives : les attributs. Ces attributs sont des mesures de caractéristiques de l'image qui doivent être invariantes en rotation, translation et changement d'échelle. Cette invariance permet d'ignorer les transformations non pertinentes que peut avoir subi l'image.

Les attributs utilisés dans ce travail sont issus d'une transformation des images par les ondelettes *lifting scheme* en nombres entiers proposées par Calderbank et al. [CDSY98]. Tous les attributs sont donc des nombres entiers. Le codage des images avant transformation est un codage à un octet non signé (valeurs entre 0 et 255). Le lifting scheme en nombres entiers utilisé modifie la plage de codage et peut renvoyer des valeurs positives ou négatives qui ne peuvent pas être codées sur un octet. Il est donc nécessaire d'utiliser un codage plus adapté, à savoir le codage sur deux octets signés (valeurs de -32767 à +32768). Il y a donc passage lors de la transformation d'une image huit bits à une image seize bits.

7.1.6.3 Les bases d'attributs

Les attributs une fois calculés sont stockés dans la base des attributs sous la forme d'une chaîne de caractères (type BLOB) qui contient une liste d'images séparées par des point-virgules (« ;»). Cette astuce de programmation permet d'avoir pour chaque image les attributs qui la concernent en une seule lecture de bloc sur le disque dur, ce qui est primordial pour une optimisation du temps de réponse du système de recherche.

La base de signature possède la même structure de chaîne de caractères pour la même raison puisqu'elle est une sous-famille de la base d'attributs.

Cette organisation en chaîne de caractères a été retenue après le constat que l'utilisation d'un ensemble d'enregistrements d'attributs par image causait des ralentissements conséquents dans la phase de traitement des données. En effet, pour écrire (ou lire) un vecteur à dix composantes, dix accès disques sont nécessaires contre un seul si tous les attributs sont stockés dans le même enregistrement.

Dans la mesure où on ne connaît pas, a priori, le nombre d'attributs extraits, la chaîne de caractères (type BLOB) est de plus la meilleure solution pour stocker les attributs.

7.1.6.4 Les arbres de navigation

Lors de la construction de l'arbre de recherche, l'expert ajoute la base d'images au système. Il demande l'analyse en ondelettes des images dont sont extraits les attributs. Ces attributs stockés dans la base sont ensuite organisés en quatre vecteurs signature selon l'avis de l'expert. A partir de ce moment, un utilisateur non-spécialiste peut tout à fait utiliser le système car il

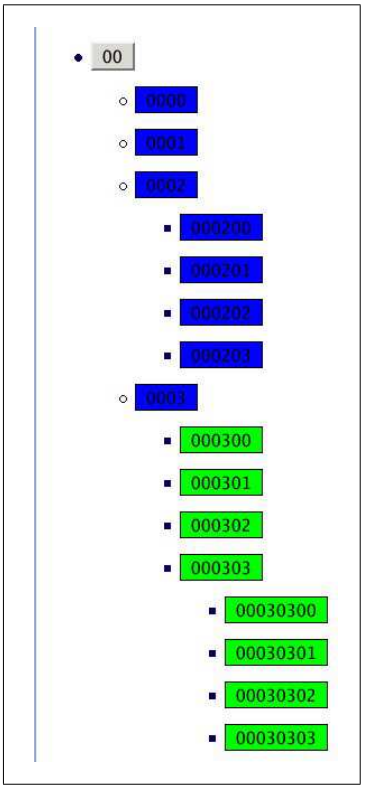


FIG. 7.2 – Exemple d'arbre de recherche avec la base de 60 images tyfipal60. Les couleurs indiquent le niveau de signature utilisé : bleu pour le niveau -1 et vert pour le niveau -2.

existe au moins une signature (celle proposée par l'expert) dans la base. L'utilisateur du système choisit la cardinalité (i.e. le nombre de fils de chaque nœud) de l'arbre et le seuil du nombre d'images à partir duquel on ne découpe plus la famille. La classification construit récursivement l'arbre. L'utilisateur peut naviguer dans l'arbre.

Un exemple d'arbre de recherche sur la base de 60 images tyfipal60 est montré en figure 7.2. Dans cet arbre, l'utilisateur a choisi un arbre quaternaire (*quadtree*). La racine de l'arbre est numérotée "00" et possède quatre nœuds fils : "0000", "0001", "0002" et "0003". Les nœuds "0002" et "0003" possèdent à leur tour quatre fils... Les couleurs permettent de distinguer la nature des vecteurs signature utilisés. Le bleu indique qu'on utilise la signature la plus grossière (niveau de signature -1) et le vert indique un changement de niveau de signature (-2) sous l'effet de la vérification du critère de changement de niveau de signature.

Les figures 7.3 à 7.7 donnent le contenu de la racine (toute la base) et des familles (découpages successifs) au premier niveau de l'arbre.

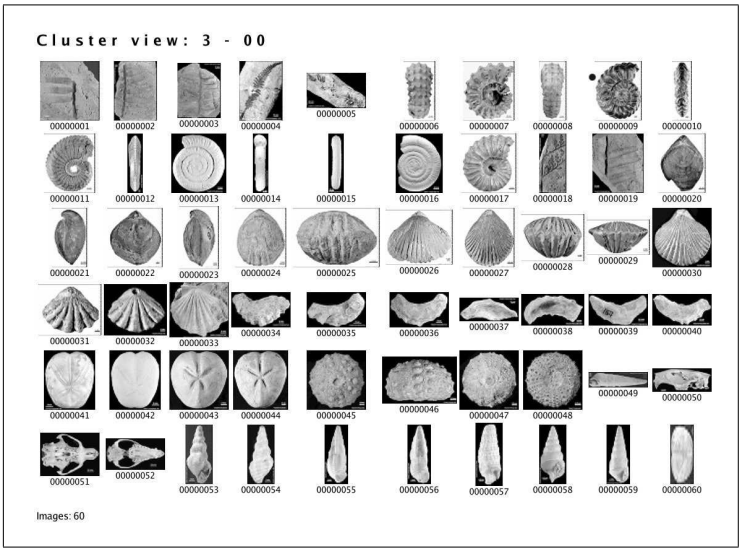


FIG. 7.3 – Images du cluster "00".

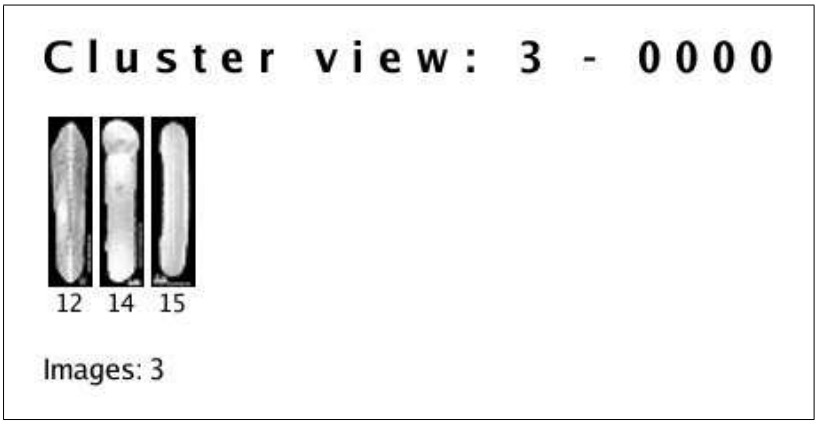


FIG. 7.4 – Images du cluster "0000".

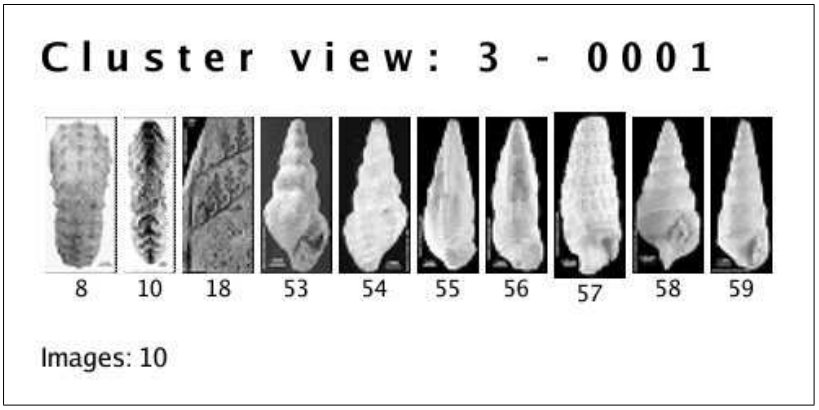


FIG. 7.5 – Images du cluster "0001".

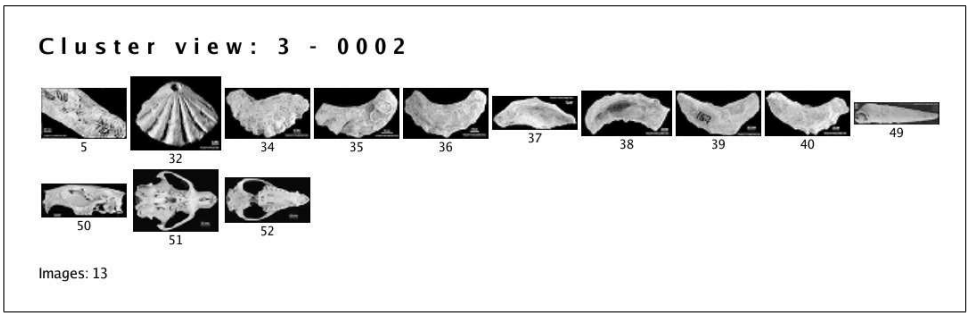


FIG. 7.6 – Images du cluster "0002".

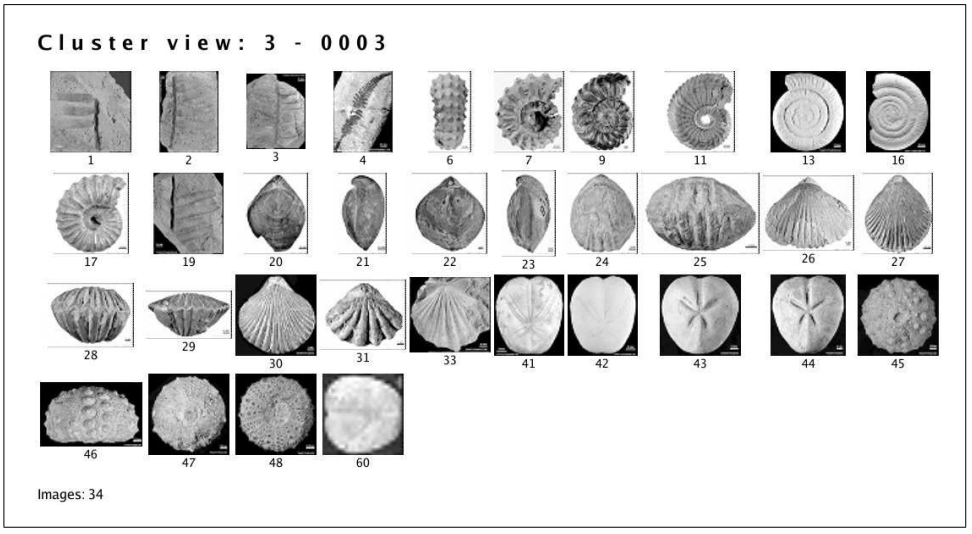


FIG. 7.7 – Images du cluster "0003".

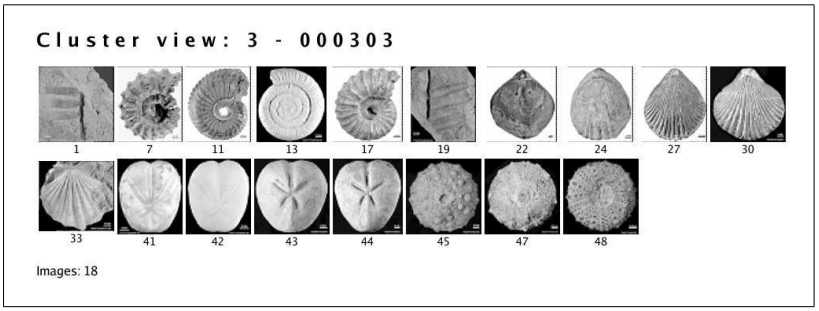


FIG. 7.8 – Images du cluster "000303".

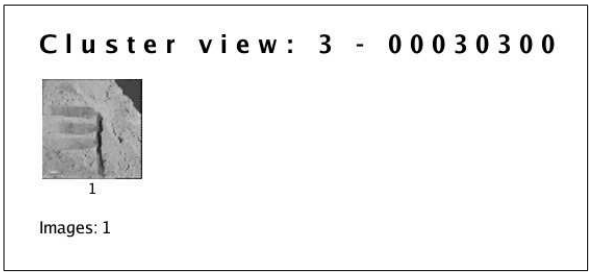


FIG. 7.9 – Images du cluster "00030300".

Une vue très intéressante est celle de la famille "00030300" (figure 7.7) qui ne compte qu'un individu. C'est l'exemple typique d'un point isolé détecté grâce au pré-calcul et à l'initialisation des noyaux sur les points les plus éloignés du centre de gravité du nuage de données.

7.1.6.5 L'image représentative

Le calcul de l'image représentative est effectué en moyennant pour chaque famille à représenter les trois images les plus proches du centre de gravité de la famille au niveau de résolution d'ondelettes auquel on se trouve. Cette méthode vise à produire une image modèle qui ne soit pas directement celle d'un objet réel afin que l'utilisateur du système ne lui associe pas forcément un concept ou une famille existante : oursin, coquillage... Au contraire, l'utilisateur doit chercher quelque chose de flou, d'abstrait.

L'image 7.10 montre quelques exemples d'images représentatives des familles d'images présentées à la section précédente. La figure 7.11 présente quelques autres images modèles issues des tests effectués.

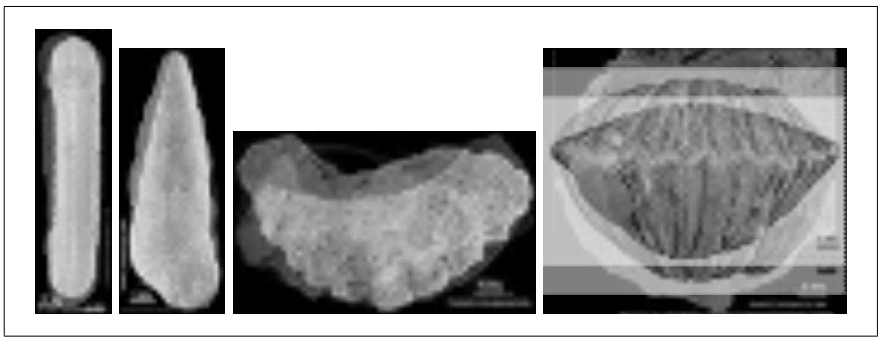


FIG. 7.10 – Images modèles des familles "0000", "0001", "0002" et "0003" respectivement.

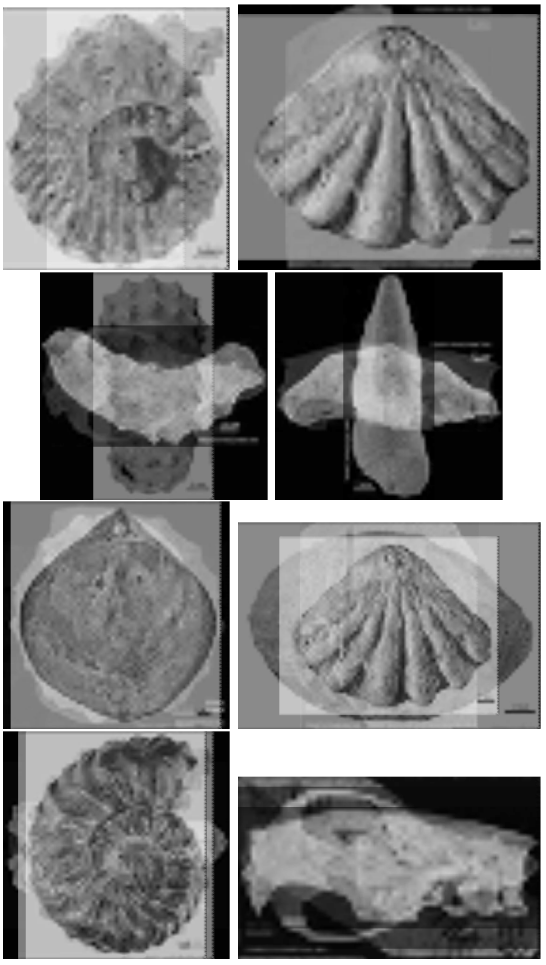


FIG. 7.11 – Exemples d'images modèles.

7.1.7 Visualisation des familles

Comme on l'a déjà souligné, la visualisation des résultats de la classification est primordiale pour que l'expert puisse se rendre compte de la pertinence des classes, de leur distance les unes par rapport aux autres et de leur homogénéité visuelle.

7.1.7.1 Visualisation 2D

La première forme de visualisation adoptée a été une visualisation en deux dimensions. Cette visualisation utilise les coordonnées des vecteurs signature projetés sur un plan 2D normalisé (coordonnées comprises entre 0 et 1) par l'analyse en composantes principales. Le calcul est effectué par un programme en C qui utilise les librairies IPP et OpenCV pour créer une image JPEG contenant la visualisation demandée. Ce calcul est effectué côté serveur et l'image résultat est envoyée au client pour visualisation dans une page web.

Quelques illustrations de cette vue 2D du résultat sont données ci-dessous. La figure 7.12 montre une vue de la classification de la base trans'tyfipal (727 images). La figure 7.13 montre un exemple de vue 2D de la base Columbia (7200 images).

L'image calculée est fixe et il est nécessaire de reprendre tout le calcul si on souhaite modifier le point de vue. C'est une contrainte puisqu'on souhaite pouvoir se déplacer librement dans l'espace des signatures classées afin d'observer les résultats associés à chaque choix de signature.

7.1.7.2 Visualisation 3D

Bien que la visualisation 2D donne de bons résultats, elle nécessite que la vue obtenue soit recalculée à chaque fois pour un grossissement ou un déplacement dans une zone spécifique de la vue. Il est apparu beaucoup plus intéressant de travailler avec une scène 3D calculée une fois pour toute dans laquelle on peut ensuite librement se déplacer (en rotation, translation et grossissement).

Pour réaliser cette vue 3D, l'extension 3D pour java de Sun Microsystems nommée Java3D a été utilisée. Java3D offre toutes les fonctions nécessaires à l'affichage 3D de scènes et à leur exploration à l'aide de la souris. Le calcul de la scène est beaucoup plus long et plus gourmand en mémoire que le calcul de la vue 2D. Mais après affichage, l'utilisation est beaucoup plus facile et les déplacements, rotations, translations et mises à l'échelle sont commandés simplement grâce à la souris.

De plus, Java3D est fourni gratuitement par Sun (comme une extension du *Java Development Kit*) et est très facile à installer. Comme Java, Java3D est intégrable très simplement dans une page web sous la forme d'une *applet* Java3D. Le calcul est effectué côté client et est assez long en raison du transfert, du serveur vers le client, des images pour la construction de la scène 3D.

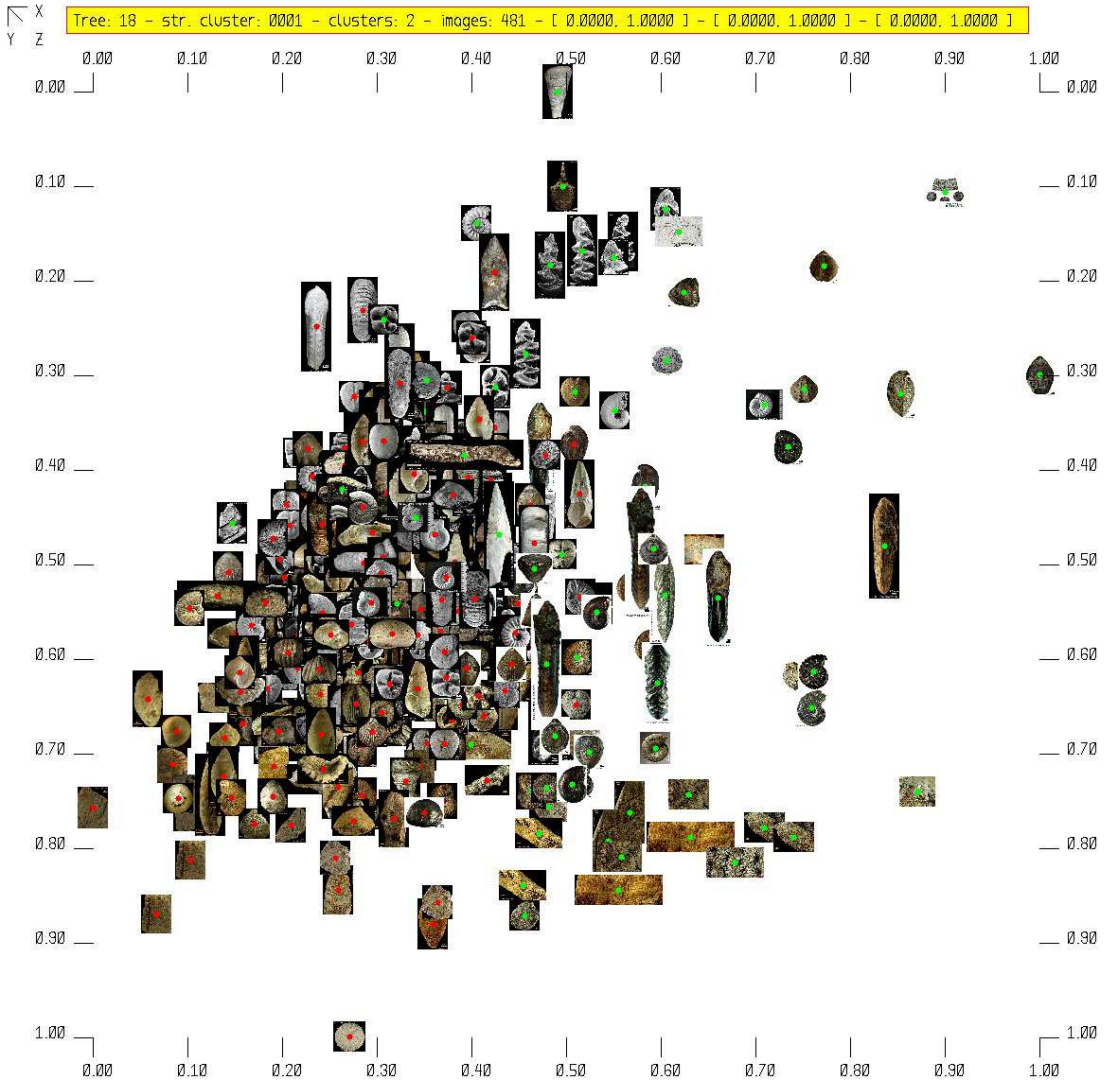


FIG. 7.12 – Exemple de regroupement d'images en deux familles, base tyfipal (727 images), intervalle x $\in$ [0 ;1] et y $\in$ [0 ;1]. L'appartenance d'une image à l'une des deux classes est indiquée par le point au milieu de l'image : rouge pour la classe 1 et vert pour la classe 2. Cette représentation ne permet pas une visualisation optimale des classes obtenues.

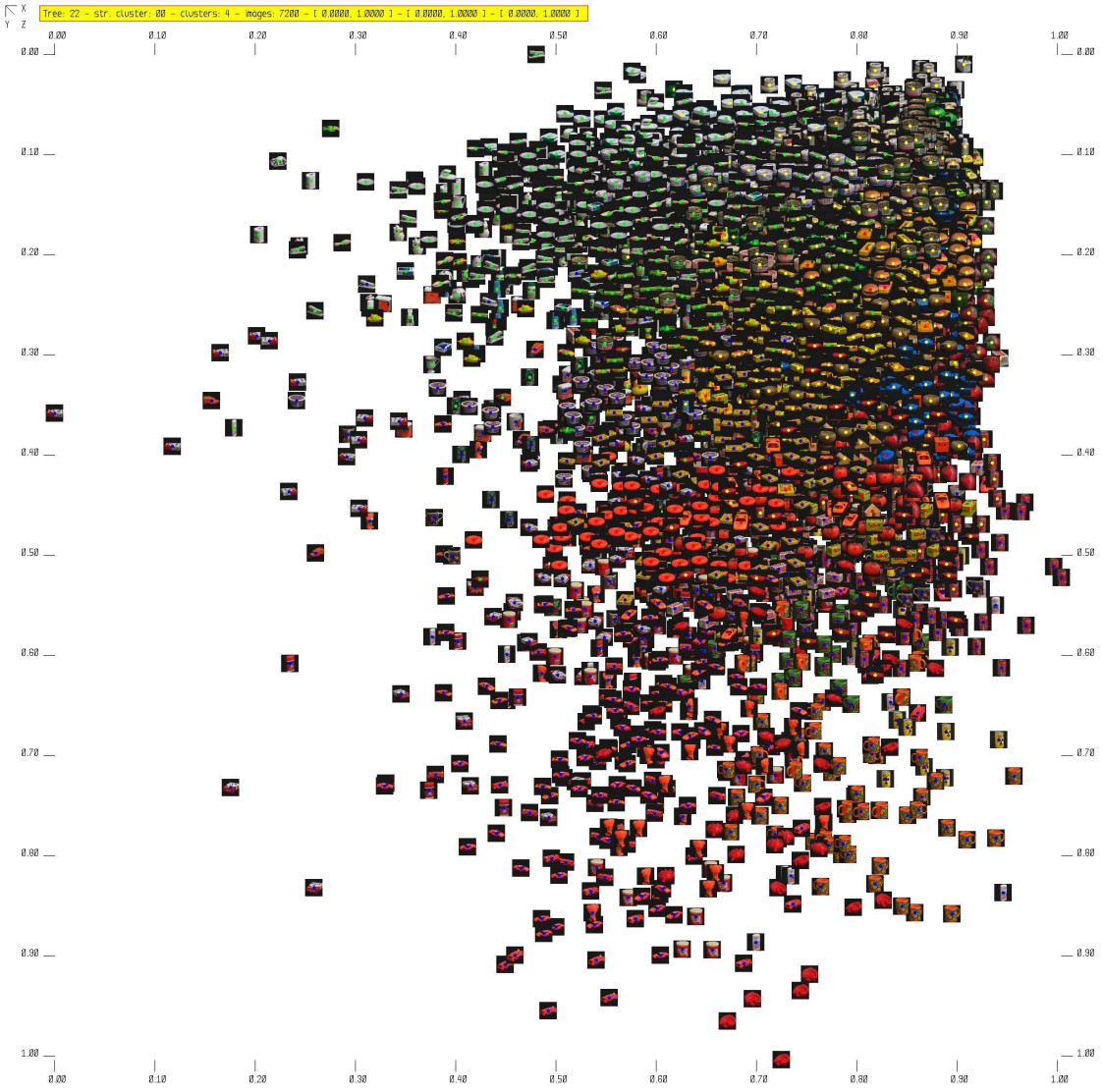


FIG. 7.13 – Exemples de regroupement d'images avec la base Columbia (7200 images).

7.1.7.3 Exemple de visualisation 3D d'un arbre de recherche

Les figures 7.14 et 7.15 sont des copies d'écran du navigateur Internet (ici Mozilla Firefox) de l'utilisateur de la base. On peut se déplacer dans la scène 3D à l'aide de la souris grâce aux fonctions Java3D. Cette interface permet de grossir, tourner, déplacer dans la vue, on peut donc avoir une vue globale et locale du résultat de la classification.

Les différentes familles obtenues sont repérées par leur couleur parmi rouge, vert, bleu et bleu ciel. Les centres des familles sont représentés par des carrés de couleur unie. On peut ainsi ce rendre compte visuellement de la distance d'une image par rapport au centre de sa famille.

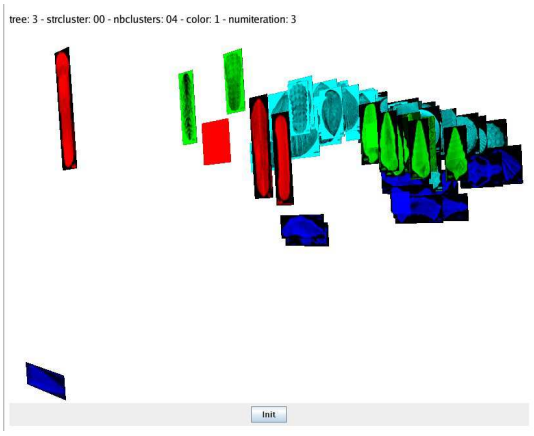


FIG. 7.14 – Visualisation Java3D de la classification exemple de soixante images (vue de face).

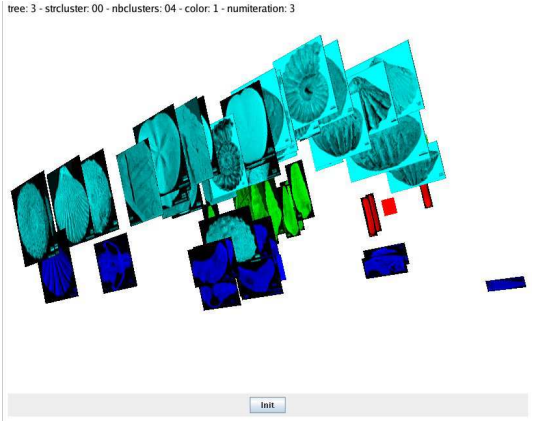


FIG. 7.15 – Visualisation Java3D de la classification exemple de soixante images (vue précédente tournée d'environ 180 degrés).

7.1.7.4 Exemple de visualisation 3D de la classification

Afin de mieux étudier les classes obtenues lors de la phase de classification, une interface de visualisation a été créée. Cette interface utilise la page web comme support de visualisation. Il suffit à l'utilisateur du système de donner l'arbre et la famille à visualiser, la page web en PHP génère automatiquement les images de la classification pour chacune des itérations des nuées dynamiques. On a donc un suivi spatial des classes au cours de la classification.

A partir des paramètres de l'utilisateur, les données sont insérées aumatiquement dans le logiciel gnuplot [18] qui trace l'espace 3D de classification. L'image obtenue est sauvée au format "png", puis traduite au format "gif" pour affichage dans la page web de visualisation. La figure 7.16 donne un exemple de tracé obtenu avec la base d'images columbia (7200 images). Cette image favorise l'étude de la dispersion des classes dans l'espace de classification.

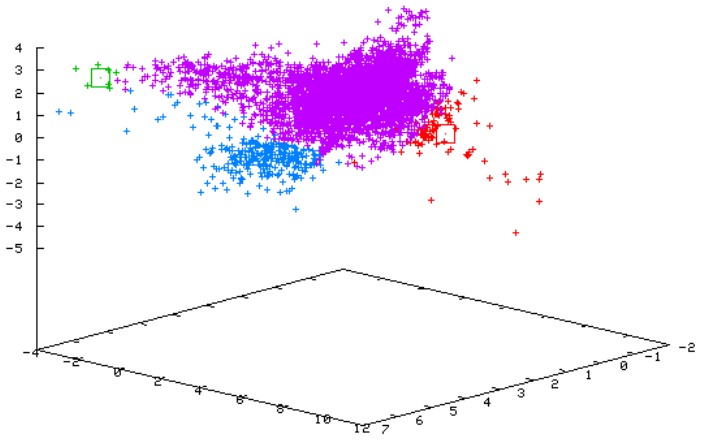


FIG. 7.16 – Vue de la classification de la base Columbia (7200 images). Ce tracé a été réalisé automatiquement avec Gnuplot.

7.2 Modèle de données

La création d'un modèle de données pour le stockage et l'interrogation de la base d'images est une étape importante du développement d'un système de recherche d'images. Comment sont stockées les informations, que contiennent-elles, comment les organiser. . . ?

7.2.1 Schéma entité-association

L'analyse du problème à traiter commence par la définition des objets qui apparaissent et des liens pertinents qui existent entre eux pour la résolution du problème. Cette partie présente le schéma entité-association créé à partir des hypothèses du problème.

Pour fonctionner, notre base d'images doit contenir un certain nombre d'objets :

– Les **images** sont les objets principaux de notre base,
– La **collection** d'images est un ensemble d'images,
– L'**analyse** correspond à l'extraction des descripteurs des images transformées en ondelettes,
– Les **signatures** proviennent de l'organisation hiérarchique des descripteurs,
– Les **arbres** sauvegardent les données créées hors-ligne pour la navigation en ligne.

La figure 7.17 représente le schéma entité-association simplifié de notre base de données. On y retrouve les éléments figurant ci-dessus.

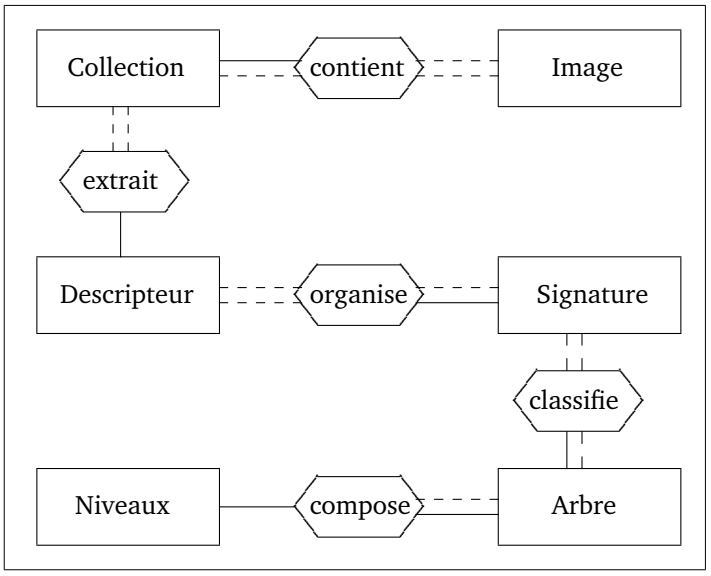


FIG. 7.17 – Schéma entité-association (simplifié) de notre modèle de données.

7.2.2 Schéma relationnel

Le schéma relationnel retenu dérive directement du schéma entité-association ci-dessus. Le choix a été de privilégier les performances du système plutôt que la mise à jour de la base de données d'images. Ainsi, la représentation des vecteurs descripteur et signature est définie par niveau d'ondelettes, mais le contenu des vecteurs proprement dit est stocké dans une chaîne de caractères.

Le schéma relationnel (simplifié) défini dans MySQL est donné ci-dessous :

```
create table collection(
numcollection INT( 6 ) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT,
namecollection VARCHAR( 255 ) NOT NULL,
nbimage INT( 8 ) UNSIGNED ZEROFILL NOT NULL,
url VARCHAR( 255 ) NOT NULL,
PRIMARY KEY( numcollection ));

create table image(
numimage INT( 8 ) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT,
numcollection INT( 6 ) UNSIGNED ZEROFILL NOT NULL,
url VARCHAR( 255 ) DEFAULT '' NOT NULL,
PRIMARY KEY( numimage, numcollection ));

create table analysis(
numanalysis INT( 4 ) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT,
numcollection INT( 6 ) UNSIGNED ZEROFILL NOT NULL,
nbwaveletlevel INT( 2 ) UNSIGNED ZEROFILL NOT NULL,
namewavelet VARCHAR( 30 ) NOT NULL,
fillvalue INT( 1 ) NOT NULL,
segmentation INT( 2 ) NOT NULL,
namecolorspace INT( 2 ) NOT NULL,
PRIMARY KEY( numanalysis ));

create table analysisfeature(
numanalysis INT( 4 ) UNSIGNED ZEROFILL NOT NULL,
numimage INT( 8 ) UNSIGNED ZEROFILL NOT NULL,
numwaveletlevel INT( 2 ) UNSIGNED ZEROFILL NOT NULL,
value TEXT NOT NULL,
PRIMARY KEY( numanalysis, numimage, numwaveletlevel ));

create table analysisfeaturenameweights(
numanalysis INT( 4 ) UNSIGNED ZEROFILL NOT NULL,
numwaveletlevel INT( 2 ) UNSIGNED ZEROFILL NOT NULL,
nbfeature INT( 4 ) NOT NULL,
namefeature TEXT NOT NULL,
weights TEXT NOT NULL,
PRIMARY KEY( numanalysis, numwaveletlevel ));

create table signature(
numsignature INT( 4 ) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT,
numanalysis INT( 4 ) UNSIGNED ZEROFILL NOT NULL,
filesignature VARCHAR( 255 ) NOT NULL,
PRIMARY KEY( numsignature ));

create table signaturefeature(
numsignature INT( 4 ) UNSIGNED ZEROFILL NOT NULL,
numimage INT( 8 ) UNSIGNED ZEROFILL NOT NULL,
numfeaturelevel INT( 2 ) UNSIGNED ZEROFILL NOT NULL,
value TEXT NOT NULL,
PRIMARY KEY( numsignature, numimage, numfeaturelevel ));
```

```
create table signaturefeaturenameweights(
numsignature INT( 4 ) UNSIGNED ZEROFILL NOT NULL,
numfeaturelevel INT( 2 ) UNSIGNED ZEROFILL NOT NULL,
nbfeature INT( 4 ) NOT NULL,
namefeature TEXT NOT NULL,
weights TEXT NOT NULL,
PRIMARY KEY( numsignature, numfeaturelevel ));

create table tree(
numtree INT( 4 ) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT,
numsignature INT( 4 ) UNSIGNED ZEROFILL NOT NULL,
arity INT( 2 ) UNSIGNED ZEROFILL DEFAULT '02' NOT NULL,
nbminimage INT( 3 ) UNSIGNED ZEROFILL DEFAULT '010' NOT NULL,
PRIMARY KEY( numtree ));

create table treelevels(
numtree INT( 4 ) UNSIGNED ZEROFILL NOT NULL,
numcluster VARCHAR( 255 ) NOT NULL,
listimages TEXT NOT NULL,
treelevel INT( 4 ) NOT NULL,
arity INT( 2 ) NOT NULL,
distmin FLOAT,
distmax FLOAT,
distmean FLOAT,
diststddev FLOAT,
PRIMARY KEY( numtree, numcluster ));
```

La table collection contient les ensembles d'images, elle est liée à la table image par le numéro de collection *numcollection*. La table *analysis* donne des informations sur le type d'analyse en ondelettes réalisée et les tables *analysisfeature* et *analysisfeaturenameweights* précisent les valeurs d'attributs ainsi que leurs noms et leurs pondérations respectives.

La table *signature* renseigne sur l'organisation des attributs provenant de *analysisfeature*. Les tables *signaturefeature* et *signaturefeaturenameweights* précisent les valeurs des attributs contenues par les vecteurs signature ainsi que les noms des attributs et les pondérations qu'ils contiennent.

Enfin les tables *tree* et *treelevels* renferment des informations sur l'arbre construit et le contenu de chacun des nœuds de l'arbre.

7.3 Complexité et temps de calcul

Bien que les calculs soient réalisés hors-ligne, leur optimisation est intéressante pour garantir au système une utilisation simultanée par de nombreux utilisateurs ainsi que pour rendre réaliste un calcul de l'arbre de recherche en ligne pour une base d'images de moins de 10 000 images dans le but d'effectuer un bouclage de pertinence.

Les calculs présentés ont été réalisés sur un ordinateur portable dont les caractéristiques

sont les suivantes : Processeur Intel Pentium-M 1,1GHz (2 Mo cache), mémoire 512 Mo, disque dur 60 Go 7200 tours par minute fonctionnant sous Linux Fedora Core 4. Le niveau de performances de cet ordinateur est moyen par rapport aux technologies Intel actuelles (*Pentium 4 Hyper-threading, Itanium, Itanium2, Dual Core*). Toutefois, le choix s'est porté sur cette architecture pour deux raisons, d'une part cet ordinateur dispose de 2 Mo de cache externe, ce qui augmente les performances, et d'autre part les résultats présentés ne peuvent qu'être meilleurs avec des ordinateurs plus puissants.

7.3.1 Temps de transformation d'une image en ondelettes

Le temps de transformation en ondelettes *lifting scheme* en nombres entiers réalisé par les fonctions IPP est excellent. La figure 7.18 donne le temps de transformation de plusieurs images en ondelettes à trois niveaux de résolution avec les ondelettes de Deslauriers-Dubuc en nombres entiers d'ordre (2,2) (voir figure 3.17) données par Calderbank et al. [CDSY98].

taille image couleur (pixels)	taille disque JPEG (octets)	taille mémoire (Ko)
256x256	21877	192
512x512	71003	768
1024x1024	177192	3072
2048x2048	439714	12288
4096x4096	1121728	49152

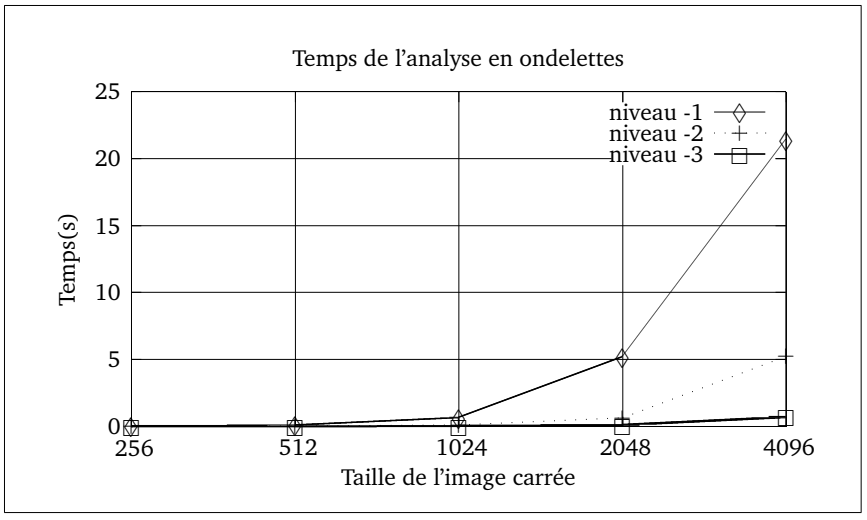


FIG. 7.18 – Temps de calcul de la transformée en ondelettes entières utilisant le *lifting scheme*.

Le temps de calcul des transformées en ondelettes à trois niveaux de résolution est assez bon jusqu'à une taille d'image de 2048 par 2048 pixels. Ensuite, pour une image 4096 par 4096, le temps augmente brutalement pour passer la barre des 20 secondes, ce qui est beaucoup même si la transformée a lieu hors ligne.

7.3.2 Temps de stockage des descripteurs dans mySQL

Chaque image est transformée en ondelettes, les attributs extraits sont stockés dans un vecteur descripteur qui est ajouté à la base de données. Le temps d'insertion d'un descripteur est faible, mais il y a un vecteur descripteur par image, donc le temps d'insertion des descripteurs dans la base croît linéairement avec le nombre d'images.

La figure 7.19 donne le temps d'écriture des signatures pour un nombre donné d'images. Ainsi, pour écrire un million d'enregistrements de descripteurs (composés de 189 attributs), il faut 22,2 secondes. Ce résultat est excellent et prouve que l'enregistrement des descripteurs d'images dans la base est très rapide et qu'il ne pénalise pas le système.

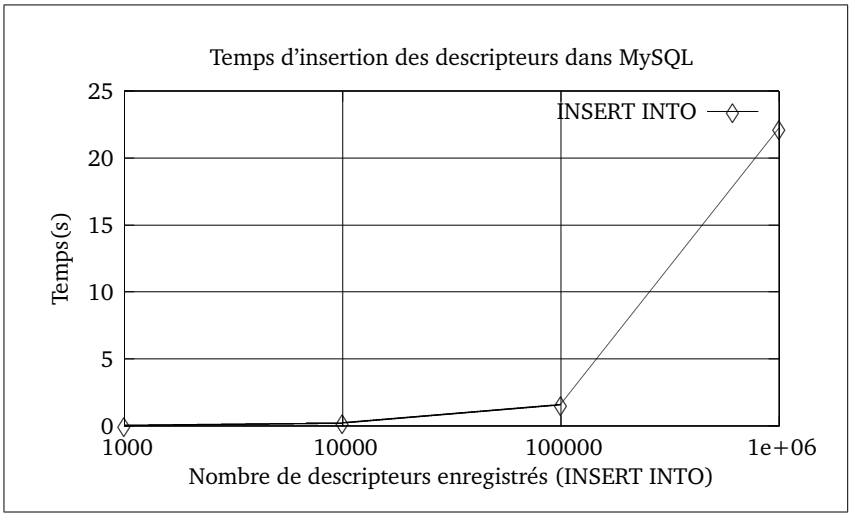


FIG. 7.19 – Temps d'écriture des descripteurs dans MySQL.

7.3.3 Temps de recherche d'un descripteur dans MySQL

MySQL est une base de données connue pour ses performances. Les fichiers de données sont automatiquement indexés par MySQL sur leur clé primaire. Chaque requête de lecture utilisant cette clé est donc très rapide car elle utilise l'index. La figure 7.20 donne le temps de lecture des vecteurs signature.

Le temps de lecture des signatures est excellent car pour lire un million de vecteurs signature, il faut compter environ une minute, ce qui correspond à la récupération en mémoire de 16 000 vecteurs (à dix attributs) par seconde.

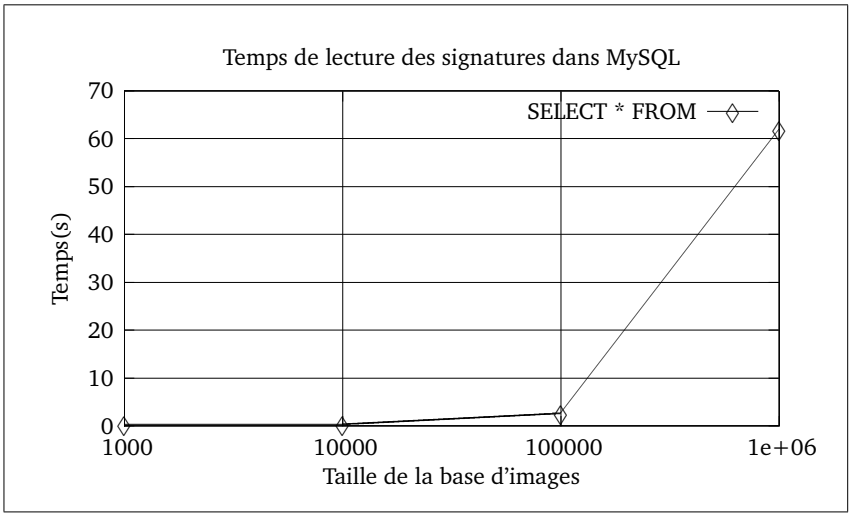


FIG. 7.20 – Temps de lecture des signatures dans MySQL.

7.3.4 Temps de classification et de construction de l'arbre

La rapidité de la classification des images en familles a été testée avec différentes tailles de base d'images. Elle dépend en effet du nombre d'images à classer et de la taille des vecteurs signature utilisés. Moins il y a de vecteurs à classer, plus on va augmenter la taille de la signature pour compenser le manque de détails.

La figure 7.21 donne les temps de construction pour des bases ayant des nombres d'images différents pour des vecteurs signatures de taille 10 (le pire des cas dans notre approche).

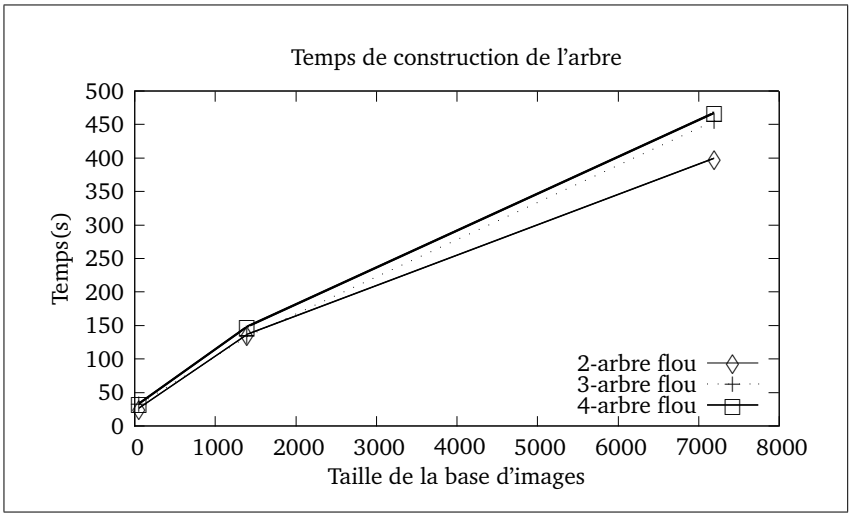


FIG. 7.21 – Temps de construction de l'arbre visuel de recherche.

Comme on le constate, il faut environ 5 minutes 10 secondes pour construire un arbre binaire pour la base Columbia (7 200 images) et 5 minutes 45 secondes environ s'il est flou (plus de calculs car virtuellement plus d'images à classer).

L'algorithme des nuées dynamiques converge très rapidement vers un état stable des centres des classes. A titre d'exemple, le classement de la base columbia (7200 images) en quatre familles ne nécessite qu'une dizaine d'itérations.

Afin de se rendre compte de la rapidité de calcul de la méthode, deux exemples sont proposés : une base réduite de 60 images issue de Trans'Tyfipal et la base Columbia (7200 images). Les tableaux ci-dessous résument les performances en terme de vitesse de l'algorithme.

Le premier tableau donne la vitesse de lecture des images, d'analyse multirésolution à trois niveaux, d'extraction de 189 attributs, de calcul de l'image reconstruite et d'écriture de ces informations dans la base MySQL. Le second tableau donne le temps de l'organisation, composé de la lecture du vecteur descripteur, de l'organisation en quatre vecteurs signature et de l'enregistrement de ces vecteurs signatures dans la base de données. Le troisième tableau donne la durée de la phase de classification pour la base indiquée avec lecture des signatures, classification floue et écriture du résultat.

- Lecture → transformation → extraction (189 attributs) → représentation → écriture

Base	Nombre d'images	Temps/image	Temps total	Taille des images
Tyfipal60	60	0,3167 s	19 s	396x424x3
Columbia	7200	0,0431 s	5 min 10 s	128x128x3

- Lecture → organisation → écriture

Base	Temps
Tyfipal60	2 s
Columbia	43 s

- Lecture → classification → A.C.P. (pour affichage seulement) → écriture

Base	Familles	Lecture	Itérations	Classification	A.C.P.	Ecriture
Tyfipal60	4	0,01 s	6	$< 10^{-3}$ s	$< 10^{-3}$ s	0,03 s
Columbia	7	0,3 s	10	0,3 s	0,17 s	0,6 s

Les performances sont très bonnes en terme de temps de calcul par image en raison de l'utilisation des librairies optimisées Intel IPP et OpenCV. Bien qu'il ne soit pas envisageable d'effectuer les transformations en ligne en raison du temps de calcul, il est possible d'effectuer le calcul de la classification directement en ligne pour des bases d'images de taille inférieure à 10 000 images. Cette classification en ligne permettrait d'effectuer des bouclages de pertinences qui influent directement sur la classification.

Une fois l'arbre créé, il faut utiliser la navigation pour trouver l'image qu'on cherche. Un protocole de validation illustrant l'efficacité de la méthode est présenté dans le chapitre suivant.

Chapitre 8

Résultats et interprétation

Ce chapitre présente les tests réalisés ainsi que les résultats de la méthode sur diverses bases d'images.

8.1 Protocole de test

8.1.1 Mesures de qualité des réponses, courbe précision/rappel

Afin de mesurer la qualité d'un système de recherche d'images par le contenu, la courbe précision/rappel est classiquement utilisée. Soit A l'ensemble des images résultat pertinentes pour une requête donnée et B l'ensemble des images résultat retournées par le système. On définit :

- La **précision** comme le rapport entre le nombre d'images pertinentes retrouvées et le nombre d'images retrouvées, $p = \frac{|A \cap B|}{|B|}$
- Le **rappel** comme le nombre d'images pertinentes retrouvées sur le nombre d'images pertinentes, $r = \frac{|A \cap B|}{|A|}$

La précision et le rappel sont compris entre 0 et 1. Ces deux mesures permettent de rendre compte de la pertinence de la réponse du système à la requête de l'utilisateur. On trace la courbe précision en fonction du rappel qui donne la pertinence des réponses du système aux requêtes.

8.1.2 mesure psycho-visuelle des résultats

Notre méthode se fonde sur la navigation psycho-visuelle des utilisateurs du système. La courbe précision/rappel ne permet pas de prendre en compte l'aspect subjectif de la recherche.

La méthodologie expérimentale a été proposée en collaboration avec M. Pierre Perruchet, directeur du laboratoire d'étude de l'apprentissage et du développement de l'université de Bourgogne. L'idée est de tester la méthode avec des utilisateurs et d'analyser les résultats à l'aide de méthodes statistiques pour en tirer des enseignements.

Le protocole de test utilisé possède deux impératifs. D'une part, il faut étudier la méthode de classification sur une base d'images réduite afin d'interpréter au mieux les résultats obtenus, le passage à la base complète est réalisé par la suite. D'autre part, afin de comparer les résultats sur un arbre strict et sur un arbre flou pour la même recherche, il faut proposer deux lots d'images à deux groupes de 10 à 15 sujets, avec une recherche stricte puis une recherche floue sur chacun des lots.

Groupe 1	Groupe 2
lot_{1f}	lot_{1s}
lot_{2s}	lot_{2f}

Le plan expérimental consiste à tester la méthode sur deux lots d'images (lot_1 et lot_2). En croisant les tests de recherche avec un arbre strict (lot_{1s} et lot_{2s}) et avec un arbre flou (lot_{1f} et lot_{2f}), pour les deux groupes d'utilisateurs, on peut comparer les résultats pour élire le meilleur arbre (flou ou strict) et ainsi prouver la validité de la méthode.

Dans notre cas, pour des raisons de facilité de mise en œuvre, un lot de six images a été proposé aux utilisateurs qui ont effectué les recherches dans des arbres stricts et flous. Ensuite, les résultats ont été comparés en ne prenant en compte qu'une partie de la recherche de l'utilisateur, c'est-à-dire qu'on a coupé les six images en deux lots de trois images.

8.2 Test de la méthode sur une base réduite

8.2.1 Test sur la base Trans'tyfipal

Afin de valider les concepts de la méthode, une base volontairement réduite d'images a été utilisée. Cette base est constituée d'un sous-ensemble représentatif des différentes images de Trans'tyfipal. on y trouve des images d'oursins, d'ammonites, d'huîtres, de coquillages, de crânes et de plantes fossiles. Cette base de soixante images est représentée en figure 8.1.

8.2.1.1 Protocole de test

Nous avons volontairement placé l'utilisateur au centre de notre méthode. Pour la tester en situation, nous avons réalisé des tests avec des utilisateurs. Notre base d'utilisateurs comprend

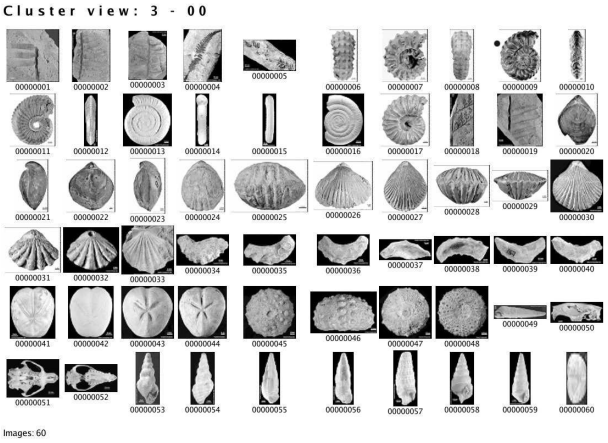


FIG. 8.1 – La base des soixante images de test.

vingt utilisateurs dont dix sont des membres du laboratoire de traitement d'images et dix autres des personnels non formés au traitement d'images. Il n'y avait volontairement aucun expert du domaine de la paléontologie.

Le test s'est déroulé de la façon suivante :

1. Les soixante images ont été classées automatiquement en deux, trois puis quatre familles (arbre de recherche binaire, ternaire et quaternaire) avec des vecteurs signature de taille 4, 6, 8 et 10. Deux classifications, l'une stricte et l'autre floue, ont été calculées.
2. Le vecteur descripteur a été organisé en hiérarchie de quatre vecteurs signature avec au premier niveau quatre attributs de forme (moments issus des images d'approximation à chaque niveau de résolution et élongation de l'objet), au second niveau six attributs de forme (moments des détails des imagettes d'ondelette), au troisième niveau, huit attributs de texture (énergie des détails diagonaux à plusieurs résolutions) et enfin au quatrième niveau dix attributs de texture (énergie des détails horizontaux, verticaux et diagonaux à plusieurs niveaux de résolution).
3. Le critère de séparabilité des classes (et donc de changement de vecteur signature) a été fixé par rapport à la distance intra-centres de classes comparée à la dispersion (écart-type) des classes autour de leur centre.
4. L'arbre de recherche construit est un arbre binaire, ternaire puis quaternaire, autrement dit, l'utilisateur dispose de deux, trois puis quatre choix à chaque étape de la navigation.
5. On a choisi six images de familles paléontologiques différentes et on a demandé à l'utilisateur de les visualiser une dizaine de secondes.
6. Les utilisateurs ont essayé de retrouver les images mémorisées dans l'arbre.

8.2.1.2 Expérimentations

La figure 8.2 présente les six images proposées pendant quelques secondes aux utilisateurs pour mémorisation avant recherche. Il y a donc six expériences : arbre binaire, ternaire et quaternaire (stricts et flous) pour six images proposées, soit trente-six arbres à parcourir. La durée du test n'a pas été imposée et la moyenne est d'environ 15 minutes.

La recherche est plus rapide pour les arbres quaternaire que pour les arbres binaires car il y a séparation plus rapide des images en familles. De même, la navigation est plus rapide pour des arbres stricts que pour des arbres flous car dans ces derniers, les images peuvent se retrouver dans plusieurs familles.

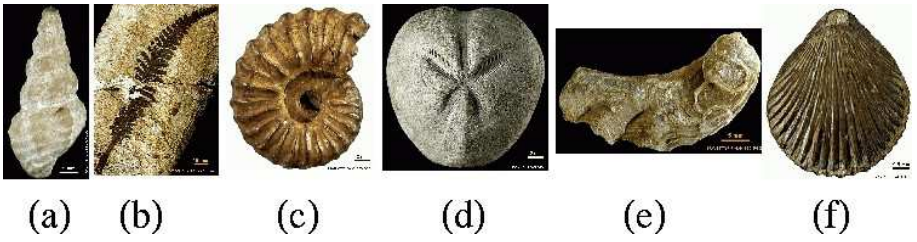


FIG. 8.2 – Les six images de test.

Les résultats des tests sont donnés pour un arbre binaire, ternaire et quaternaire, chacun étant strict puis flou. Le premier tableau résume le pourcentage d'utilisateurs qui ont retrouvé l'image dans l'arbre.

Résultats de la recherche :

	a	b	c	d	e	f
binaire strict	10%	0%	65%	60%	75%	60%
binaire flou	100%	10%	20%	0%	100%	100%
ternaire strict	100%	100%	100%	0%	100%	100%
ternaire flou	100%	40%	0%	25%	65%	100%
quaternaire strict	10%	10%	10%	90%	100%	75%
quaternaire flou	100%	100%	10%	30%	100%	90%

Le tableau ci-dessous présente les résultats croisés. C'est-à-dire le pourcentage de réussite de la recherche pour les groupes 1 et 2 et les lots 1 et 2.

	Groupe 1	Groupe 2
binaire strict	55%	50%
binaire flou	50%	60%
ternaire strict	80%	80%
ternaire flou	50%	60%
quaternaire strict	50%	55%
quaternaire flou	70%	75%

On constate que l'utilisation d'un arbre flou, quel que soit la cardinalité de l'arbre donne de meilleurs résultats que l'utilisation d'un arbre strict. Toutefois, dans certains cas, l'arbre flou donne des résultats très décevants. Ces résultats peuvent s'expliquer par le déplacement du centre de gravité des classes dû à la présence d'images dans plusieurs classes à la fois et le choix d'un paramètre de flou inadapté. Le réglage du paramètre de changement de niveau dans l'arbre est donc très important pour l'utilisation du meilleur vecteur signature.

On voit aussi que les meilleurs résultats sont obtenus par l'arbre ternaire strict. L'explication de ces bons résultat est qu'il y a dans la base de départ trois familles d'images distinctes par leur forme : les objets allongés (moules, coquillages), les objets arrondis (ammonites, oursins) et les objets à forme triangulaire (coquilles). Comme nous utilisons d'abord une signature de forme dans notre classification, les images sont mieux réparties dans trois familles que dans les arbres binaires ou quaternaires.

Les tests ont été limités à l'arbre quaternaire car des arbres de cardinalité supérieure auraient conduit à une classification à un seul niveau en raison du petit nombre d'images dans cette base de test.

Le test de changement de niveau de signature est assez simple. On calcule pour chaque classe c son centre de gravité et l'écart-type e_c autour de ce centre. On prend ensuite les deux classes dont les centres sont les plus proches $c = 1$ et $c = 2$ et on teste si la distance inter-centre est inférieure à $p * (e_1 - e_2)$, si c'est le cas, cela veut dire que les classes sont trop proches, donc on n'arrive plus à séparer ces classes avec le niveau de signature courant, donc on change de vecteur signature. Le paramètre de changement de niveau a été fixé expérimentalement à $p = 1,7$ pour la base tyfipal60 et l'ensemble des vecteurs signature considérés. Mais il faut refaire cette optimisation de p pour chaque base et chaque signature, c'est donc une contrainte importante.

8.2.2 Courbe précision/rappel

La courbe précision/rappel idéale est indiquée sur la figure 8.3. Cette courbe a été calculée pour la base tyfipal60 de soixante images avec le vecteur descripteur (189 attributs) et les quatre vecteurs signature (4, 6, 8 et 10 attributs). On voit que l'utilisation du vecteur descripteur complet est moins performante que l'utilisation des quatre vecteurs signature pris indépendamment. Pour cette base, l'utilisation de vecteurs signature de taille réduite ne freine pas les performances, bien au contraire, le temps gagné est très important sans la moindre conséquence sur la qualité.

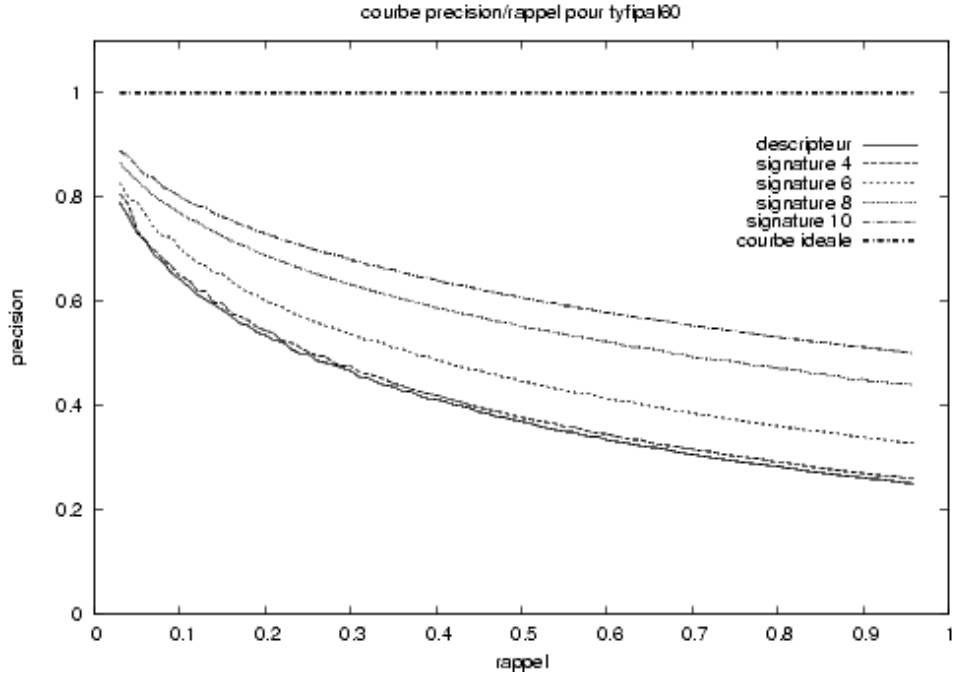


FIG. 8.3 – Courbe précision/rappel pour tyfipal60.

8.3 Test de la méthode sur une base complète

8.3.1 Test sur la base trans'tyfipal

La base trans'tyfipal complète contient 60 000 images. Celle qui était en notre possession ne comprenait que 727 images. Les résultats obtenus sur la base globale sont semblables à ceux obtenus sur la base réduite avec toutefois un nombre d'erreurs plus important pour l'utilisateur.

Le test est le même que précédemment, c'est-à-dire la recherche des six mêmes images mémorisées dans l'arbre de recherche. La figure 8.4 donne un exemple d'images modèles proposées à l'utilisateur au début de sa navigation. Le tableau ci-dessous donne le pourcentage de réussite de la recherche parmi les images de la base trans'tyfipal complète avec l'arbre strict et avec l'arbre flou.

Le tableau ci-dessous présente le pourcentage de réussite de la recherche des images deux groupes d'utilisateurs et les deux lots d'images (floues et strictes).

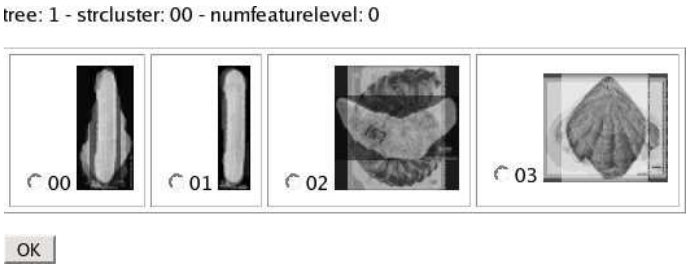


FIG. 8.4 – Exemple de choix proposés à l'utilisateur lors de sa navigation dans la base Tyfipal (4-arbre).

	Groupe 1	Groupe 2
2-arbre strict	40%	40%
2-arbre flou	45%	55%
3-arbre strict	60%	70%
3-arbre flou	60%	60%
4-arbre strict	40%	35%
4-arbre flou	50%	45%
5-arbre strict	45%	40%
5-arbre flou	45%	45%
6-arbre strict	40%	35%
6-arbre flou	65%	50%
7-arbre strict	50%	60%
7-arbre flou	60%	55%

Les résultats avec la base complète sont moins bons qu'avec la base de test notamment pour les arbres de faible cardinalité. Toutefois, l'arbre flou apporte un gain au niveau du taux de réussite dans la majorité des cas. Les arbres de plus grande cardinalité donnent de meilleurs résultats en séparant les images en familles qui correspondent mieux à la réalité de la base.

8.3.2 Test sur la base Columbia

La base Columbia est une base constituée de cent petits objets qui ont été photographiés sous 72 angles différents. Cette base possède la caractéristique d'être très utilisée en indexation d'images.

Les tests réalisés ont été les mêmes qu'avec la base tyfipal. Un arbre (généré après plusieurs tentatives pour trouver visuellement de bons attributs, voir figure 8.5) a été proposé à quelques utilisateurs pour la navigation. Un certain nombre d'images (six pour être précis) ont été données aux utilisateurs pour mémorisation et ces derniers ont tenté de retrouver les six images dans l'arbre. La figure 8.6 donne un exmple de choix proposé à l'utilisateur lors de sa navigation dans l'arbre.

La figure 8.7 montre le début de l'arbre strict construit et proposé aux utilisateurs. Cet

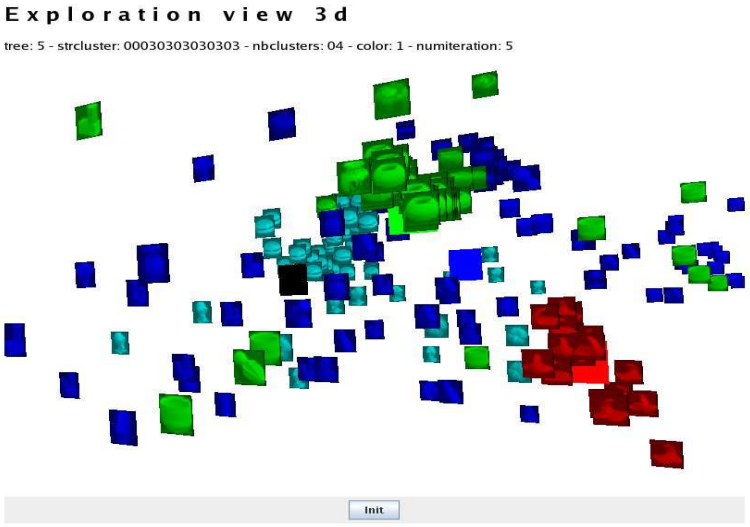


FIG. 8.5 – Exemple de vue 3D de la base Columbia.

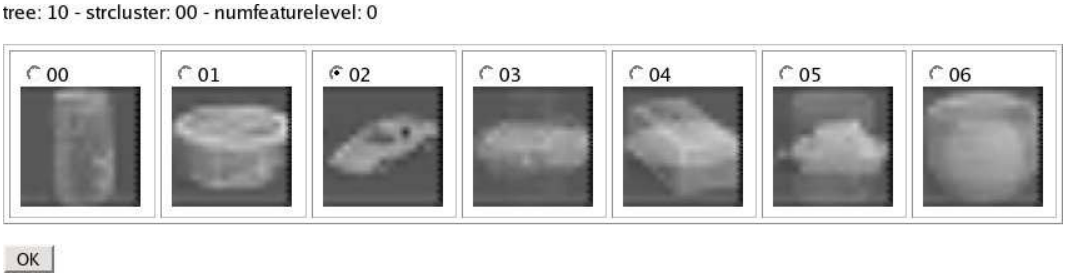


FIG. 8.6 – Exemple de choix proposés à l'utilisateur lors de sa navigation dans la base Columbia (7-arbre).

arbre a nécessité environ sept minutes pour sa construction, c'est-à-dire lecture et transformée des images en ondelettes, extraction des attributs, création des images modèles et stockage dans MySQL. La durée de création de l'arbre proprement dite a été de quatre minutes (classification stricte) et de cinq minutes (classification floue).

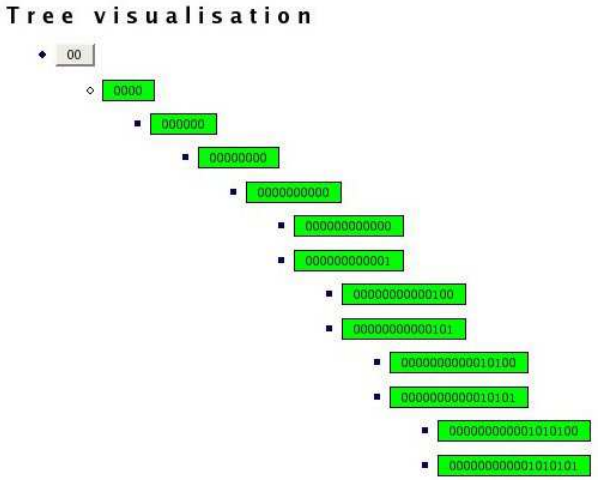


FIG. 8.7 – Début de l'arbre strict généré.

Le test de changement de niveau de signature a très bien fonctionné. La figure 8.8 illustre ce changement de niveaux avec en vert le niveau de départ (-1), en bleu le niveau suivant (-2) et en blanc le niveau maximal atteint lors de la recherche (-3).

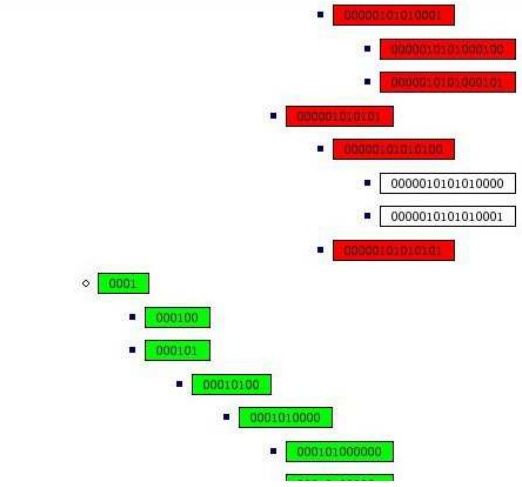


FIG. 8.8 – Les différents niveaux de signature lors de la recherche.

Les six images données aux utilisateurs pour la navigation sont représentées en figure 8.9. On y trouve un camion, un ballon, une boîte, une tasse, un tore et une voiture.

FIG. 8.9 – Les six images mémorisées par les utilisateurs.

Les résultats de la recherche par navigation sont donnés ci-dessous. Le pourcentage indique le nombre d'utilisateurs qui ont retrouvé l'image proposée dans l'arbre.

	Groupe 1	Groupe 2
2-arbre strict	10%	20%
2-arbre flou	30%	40%
3-arbre strict	10%	20%
3-arbre flou	10%	20%
4-arbre strict	50%	40%
4-arbre flou	60%	60%
5-arbre strict	40%	40%
5-arbre flou	50%	50%
6-arbre strict	50%	60%
6-arbre flou	50%	60%
7-arbre strict	60%	40%
7-arbre flou	50%	70%

Le pourcentage de réussite de la recherche est moins bon que pour la base tyfipal. Ce taux d'erreur plus important peut s'expliquer par deux raisons, d'une part les images modèles sont très floues car les images de base sont petites (128*128 pixels), d'autre part la classification donne des familles proches les unes des autres, malgré l'introduction du paramètre de flou. Néanmoins, les résultats vont en s'améliorant avec la cardinalité de l'arbre. La base étant composée de nombreux objets, la séparation en familles de plus en plus nombreuses permet de se rapprocher de la réalité de la base.

Les tableaux ci-dessous résument les tests effectués sur les deux bases. Ils donnent le pourcentage de réussite de la recherche des images mémorisées par l'utilisateur. Pour plus de lisibilité, nous avons simplifié les résultats issus des tests.

Pour la base d'images Trans'Tyfipal :

	2-arbre	3-arbre	4-arbre
Strict	40 %	**65 %**	37,5 %
Flou	**47,5 %**	60 %	**47,5 %**

Pour la base d'images Columbia :

	2-arbre	3-arbre	4-arbre	5-arbre	6-arbre	7-arbre
Strict	15 %	15 %	45 %	40 %	55 %	50 %
Flou	**35 %**	15 %	**60 %**	**50 %**	55 %	**60 %**

On constate que le pourcentage de réussite avec l'arbre flou est supérieur au pourcentage de réussite avec l'arbre strict. En règle générale, l'arbre flou apporte une amélioration dans la recherche.

Les pourcentages de réussite donnent des résultats ne dépassant pas 65%. Ce résultat est difficilement comparable aux taux de réussite d'autres méthodes qui sont aux alentours de 90 à 95% de réussite. Cette différence s'explique simplement par le fait que nous avons choisi une méthode de classification sans apprentissage et que nous utilisons des attributs très simples : moments géométriques et énergie des coefficients d'ondelettes.

En utilisant des méthodes de classification par apprentissage et d'autres attributs moins généraux, il est vraisemblable que les résultats seraient en très nette amélioration. C'est pour cette raison que ces résultats, bien que faibles par rapport à d'autres techniques sont très prometteurs.

Quatrième partie

Conclusion

Chapitre 9

Bilan et perspectives

9.1 Conclusion

Les systèmes de recherche d'images par le contenu classiques ne permettent plus de répondre aux requêtes des utilisateurs avec des temps de réponse satisfaisants. L'utilisation d'un vecteur descripteur unique de grande taille ($n > 100$) pénalise les performances, est sensible à la malédiction de la dimension et ne permet pas de prendre en compte de façon hiérarchique les différents attributs qui le composent, l'augmentation du nombre d'images numériques dans les bases d'images impose un pré-classement des images avant la requête de l'utilisateur.

Cette thèse présente un système de recherche d'images par le contenu utilisant la navigation visuelle de l'utilisateur. Un vecteur descripteur est extrait des images multirésolutions calculées par une analyse en ondelettes à trois niveaux de résolution. L'approche retenue construit un arbre de recherche flou dans lequel les images sont regroupées en familles, à l'aide de l'algorithme des nuées dynamiques, en utilisant une hiérarchie de vecteurs signature.

Les vecteurs signature de taille réduite croissante sont construits par l'expert du domaine à l'aide d'informations extraites du vecteur descripteur et sont organisés en hiérarchisant les types d'attributs (couleur, texture et forme). Ainsi, lors de la classification, les calculs de distance sont effectués sur des attributs commensurables (puisque de même type) et la taille croissante des vecteurs permet de réduire le temps de calcul. Au cours de la classification automatique, la taille du vecteur augmente avec la diminution du nombre d'images à classer.

L'utilisateur du système a la possibilité de choisir parmi plusieurs hiérarchies de signatures prédéfinies par l'expert en fonction de ce qu'il recherche. Si il cherche une voiture rouge par

exemple, la couleur puis la forme seront prépondérantes. Si par contre il cherche les oursins, la forme et la texture seront les critères importants. Le système adapte les signatures aux besoins de l'utilisateur.

La recherche utilise des attributs, issus de l'analyse en ondelettes des images, qui permettent une approche par raffinements successifs. Dans un premier temps, les attributs sont issus des images d'analyse en ondelettes avec un faible niveau de détail (imagettes grossières) puis avec le changement de vecteur signature au cours du processus de classification, les vecteurs signature contiennent des attributs issus d'images de plus en plus détaillées.

Lors de la navigation des images modèles sont proposées à l'utilisateur. Ces images modèles sont construites à partir de la moyenne des trois images les plus proches du centre de gravité de la classe à laquelle elles appartiennent. Tout comme les attributs, les images modèles utilisent les imagettes des transformées en ondelettes à trois niveaux de résolution. Elles sont donc de plus en plus détaillées au fur et à mesure de la recherche dans l'arbre.

Le système mis au point permet d'effectuer des tests d'attributs et de signatures avec visualisation en trois dimensions des résultats de la classification. Notre système permet également d'effectuer des recherches par image exemple pour une signature donnée, la signature considérée de l'image exemple est comparée aux signatures des images de la base. On obtient les images les plus proches de l'image exemple au sens de la distance choisie.

L'ajout d'images dans la base ne nécessite pas de recalculer complètement toutes les signatures sauf si les attributs de l'image ajoutée ne sont pas dans la plage de valeurs minimales et maximales utilisées pour la normalisation. Dans ce dernier cas seulement, il est nécessaire de recalculer tous les attributs de la base d'images. Sinon, l'image est classée dans la famille dont le centre de gravité est le plus proche à chaque niveau de l'arbre.

Notre contribution consiste également en la création d'un démonstrateur qui implémente la méthode mise au point. Ce démonstrateur utilise une interface Web avec le serveur Apache, le langage de scripts PHP et la base de données MySQL. Les traitements d'images sont effectués à l'aide de programmes en C qui utilisent les librairies Intel IPP et Intel OpenCV.

Le démonstrateur a été testé sur des bases d'images réelles avec des utilisateurs. Les tests montrent la faisabilité de la méthode et les résultats sur les bases de test prouvent la validité de la méthode sur des bases d'images homogènes.

Notre système est inspiré du comportement de la perception visuelle humaine en utilisant une approche de type grossier-à-fin pour l'analyse multirésolution, une classification floue par raffinements successifs (nuées dynamiques floues), une hiérarchie dans l'utilisation des signatures et dans les attributs qu'elles contiennent (couleur, texture et forme). L'utilisateur est au centre de notre approche.

9.2 Champ d'application de la méthode

Il est important de préciser le champ d'application de la méthode hiérarchique que nous proposons. La méthode développée dans cette thèse possède deux aspects.

D'une part le système permet de tester des attributs sur une base d'images grâce à l'expérience d'un expert du domaine. L'expert va être capable grâce au module de représentation 3D des classes de se promener virtuellement dans l'espace des images et de définir les meilleurs attributs pour résoudre son problème. On peut donc utiliser la méthode pour tester des attributs (multirésolution ou non).

D'autre part, le système peut être utilisé par un utilisateur non-spécialiste du domaine pour retrouver des images similaires à ce qu'il recherche selon l'une des hiérarchies proposées par l'expert. La méthode offre la possibilité de construire hors ligne des arbres de recherche flous utilisés en ligne par les utilisateurs.

L'utilisateur est guidé dans son cheminement dans l'arbre de recherche par les images modèles. La classification floue permet d'autoriser quelques erreurs dans le processus de classification. L'utilisateur reste le principal acteur de la recherche, c'est lui qui choisit les images modèles les plus ressemblantes à ce qu'il cherche.

La méthode fonctionne pour des bases d'images homogènes. Une application peut par exemple être la recherche visuelle d'un champignon (pour identification) dans une base de champignons. On peut ainsi rechercher des objets similaires à une image réelle ou bien créée par le cerveau. L'utilisateur est le seul à connaître précisémment ce qu'il cherche, il reste maître de sa recherche.

En revanche, pour des bases hétérogènes, la méthode montre ses limites en raison de la construction de l'image modèle. En effet, cette image modèle est la moyenne des trois images les plus proches du centre de gravité de la classe considérée. Dans certaines classes hétérogènes, les images sont visuellement très différentes en raison des attributs choisis pour leur classement. L'image modèle obtenue est donc très dégradée et ne permet pas de faire un choix parmi les images modèles proposées.

9.3 Perspectives

Les évolutions possibles de ce travail sont nombreuses en raison du nombre de paramètres impliqués dans les différentes étapes.

En premier lieu, il est envisageable d'étudier le comportement de la méthode avec des bases plus grandes pour voir l'influence sur le temps de calcul des descripteurs, des signatures et surtout de l'image modèle... Bien que les calculs soient très rapides, le fait d'utiliser une base d'un million d'images par exemple va sans doute poser des problèmes de temps de calcul, de

taille mémoire et de stockage et d'interrogation dans la base MySQL. En testant la méthode proposée en situation avec une base d'images de l'ordre d'un million d'images, il sera possible de déterminer si le résultat obtenu est satisfaisant.

La seconde amélioration possible intervient dans la classification pour déterminer automatiquement le paramètre de changement de niveau de signature ainsi que le découpage nécessaire ou non d'une famille avec l'inspection d'un critère qualitatif associé au critère quantitatif utilisé. Il en va de même pour la détermination d'un seuil de changement de signature, l'étude d'un critère dynamique de changement de niveau dans les signatures, en fonction des classes obtenues, peut améliorer la qualité des résultats.

Le paramètre de réglage du flou dans la phase de classification est empirique, il est indispensable d'essayer de modéliser et de maîtriser ce paramètre. S'il est mal règlé, les classes sont soit trop floues, soit trop strictes, ce qui peut conduire à une classification peu pertinente des données.

Un autre aspect du système est la méthode de classification retenue. La méthode des nuées dynamiques offre une classification rapide et pertinente. En utilisant d'autres méthodes de classification, il est possible d'améliorer la qualité de la classification. Une méthode de classification avec apprentissage doit permettre une meilleure classification des images en exploitant les connaissances acquises lors d'une phase d'entraînement du système. Des travaux récents [CDD+05] proposent une méthode de validité de familles dans les techniques de *clustering*.

Un système de bouclage de pertinence est une solution intéressante dans notre approche car le chemin parcouru dans l'arbre de recherche peut apporter des informations sur la manière de procéder de l'utilisateur. Ainsi, des techniques de pondération d'attributs peuvent conduire à de meilleurs résultats sans nuire à la rapidité du système.

En ce qui concerne la création des images modèles, l'étude expérimentale de différentes possibilités (compositions d'images proches du centre de gravité de la classe, plusieurs images prises dans la classe, ...) en collaboration avec des psychologues peut conduire à des résultats intéressants de représentation de familles d'images. L'aspect psycho-visuel est très important pour les méthodes de navigation.

Il est également possible d'ajouter des informations extérieures à notre modèle de recherche. Les méta-données de lieu de découverte des objets paléontologiques ou de leur datation (par exemple) peuvent permettre d'influencer la classification en ajoutant des informations de proximité de lieu et/ou de date entre les images. Il est aussi possible d'imaginer une phase de dialogue entre l'utilisateur et le système de recherche d'images en suivant un questionnaire préétabli par l'expert pour réduire encore le nombre d'erreurs lors de la classification (c'est une approche possible de bouclage de pertinence).

Enfin, l'utilisation de calculs rapides en nombres entiers doit permettre d'effectuer la classification des images en temps réel, en ligne, sur des bases d'images de petite taille ($n < 10000$).

Ce calcul en ligne doit permettre de prendre en compte des méthodes de bouclage de pertinence où les actions passées de l'utilisateur sont conservées et influent sur les résultats futurs.

Notre approche donne de bons résultats tant en rapidité qu'en pertinence. Elle permet de s'affranchir de la malédiction de la dimension en calculant des distances sur des espaces de faibles dimensions ($n < 10$) tout en rendant les distances plus pertinentes (on les calcule sur des attributs de nature très voisine et donc comparables en valeurs minimale et maximale).

Cinquième partie

Annexes

Annexe A

Librairies Intel

A.1 Présentation

L'association des librairies Intel IPP et OpenCV fournit un ensemble de fonctions dans le domaine du traitement du signal et de l'image. IPP est une librairie bas-niveau qui fournit des fonctions simples optimisées pour les processeurs de la famille Intel tandis qu'OpenCV dispose de fonctions haut-niveau permettant de réaliser des algorithmes d'analyse et des calculs sur des images. Les sections suivantes présentent en détails ces deux librairies.

L'emploi de ces deux librairies a nécessité de nombreuses heures de prise en main en raison d'une absence de manuel d'apprentissage. C'est pour cette raison que j'ai réalisé un livret (en anglais) donnant l'idée générale de chacune des librairies et décrivant quelques exemples simples de programmation. Ce livret est disponible sur le site officiel d'OpenCV [Lan03] ainsi que sur mon site personnel [Lan05b].

A.2 Intel Performance Primitives (IPP)

IPP [34] est une suite commerciale de fonctions bas-niveau pour le traitement du signal et des images et pour le calcul matriciel. Cette librairie est développée et maintenue par Intel Corporation. Elle offre des fonctions optimisées pour les processeurs Intel (Pentium, Pentium II, Pentium III, Pentium 4, Xeon, Itanium et Itanium2) et prend en considération les capacités câblées d'accélération matérielle du processeur comme les instructions MMX (MultiMedia eXtension), SSE (Streaming Single instruction multiple data Extensions), et SSE2. La condition

pour utiliser cette suite de librairie est de posséder un microprocesseur fabriqué par Intel Corporation. Elle se décompose en trois librairies : traitement du signal, traitement d'images et calcul matriciel.

Les fonctions de traitement du signal et de l'image de la librairie IPP sont très nombreuses, elles permettent de balayer un large choix d'opérations sur des signaux 1D et 2D. La partie calcul de la librairie permet des opérations de base sur les matrices. La plupart de ces fonctions sont câblées sur le microprocesseur et offrent de grandes performances.

- **Traitement du signal**
  - Génération de signaux,
  - Opérations arithmétiques et logiques,
  - Conversions,
  - Statistiques,
  - Filtrage et convolution,
  - FFT, DFT, DCT, ondelettes,
  - Reconnaissance de la parole,
  - Codage et décodage du son,
  - Décodage MP3 matériel...
- **Traitement d'images**
  - Génération de signaux,
  - Opérations arithmétiques et logiques,
  - Conversions d'espace couleur,
  - Seuillage et comparaison,
  - Opérations morphologiques,
  - Filtrage,
  - FFT 2D, DFT 2D, DCT 2D, ondelettes 2D,
  - Statistiques,
  - Transformations géométriques,
  - Codage et décodage JPEG et JPEG2000,
  - Codage et décodage vidéo H.263 et H.264,
  - Codage et décodage MPEG4...
- **Calcul matriciel**
  - Génération de matrices,
  - Opérations arithmétiques et logiques,
  - Produit matriciel, transposition, trace,

– Résolution de systèmes d'équations linéaires,
– Calcul des moindres carrés...

A.3 Open Computer Vision Library (OpenCV)

OpenCV [33] est une librairie libre, ouverte et gratuite développée par Intel Corporation. Elle propose des fonctions haut-niveau en vision et en traitement d'images. Cette librairie étant libre et ouverte, elle n'est pas réservée aux seuls processeurs d'Intel mais peut être installée sur diverses plateformes.

Les fonctionnalités d'OpenCV sont en perpétuel changement en raison de la licence ouverte et libre de la librairie. OpenCV fournit des fonctions haut-niveau pour la vision artificielle :

– Opérations de base sur les images,
– Analyse et traitement d'images,
– Analyse structurelle,
– Détection de mouvement et suivi d'objets,
– Reconnaissance de formes,
– Reconstruction 3D et calibration de caméra,
– Interface graphique et gestion de l'acquisition...

OpenCV implémente de nombreux algorithmes issus de publications récentes et constitue un environnement de développement idéal pour les applications en vision artificielle. De nombreuses documentations existent autour d'OpenCV ainsi qu'un forum de discussion très actif permettant d'obtenir rapidement la réponse à une question.

Annexe **B**

Degré d'appartenance

B.1 Définition

Le degré d'appartenance d'une image à une classe permet de définir une mesure d'appartenance relative d'une image à chacune des classes. Il sert à décider si une image va appartenir à la classe en cours de traitement dans l'algorithme et si elle va influencer les calculs.

Un exemple visuel est un bon moyen pour comprendre comment calculer ce degré d'appartenance. La figure B.1 donne un exemple d'un point M et de trois centres de classes (1, 2 et 3).

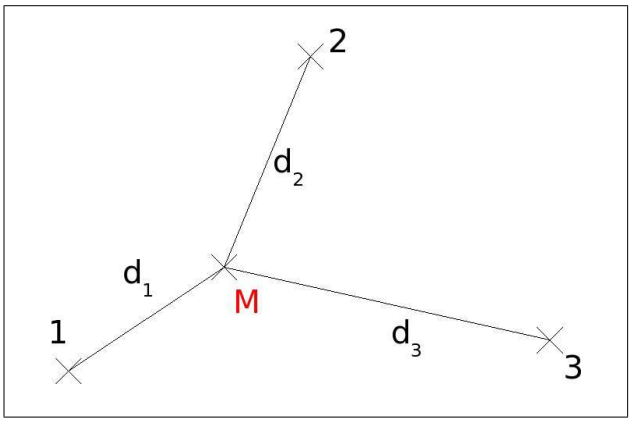


FIG. B.1 – Exemple de calcul de degré d'appartenance avec trois classes.

B.2 Calcul

Soient deg_1, deg_2, deg_3 les degrés d'appartenance de l'image M aux classes respectives 1, 2 et 3.

Soient d_1, d_2, d_3 les distances entre M et chacun des centres des classes (respectivement).

On souhaite définir le pourcentage d'appartenance d'une image aux différentes classes. Puisque le degré d'appartenance est un pourcentage, on a :

$$deg_1 + deg_2 + deg_3 = 100 \tag{B.1}$$

De plus, comme le degré d'appartenance caractérise la distance à une classe en pourcentage, on a :

$$deg_1.d_1 = deg_2.d_2 \tag{B.2}$$

$$deg_1.d_1 = deg_3.d_3 \tag{B.3}$$

Ces équations deviennent :

$$deg_2 = \frac{deg_1.d_1}{d_2} \tag{B.4}$$

$$deg_3 = \frac{deg_1.d_1}{d_3} \tag{B.5}$$

Les équations B.1, B.4 et B.5 donnent :

$$deg_1 + \frac{deg_1.d_1}{d_2} + \frac{deg_1.d_1}{d_3} = 100 \tag{B.6}$$

Soit :

$$deg_1(1 + \frac{d_1}{d_2} + \frac{d_1}{d_3}) = 100 \tag{B.7}$$

Enfin, il vient :

$$deg_1 = \frac{100}{1 + \frac{d_1}{d_2} + \frac{d_1}{d_3}} \tag{B.8}$$

$$deg_2 = \frac{deg_1.d_1}{d_2} \tag{B.9}$$

$$deg_3 = \frac{deg_1.d_1}{d_3} \tag{B.10}$$

On peut réécrire l'équation B.8 en :

$$deg_1 = \frac{100}{\frac{d_1}{d_1} + \frac{d_1}{d_2} + \frac{d_1}{d_3}} \tag{B.11}$$

D'où la formule de calcul du degré d'appartenance d'un point M à la classe i parmi k classes :

$$deg_{M \in i} = \frac{100}{\sum_{p=1}^{k} \frac{d_i}{d_p}} \tag{B.12}$$

B.3 Application numérique

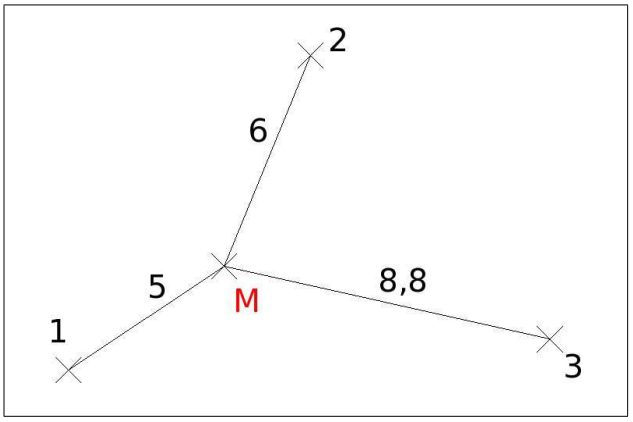


FIG. B.2 – Application numérique du calcul de degré d'appartenance avec trois classes.

En appliquant la formule B.12, on obtient directement :

$$deg_{M \in 1} = \frac{100}{1 + \frac{5}{6} + \frac{5}{8,8}} = \frac{100}{2,39} = 41,84\%$$

$$deg_{M \in 2} = \frac{100}{\frac{6}{5} + 1 + \frac{6}{8,8}} = \frac{100}{2,88} = 34,72\%$$

$$deg_{M \in 3} = \frac{100}{\frac{8,8}{5} + \frac{8,8}{6} + 1} = \frac{100}{4,22} = 23,69\%$$

Si on fixe le seuil d'appartenance à 40% dans la classification floue, l'image n'appartiendra qu'à une seule classe. Si on le fixe à 30%, l'image appartiendra bien aux classes 1 et 2 et interviendra dans les calculs comme membre de ces deux classes sans influencer le calcul sur la troisième classe.

Bibliographie

[AB91] E. H. Adelson and J. R. Bergen. *The Plenotpic Function and the Elements of Early Vision*, pages 3–20. MIT Press, Cambridge, Massachussets, USA, 1991. Computational Models of Visual Processing.

[ABP99] S. Ardizzoni, I. Bartolini, and M. Patella. Windsurf : Region-based image retrieval using wavelets. In *DEXA Workshop*, pages 167–173, 1999.

[All99] D. Alleysson. *Le traitement du signal chromatique dans la rétine : un modèle de base pour la perception humaine des couleurs*. PhD thesis, Université Joseph Fourier - Grenoble, 1999.

[Bat87] G. Battle. A block spin construction of ondelettes, Part I : Lemarié functions. *Communications on Mathematical Physics*, 110 :601, 1987.

[BCC+95] P. Bolon, J.-M. Chassery, J.-P. Cocquerez, D. Demigny, C. Graffigne, A. Montanvert, S. Philipp, R. Zédoudj, and J. Zérubia. *Analyse d'images, filtrage et segmentation*. Masson, 1995.

[Ber04] S.-A. Berrani. *Recherche approximative de plus proches voisins avec contrôle probabiliste de la précision : application à la recherche d'images par le contenu.* PhD thesis, Université de Rennes 1, février 2004.

[BFG96] J. R. Bach, C. Fuller, and A. Gupta. Virage image search engine : an open framework for image management. *Symposium on Electronic Imaging : Science and Technology*, Storage and Retrieval for Image and Video Databases IV :76–87, 1996. http ://www.virage.com.

[BH95] B. Burke-Hubbard. *Ondes et ondelettes, la saga d'un outil mathématique*. Pour la science, 1995.

[Car95] T. Carron. *Segmentation d'images couleur dans la base teinte-luminance-saturation : approche numérique et symbolique*. PhD thesis, Université de Savoie, 1995.

[CB00] J. Y. Chen and J. C. Bouman, C. A.and Dalton. Hierarchical browsing and search of large image databases. *IEEETIP : IEEE Transactions on Image Processing*, 9(3) :442–455, 2000.

[CC00] L. Chen and L. Chien. Color image segmentation using progressive wavelet transform for image retrieval. *Proceedings of ISIVC'2000 - Rabat - Morocco*, pages 245–252, 2000.

[CDD+05] F. Cao, J. Delon, A. Desolneux, P. Musé, and F. Sur. A unified framework for detecting groups and application to shape recognition. Rapport interne 1746 ISSN 1166-8687, IRISA Rennes, september 2005.

[CDF92] A. Cohen, I. Daubechies, and J. Feauveau. Bi-orthogonal bases of compactly supported wavelets. *Comm. Pure and Appl. Math.*, 45 :485–560, 1992.

[CDSY98] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Applied and Computational Harmonic Analysis (ACHA)*, 5(3) :332–369, 1998.

[Cod70] E. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6) :377–387, 1970.

[Coh99] S. Cohen. *Finding colors and shape patterns in images*. PhD thesis, Stanford University, 1999.

[CTB+99] C. Carson, M. Thomas, S. Belongie, J. Hellerstein, and J. Malik. Blobworld : A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*. Springer, 1999.

[Dau90] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5) :961–1005, 1990.

[DMM04] A. Desolneux, L. Moisan, and J.-M. Morel. *Seeing, Thinking and Knowing*, chapter Gestalt Theory and Computer Vision, pages 71–101. A. Carsetti ed., Kluwer Academic Publishers, 2004.

[Do98] M. Do. Invariant image retrieval using wavelet maxima moment. Technical report, École Polytechnique Fédérale de Lausanne, 1998.

[DQ96a] D. Declercq and A. Quinquis. *Détection et estimation des signaux*. Hermès, 1996.

[DQ96b] D. Declercq and A. Quinquis. *Le filtrage des signaux*. Hermès, 1996.

[DQ96c] D. Declercq and A. Quinquis. *Le signal aléatoire*. Hermès, 1996.

[DQ96d] D. Declercq and A. Quinquis. *Le signal déterministe*. Hermès, 1996.

[Fau03] J. Fauqueur. *Contributions pour la recherche d'images par composantes visuelles*. PhD thesis, Université de Versailles - Saint-Quentin, 2003.

[FBF+94] C. Faloutsos, R. Barber, M. Flickner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, (3) :231–262, 1994.

[FCPF01] J. Fournier, M. Cord, and S. Philipp-Foliguet. Retin : A content-based image indexing and retrieval system. *IEEE Pattern Analysis and Applications*, 4(2/3) :153–173, 2001.

[Fla93] P. Flandrin. *Temps-fréquence*. Hermès, 1993.

[FMW97] D. Forsyth, J. Malik, and R. Wilensky. La recherche d'images numériques. *Pour la science*, (238) :86–92, Août 1997.

[Fou02] J. Fournier. *Indexation d'images par le contenu et recherche interactive dans les bases généralistes*. PhD thesis, Université de Cergy-Pontoise, 2002.

[FSN+95] M. Flickner, H. Sahwney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Stelle, and P. Yanker. Query by image and video content : The qbic system. *Computer*, pages 23–32, September 1995. http ://www-qbic.almaden.ibm.com.

[Gab46] D. Gabor. Theory of communications. *Journal I.E.E.*, 3(93) :429–457, 1946.

[Gar83] G. Gardarin. *Bases de données, les systèmes et leurs langages*. Eyrolles, 1983.

[GJ97] A. Gupta and R. Jain. Visual information retrieval. *Communications of the ACM*, 40(5) :71–79, May 1997.

[GM84] A. Grossmann and J. Morlet. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.*, 15(4) :723–736, 1984.

[Gou02] A. Gouze. *Schéma lifting quinconce pour la compression d'images*. PhD thesis, Université de Nice - Sophia Antipolis, 2002.

[Gro98] P. Gros. *De l'appariement à l'indexation des couleurs*. PhD thesis, Institut National Polytechnique de Grenoble, 1998.

[GW02] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, second edition, 2002. `http://www.imageprocessingbook.com`.

[Hu62] M.K. Hu. Visual pattern recognition by moments invariants, computer methods in image analysis. *Transactions on Information Theory*, 8, 1962.

[IP95] F. Idris and S. Panchanathan. Image indexing using wavelet vector quantization. *SPIE*, 2606 :269–275, 1995.

[JFS95] C. Jacobs, A. Finkelstein, and D. Salesin. Fast multiresolution image querying. *Proceedings of SIGGRAPH95, Los Angeles, California*, August 1995.

[Koh90] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9) :1464–1480, 1990.

[Lel98] C. Leloup. *Moteurs d'indexation et de recherche - Environnement client-serveur - Internet et Intranet*. Eyrolles, 1998.

[LG01] P. Lambert and H. Grecu. Indexation par descripteurs flous : Application à la recherche d'images. *18ème colloque sur le traitement du signal et des images (GRETSI01)*, II :372–379, 2001.

[LK97] K.-C. Liang and C. Kuo. Progressive indexing, retrieval and transmission of wavelet compressed image database. *SPIE*, 3169 :190–199, 1997.

[LM86] P. Lemarié and Y. Meyer. Ondelettes et bases hilbertiennes. *Rev. Mat. Iberoamericana*, 2 :1–18, 1986.

[Lou00] E. Loupias. *Indexation d'images : aide au télé-enseignement et similarités préattentives*. PhD thesis, Institut National des Sciences Appliquées de Lyon, 2000.

[LS03] B. Le Saux. *Classification non-exclusive et personnalisation par apprentissage : Application à la navigation dans les bases d'images*. PhD thesis, Université de Versailles - Saint-Quentin, 2003.

[Mal89] S. Mallat. A theory for multiresolution signal decomposition : The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7) :674–693, July 1989.

[Mal96] S. Mallat. Wavelets for a vision. *Proceedings of the IEEE*, 84(4) :604–614, April 1996.

[Mal99] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[MAP98] M. K. Mandal, T. Aboulnasr, and S. Panchanathan. Illumination invariant image indexing using moments and wavelets. *Journal of electronic Imaging*, 7(2) :282–293, 1998.

[Mat99] S. Matusiak. *Description invariante et locale des formes planes, application à l'indexation d'une base d'images*. PhD thesis, Université de Valenciennes et du Hainaut Cambresis, 1999.

[Mey90] Y. Meyer. *Ondelettes et Opérateurs, I : Ondelettes, II : Opérateurs de Calderón-Zygmund, III :* (with R. Coifman)*, Opérateurs multilinéaires*. Hermann, Paris, 1990. English translation of first volume is published by Cambridge University Press.

[Mik02] K. Mikolajczyk. *Detection of local features invariant to affine transformations, application to matching and recognition*. PhD thesis, Institut National Polytechnique de Grenoble, 2002.

[ML90] C. Marée and G. Ledant. *SQL - Initiation, programmation et maîtrise*. Armand Colin, 1990.

[Nad00] M. Nadenau. *Integration of human color vision models into high quality image compression*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), 2000.

[NEC+01] D. Nanci, B. Espinasse, B. Cohen, H. Heckenroth, and J.-C. Asselborn. *Ingénierie des systèmes d'information : Merise*. Vuibert, 2001.

[Par03] C. Parisot. *Allocation basée modèles et transformée en ondelettes au fil de l'eau pour le codage d'images et de vidéos*. PhD thesis, Université de Nice - Sophia Antipolis, 2003.

[PDAV98] Z. Pečenović, M. Do, S. Ayer, and M. Vetterli. New methods for image retrieval. In *Proceedings of the International Congress on Imaging Science*, volume 2, pages 242–246, University of Antwerp, Belgium, September 1998.

[PDVP00] Z. Pečenović, M. Do, M. Vetterli, and P. Pu. Integrated browsing and searching of large images collections. In *Proceedings of 4th International Conference on Visual Information Systems*. Lyon, France, 2000.

[Peč98] Z. Pečenović. Finding rainbows on the internet. Technical report, École Polytechnique Fédérale de Lausanne, 1998.

[Pos87] J.-G. Postaire. *De l'image à la décision*. Dunod informatique, 1987.

[PPS96] R. W. Piccard, A. Pentland, and S. Sclaroff. Photobook : Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3) :233–254, 1996. http ://www-white.media.mit.edu/vismod/demos/photobook/.

[QMTM04] T. Quack, U. Monich, L. Thiele, and B. Manjunath. Cortina : A system for large-scale, content-based web image retrieval. In *ACM Multimedia 2004*, `http://vision.ece.ucsb.edu`, Oct 2004.

[Rei95a] H. Reinhard. *Éléments de mathématiques du signal, tome 1 - Signaux déterministes*. Dunod, 1995.

[Rei95b] H. Reinhard. *Éléments de mathématiques du signal, tome 2 - Signaux aléatoires*. Dunod, 1995.

[RHC99] O. Rui, T. Huang, and S.-F. Chang. Image retrieval : Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10 :39–62, March 1999.

[Rub99] Y. Rubner. *Perceptual metrics for image database navigation*. PhD thesis, Stanford university, 1999.

[RV91] O. Rioul and M. Vetterli. Wavelet and signal processing. *IEEE Signal Processing Magazine*, 8(4) :14–38, october 1991.

[RYM99] V. Rosenthal and Visetti Y.-M. Sens et temps de la gestalt. *Intellectica*, 28 :147–227, 1999.

[San01] S. Santini. *Exploratory image database, content-based retrieval*. Academic Press, 2001.

[SB91] M. Swain and D. Ballard. Color indexing. 7(1) :11–32, 1991.

[Sch95] T. Schneider. *Information Theory Primer*, July 1995. source : ftp ://ftp.ncifcrf.gov/pub/delila/primer.ps.

[Sch96] C. Schmid. *Appariement d'images par invariants locaux de niveaux de gris. Application à l'indexation d'une base d'objets*. PhD thesis, Institut National Polytechnique de Grenoble, 1996.

[Sha48] C. Shannon. A mathematical theory of communication. *Bell System Technology Journal*, 27 :379–423, 623–656, 1948.

[SHB99] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*. PWS Publishing, seconde edition edition, 1999.

[Sta95] H.-G. Stark. Image indexing and content base access to databases of medical images with wavelets. *SPIE*, 2569 :790–800, 1995.

[Swe95] W. Sweldens. The lifting scheme : A new philosophy in biorthogonal wavelet constructions. In A. F. Laine and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, pages 68–79. Proc. SPIE 2569, 1995.

[Swe96] W. Sweldens. The lifting scheme : A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2) :186–200, 1996.

[TB98] M. Tarr and H. Bülthoff. *Object Recognition in Man, Monkey, and Machine*. MIT/ELSEVIER, 1998.

[Tru98a] F. Truchetet. *Ondelettes pour le signal numérique*. Hermès, 1998.

[Tru98b] F. Truchetet. *Traitement linéaire du signal numérique*. Hermès, 1998.

[VBK01] R. Veltkamp, H. Burkhardt, and H. Kriegel. State-of-the-art in content-based image and video retrieval, 2001.

[VD00] M. Vissac and J.-L. Dugelay. Un panorama sur l'indexation d'images fixes. *Proceedings of ISIVC'2000 - Rabat - Morocco*, pages 72–80, 2000.

[Wan95] B. A. Wandell. *Foundations of vision*. Sinauer Associates, Inc., 1995.

[Zac00] J. Zachary. *An information theoretic approach to content based image retrieval*. PhD thesis, Louisiana state university, 2000.

Webographie

[1] *Base Columbia*. http://www.cs.columbia.edu/CAVE/research/softlib/coil-100.html.

[2] *Base de données relationnelle MySQL*. http://www.mysql.com.

[3] *Base d'images Corbis*. http://www.corbis.com.

[4] *Base d'images Corel*. http://www.corel.com.

[5] *Base d'images de Carnegie Mellon*. http://www-2.cs.cmu.edu/~cil/v-images.html.

[6] *Base d'images GoodShoot*. http://www.goodshoot.com.

[7] *Base d'images paléontologique Trans'tyfipal*. http://tyfipal.u-bourgogne.fr.

[8] *Extensible Markup Language*. http://www.w3.org/XML.

[9] *Grotte Chauvet-Pont-D'Arc*. http://www.culture.gouv.fr/culture/arcnat/chauvet/fr. Des images du passé.

[10] *Groupement de recherche ISIS*. http://www-isis.enst.fr.

[11] *Institut Lumière*. http://www.institut-lumiere.org.

[12] *Institut National de l'Audiovisuel*. http://www.ina.fr.

[13] *Joint Picture Expert Group (JPEG), page d'accueil*. http://www.jpeg.org.

[14] *KIWI, le système de recherche de l'INSA de Lyon*. http://telesun.insa-lyon.fr/kiwi.

[15] *Laboratoire Électronique, Informatique et Images*. http://www.le2i.com.

[16] *LabView*. http://www.ni.com.

[17] *Langage de scripts côté serveur*. http://www.php.net.

[18] *Logiciel de tracé de courbes*. http://www.gnuplot.info.

[19] *Maison Joseph (Nicéphore) Niépce*. http://www.niepce.com.

[20] *Mathsoft wavelets home page*. http://www.mathsoft.com/wavelets.html. Many applications of wavelets in many different domains.

[21] *Matlab*. http://www.mathworks.com.

[22] *Moteur de recherche web Google*. http://www.google.fr.

[23] *Moteur de recherche web Lycos*. http://www.lycos.fr.

[24] *Motion Picture Expert Group (MPEG), page d'accueil*. http://www.chiariglione.org/mpeg.

[25] *Projet benchathlon*. http://www.benchathlon.net.

[26] *Projet librairie numérique de l'université de Berkeley, Californie, USA*. http://elib.cs.berkeley.edu/photos/blobworld.

[27] *RETIN de l'ENSEA de Cergy-Pontoise*. http://www-etis.ensea.fr/~image.

[28] *Serveur Web Apache*. http://www.apache.org.

[29] *Statistiques sur Internet*. http://www.netcraft.com.

[30] *Système Ikona du projet iMEDIA de l'INRIA*. http://www-rocq.inria.fr/imedia/ikona.html.

[31] *Text REtrieval Conference*. http://trec.nist.gov.

[32] *World Wide Web Consortium*. http://www.w3.org.

[33] *INTEL Corporation - Open Source Computer Vision Library - reference manual*. http://www.sourceforge.net/projects/opencvlibrary, 1999-2005.

[34] *INTEL Corporation - Integrated Performance Primitives for Intel Architecture - reference manual*. http://www.intel.com/software/products/perflib, 2000-2005.

[35] Jean-Michel Jolion. *Probabilités et statistique*. http://rfv.insa-lyon.fr/~jolion.

Bibliographie personnelle

[La05] Jérôme Landré and al., *Sécurité wi-fi, une expérience d'établissement*, `http://www.cru.fr/wl`, version 0.1 ed., Mai 2005, En ligne sur le site du C.R.U.

[Lan03] Jérôme Landré, *Programming with intel ipp and intel opencv under gnu linux - a beginner's tutorial*, `http://www.sourceforge.net/opencvlibrary`, version 0.4 ed., July 2003, Présent sur le site officiel d'OpenCV dans la rubrique documentation.

[Lan04] _____, *Installation d'un serveur nat (network address translation) sous linux*, Colloque CRI-IUT 2004 - Le Creusot - France, Juin 2004.

[Lan05a] _____, *freeradius, un serveur pour la sécurisation des réseaux wi-fi*, Colloque CRI-IUT 2005 - Arles - France, Juin 2005.

[Lan05b] _____, *Site web personnel*, `http://jlandre.ifrance.com`, 2005.

[LT00] Jérôme Landré and Frédéric Truchetet, *Content-based multiresolution indexing and retrieval of paleontology images*, Proceedings of Storage and Retrieval for Media Databases, SPIE (San José - California - USA), vol. 4315, December 2000, pp. 482–489.

[LT01] _____, *Hierarchical architecture for content-based image retrieval of paleontology images*, Proceedings of Storage and Retrieval for Media Databases, SPIE (San José - California - USA), vol. 4676, December 2001, pp. 138–147.

[LT02] _____, *Approche exploratoire multirésolution basée sur le contenu d'une base d'images paléontologique*, Proceedings of Information Processing and Management of Uncertainty (IPMU) (Annecy - France), Juillet 2002, pp. 2013–2020.

[LT03a] _____, *Analyse multirésolution pour la recherche et l'indexation d'images d'une base d'images paléontologiques*, GDR ISIS - thème E, Mai 2003.

[LT03b] _____, *multiresolution hierarchical content-based image retrieval of paleontology images*, Proceedings of SPIE Photonics East 2003 (Providence - Rhode Island - USA), December 2003.

[LTMD01] Jérôme Landré, Frédéric Truchetet, Sophie Montuire, and Bruno David, *Automatic building of a visual interface for content-based multiresolution retrieval of paleontology images*, Journal of Electronic Imaging, SPIE **10** (2001), no. 4, 957–965.