APPROCHE PROBABILISTE HYBRIDE POUR LA RECHERCHE D'IMAGES PAR LE CONTENU AVEC PONDÉRATION DES CARACTÉRISTIQUES

par

Touati Hamri

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, décembre 2007

Pagination non continue
mais complet tel quel.

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-49510-0
Our file Notre référence
ISBN: 978-0-494-49510-0

Le 15 janvier 2008

le jury a accepté le mémoire de M. Touati Hamri dans sa version finale.

Membres du jury

M. Djemel Ziou
Directeur
Département d'informatique

M. Mohammed Lamine Kherfi
Membre
Département de mathématiques et d'informatique - Université du Québec à Trois-Rivières

M. Ernest Monga
Président-rapporteur
Département de mathématiques

À mon cher père et ma chère mère.

À mes frères et soeurs.

À mes neveux et nièces, mes beau frères.

À tous mes amis.

SOMMAIRE

Durant la dernière décennie, des quantités énormes de documents visuels (images et vidéos) sont produites chaque jour par les scientifiques, les journalistes, les amateurs, etc. Cette quantité a vite démontré la limite des systèmes de recherche d'images par mots clés, d'où la naissance du paradigme qu'on nomme *Système de Recherche d'Images par le Contenu*, en anglais Content-Based Image Retrieval (CBIR). Ces systèmes visent à localiser les images similaires à une requête constituée d'une ou plusieurs images, à l'aide des caractéristiques visuelles telles que la couleur, la forme et la texture. Ces caractéristiques sont dites de *bas-niveau* car elles ne reflètent pas la sémantique de l'image. En d'autres termes deux images sémantiquement différentes peuvent produire des caractéristiques bas-niveau similaires. Un des principaux défis de cette nouvelle vision des systèmes est *l'organisation de la collection d'images* pour avoir un temps de recherche acceptable. Pour faire face à ce défi, les techniques développées pour l'indexation des bases de données textuelles telles que les arbres sont massivement utilisées. Ces arbres ne sont pas adaptés aux données de grandes dimensions, comme c'est le cas des caractéristiques de bas-niveau des images. Dans ce mémoire, nous nous intéressons à ce défi. Nous introduisons une nouvelle approche probabiliste hybride pour l'organisation des collections d'images. Sur une collection d'images organisée hiérarchiquement en noeuds selon la sémantique des images, nous utilisons une approche générative pour l'estimation des mélanges de probabilités qui représentent l'apparence visuelle de chaque noeud dans la collection. Ensuite nous appliquons une approche discriminative pour l'estimation des poids des caractéristiques visuelles. L'idée dans notre travail, est de limiter la recherche seulement aux noeuds qui représentent mieux la sémantique de la requête, ce qui donne une propriété sémantique à la recherche et diminue le fossé sémantique causé par les caractéristiques de bas-niveau.

REMERCIEMENTS

Je tiens en premier lieu à exprimer ma grande reconnaissance et mes vifs remerciements à mon directeur de recherche, le professeur *Djemel Ziou* pour son aide, ses conseils, ses encouragements, sa contribution, sa disponibilité, ses critiques et sa patience.

J'aimerais également remercier mes collègues du laboratoire MOIVRE (MOdélisation en Imagerie, Vision et REseaux de neurones) *Zahir*, *Lotfi*, *Ali*, *Nizar*, *Mohand*, *Sabri* et *Ouael* pour leur soutien et agréable compagnie. Je tiens à remercier aussi mes collègues *Ahmed* et *Sidali* pour leur soutien.

Mes remerciements s'adressent à tous les membres du Département d'informatique de l'Université de Sherbrooke qui ont contribué à ma formation.

Mes remerciements vont également aux Laboratoires Universitaires Bell et au Conseil de Recherche en Sciences Naturelles et en Génie du Canada (CRSNG) pour leur soutien.

Enfin, j'adresse un remerciement particulier pour toute ma famille ainsi que tous ceux qui me sont proches pour leur soutien moral durant mon séjour au Québec.

Table des matières

Introduction

La Recherche d'Images par le Contenu ou *Content-Based Image Retrieval* (CBIR), est une technique qui localise, dans une collection, des images similaires à une requête en utilisant les caractéristiques visuelles telles que la couleur, la texture et la forme. Ces dernières années beaucoup de systèmes CBIR ont été développés [6] [15]. Ils sont motivés par les multiples inconvénients des systèmes de recherche par mots clés. Ces derniers, pour pouvoir retrouver une image, étiquettent toutes les images de la collection avec des mots clés, puis les techniques standards de recherche de texte sont appliquées pour retracer les images qui ont les mêmes étiquettes que la requête. Les systèmes de recherche d'images par mots clés souffrent de la capacité limitée des mots à décrire le contenu d'une image. En plus, l'énorme quantité d'images disponibles dans les bases de données et Internet rend le processus d'annotation très coûteux. Ces inconvénients ont donné naissance au nouveau paradigme CBIR, dans lequel la similarité entre images est déterminée par leurs contenus visuels. Plusieurs descripteurs globaux et locaux décrivant le contenu (couleur, texture et la forme) ont été utilisés dans la littérature [12] [4] [11] [13] [7] [8], un descripteur global décrit l'image complète, le local décrit une région dans l'image. Ces descripteurs de contenu sont qualifiés de bas-niveau car ils ne reflètent pas la sémantique des images. Deux images totalement différentes peuvent avoir les mêmes descripteurs. Ce fossé sémantique constitue le premier défi pour la recherche d'images par le contenu. Une solution possible consiste à introduire la sémantique grâce au retour de la pertinence [10] [7] [3] [13]. Les utilisateurs sont amenés à faire des jugements sur la pertinence par rapport à leurs besoins et les images retournées par le système. Le jugement obtenu permet de modifier des paramètres du processus de recherche. Un autre défi pour les systèmes de recherche d'images par le contenu est d'être des systèmes temps réel, en d'autres termes avoir un temps de recherche acceptable indépendamment du nombre d'images dans la collection. L'organisation de la collection d'images s'impose alors, actuellement la plupart des systèmes utilisent les structures de données [1] telles que les arbres. [9] compare différentes structures d'arbres utilisées par des CBIRs, il constate que leur performance d'indexation diminue rapidement quand la dimension des descripteurs visuels augmente. Ces structures de données ne sont pas adaptées à gérer et indexer des données à grandes dimensions ce qui est le cas de la plupart des descripteurs visuels. Les approches alternatives qui semblent prometteuses sont basées sur des modèles probabilistes [2] [5] [14]. L'utilisation de ces modèles pour l'organisation des collections d'images diminue aussi le fossé sémantique produit par les descripteurs visuels de bas-niveau. Dans ce mémoire, nous développons une approche probabiliste hybride (générative/ discriminative) pour l'organisation et la recherche d'images par le contenu. Sur une collection d'images organisée hiérarchiquement en noeuds selon la sémantique des images, nous appliquons une approche générative pour l'estimation des mélanges de probabilités qui représentent l'apparence visuelle de chaque noeud dans la collection. Ensuite nous utilisons une approche discriminative pour l'estimation des poids des caractéristiques visuelles pour maximiser la séparation entre les noeuds. L'idée de base dans notre travail est l'utilisation de notre approche pour identifier les noeuds qui représentent mieux la sémantique de la requête, après la recherche d'images est limitée à ces noeuds. Cette utilisation nous permet de diminuer le fossé sémantique causé par les caractéristiques visuelles de bas-niveau. Nous proposons un algorithme d'extraction de descripteurs visuels locaux de couleur, de texture et de forme qu'on nomme *puzzle*, qu'on compare au descripteur SIFT identifié dans la littérature parmi les meilleurs descripteurs visuels locaux proposés [11]. Dans le reste du mémoire nous détaillons notre modèle pour la recherche d'images par le contenu via un article, et une conclusion résume le travail et propose quelques perspectives.

Approche probabiliste hybride pour la recherche d'images par le contenu avec pondération des caractéristiques

Dans ce chapitre, nous exposons le travail intitulé **"A Hybrid Probabilistic Framework for Content-Based Image Retrieval with Feature Weighting"**. Dans ce travail nous développons un modèle probabiliste hybride pour l'organisation des collections d'images. Les méthodes existantes sont basées essentiellement sur les structures de données telles que les arbres. Ces structures de données ne sont pas adaptées à des données à grande dimension tels que les descripteurs visuels des images. Les modèles probabilistes semblent mieux adaptés pour l'organisation des collections d'images à cause de leur capacité à représenter efficacement les données à grande dimension. Kherfi et Ziou [5] ont proposé un modèle probabiliste hiérarchique pour la recherche d'images par le contenu. Ils ont montré la capacité d'une telle approche à gérer et organiser les collections d'images. Dans la même vision, nous proposons un nouveau modèle probabiliste hybride. La collection est décrite à travers une ontologie hiérarchique décrite par un arbre. Nous utilisons une approche générative pour la représentation des noeuds d'images par des mélanges de probabilités. Les approches génératives sont connues pour leur flexibilité vis-à-vis l'estimation et la mise à jour des paramètres des mélanges de probabilités. Le modèle génératif est consolidé par une analyse discriminative pour renforcer davantage les caractéristiques visuelles pertinentes. Dans ce travail, nous avons aussi développé notre propre algorithme pour l'extraction des descripteurs visuels de couleur, texture et forme. Notre approche a été validée sur une collection de 4300 images. Dans ce qui suit, nous détaillons le modèle probabiliste dans un rapport de recherche à soumettre à un journal international. Cet article constitue l'aboutissement de mes travaux de maîtrise en informatique sous la direction du professeur Djemel Ziou.

A Hybrid Probabilistic Framework for Content-Based Image Retrieval with Feature Weighting

Touati Hamri and Djemel Ziou

Département d'informatique, Université de Sherbrooke, Québec, Canada.

Emails:{touati.hamri, djemel.ziou}@usherbrooke.ca

Abstract

In this paper, a hybrid probabilistic framework for CBIR modeling is proposed. To build a retrieval system that runs on a collection of thousands of images, the collection is indexed. The indexing techniques currently used are based on the classical multidimensional access methods, for example, trees. The performance of such techniques decreases rapidly with the increase of data dimensionality. Since data types such as images are generally represented by high-dimension low-level features, these data structures are not suitable. Here, we develop a probabilistic framework for image collections organization, which is better suited to high-dimension data, and brings a semantic property to the retrieval process, narrowing the gap between human perception and the low-level features. To make our framework more flexible than existing ones, we use a generative approach to estimate the model parameters. We develop a discriminative approach for feature weighting to improve the clustering performance of the generative model. Furthermore, we propose an algorithm to extract local color, shape, and texture features. Our local shape feature performs better than the well-known SIFT in our model.

Keywords: Content-Based Image Retrieval, collection organization, feature weighting, generative model, discriminative model.

1 Introduction

In the last ten years a number of Content-Based Image Retrieval (CBIR) systems have been proposed[18][36]. These systems are motivated by several drawbacks of keyword-based image retrieval systems, including the limited capacity of keywords to describe image content and the rapid expansion of multimedia technology which increases the number of images in databases and Internet, making the annotation process very expensive. CBIR systems retrieve relevant images in a database using visual content of the images, colors, textures and shapes. In general, such systems differ from each other in five ways: what visual *features* they use, how they evaluate the *similarity* of the images, how they *index* their collection to increase the efficiency of the retrieval process, how they express their *query*, and the manner in which they employ user *feedback* to improve retrieval.

For the features, several describing color, texture and shape have been used in literature. Color features include the color histogram [9] [8], the color coherence vector [30], the color co-occurrence matrix [37] [7] [21] [15] and color moments [9] [19] [33] [8] [24]. Under texture features we find values derived from the gray-level co-occurrence matrix[19], the Tamura feature[37], wavelet coefficients[9] [33], Gabor filter-based features [8] and local binary patterns [34]. More details on texture can be found in [26]. Some shape features are the normalized inertia [9], the directional fragment histogram [38], Zernike moments [8], the histogram of edge direction [33] and the edge map [2] [40] [20]. A feature descriptor can be dense or discrete. A dense feature is computed on all pixels, while discrete ones are computed on a subset of pixels. To extract discrete features two techniques are commonly used. The first applies a segmentation algorithm to divide the image into homogeneous regions, after which a feature descriptor is extracted from each region [17] [9]. The second technique is to detect salient points (called also interest points or regions of interest, ROI), after which we calculate a feature vector around each of them. Several salient point detectors and descriptors have been proposed in the literature, including the Scale Invariant Feature Transform (SIFT) [23] and the Harris-Laplace regions [27]. For more details, see [28], where Mikolajczyk and Schmid compare several salient point detectors and descriptors.

Once image features are extracted, another problem is how we can measure the similarity between them. Rubner et al [32] give a good summary of the various similarity measurements. They classify them into heuristic distances like *the Minkowski-form distance* L_p, non-parametric test statistics such as *the* X^2*-statistic*, information-theory divergences such as *the Kullback-Leibler divergence*, and ground distances such as *the quadratic form*. Note that when we deal with ROI we need a matching strategy in addition to the similarity measurement, because of the fact that one image region can match several regions in another image. The study described in [28] compares three strategies: *threshold-based matching*, *nearest-neighbor matching*, and *nearest-neighbor distance ratio matching*. Generally, authors use similarity measurements suited to their features and query model. For example, in [17], a probabilistic metric based on likelihood estimation is used.

CBIR systems use low-level features to represent images. However, these features don't necessary represent the human perception of these images. To overcome this gap between low-level features and image semantics, authors have introduced the user need as a dimension in the system. This intervention is known in the literature as *Relevance Feedback* (RF): the user is asked to give a judgment on the retrieved images by selecting those which are relevant and irrelevant to his query. The system will use this judgment to improve retrieval. Two main approaches exist in the literature, *the optimal query* technique and *feature weighting*. In the optimal query technique (see [25]) the system updates the query according to user feedback by finding the query value that minimizes the distance from the user-relevant images, and maximizes that from the irrelevant images. In feature weighting (see [19] [10] [33]) systems increase weights for the features that discriminate between relevant and irrelevant images, and reduce them for non-discriminating features. When measuring the similarity between the query and images, the features with high weights thus make a greater contribution.

Another important thing in CBIR systems is the indexing of the image database. To build a retrieval system that runs on a collection of thousands of images, the collection must be indexed. Compared to the body of literature on retrieval approaches, little work has been done on visual content-based indexing. The currently used indexing practice seems to be based on the well-known multidimensional access methods (R-tree,

R^+-tree,R^*-tree, KD2B-tree). A good summary of these methods can be found in [1]. In a comparative study of the existing structures used in CBIR, [22] Ling et al report that their performance decreases rapidly as the data dimension increases. So these data structures are not useful when dealing with high-dimension data, which is the case for data such as images and video. Alternative approaches that seem promising are based on probabilistic and neural network models such as those reported in [3] [16] [35]. Such frameworks are commonly used in the object recognition field for example [14] [39] [12]. These frameworks have an important property: they incorporate a semantic meaning for clusters. Using them to index databases for CBIR will help to improve retrieval performance by eliminating semantically irrelevant clusters, giving a semantic property to the retrieval process and bringing it closer to human perception. This kind of approach is in its infancy and more effort is needed in this direction.

This work is related to the research done by Kherfi et al [16]. They develop a probabilistic approach for modeling image collections, applied to a hierarchical collection of images. Their approach offers several interesting characteristics. It is applicable for different purposes: indexing, retrieval, browsing and summarizing. It combines image content with keywords, narrowing the semantic gap; it is hierarchical and thus suitable for indexing. The advantages of the hierarchical model have been demonstrated by several studies like [3] and [35]. However, the maximization of their model parameters is done for the whole collection at the same time, so that the optimal parameters for a given class in the collection depend on the parameters of all classes at the same level. This fact makes updating the collection (adding/removing classes or images) very heavy, and requires the model to be refit at each significant update. By significant, we mean an update that requires a model parameter update, for example adding a class. To better understand this problem, let us recall the concept of a *generative/discriminative* probabilistic framework. In general, we can divide the probabilistic models existing in the literature into two approaches: generative and discriminative. In generative models like those described in [11] [5], the model for each class is learned separately by using only its data set. With this paradigm, the aim is to be able to reproduce the class. Using probabilistic terminology, let's call the data o and the class m. With generative models we aim to estimate the probability density function $p(o \mid \theta_m)$ where θ_m are the PDF parameters that represent

m. In the discriminative approach [35][13][39], the borders between classes are learned using all classes in the data set at the same time. So $p(\theta_m \mid o)$ is learned instead of $p(o \mid \theta_m)$ in the generative approach. In the literature, discriminative approaches have proven their superiority to generative ones in classification performance [29]. However unlike generative approaches, they do not handle the problem of missing data and adding or removing a class means performing a new learning process for all the data, which is very time-consuming. Generative and discriminative models can be combined, as in [14], to produce what we call *a hybrid model.* Such models aim to combine the flexibility of the generative and the performance of the discriminative approach. In our work we aim to produce a probabilistic approach that is flexible and discriminative. Using the same paradigm proposed by Kherfi et al., we develop a hybrid model. First we use a generative approach to represent our image classes with different features, producing a PDF for each class and feature separately; then we apply a discriminative approach, to estimate the feature weights to be used to combine features that maximize the separation between classes. We have applied this hybrid approach to a hierarchical collection, where each class in the hierarchy will have its own feature weights that maximize the discrimination between it and its neighbors.

Section 2 presents our approach. We explain the technique used to model the collection and the feature weighting schema, followed by a detailed description of our clustering and retrieval process. Section 3 presents our feature extraction algorithm, that we call *puzzle-feature*, which gives us local color, shape and texture features. We go onto demonstrate that in our model, it performs better then the well-known SIFT. Section 4 presents the details and results of our experimental tests. Section 5 gives our conclusion and directions for future work.

2 Collection modeling

A collection is a set of visual documents, an ontology, and a global description of the collection. A visual document is the representation of a concept formed by one or more objects, the relationship between them, and any kind of associated metadata. It can be an image or regions of interest that are structured on a local ontology, and represented by radiometric vectors. Each radiometric feature is represented by local and/or global vectors. For example, color may be represented by a global histogram of color and/or by color vectors representing key regions of the visual document. A prior; knowledge describing the semantic content (such as object name) is associated with each node in the collection, for example the "animals" node or airplanes node.

For our purposes, an ontology is a hierarchical (tree) data structure containing all the relevant objects related to a specific domain, their relationships and the rules within that domain. The choice of a hierarchical structure will help us in the feature weighting process, where the idea is to define, for each level in the hierarchy, its proper feature weights. For example for cars and apples, shape features will be helpful to separate them while in the case of red apples and green apples, color features will be more helpful than shape. So a hierarchical structure has a very interesting property that we will exploit in our model. Figure 1 gives an example of a hierarchical structure, where the *root* is the node "animal". In "animal", we find "air", "land", and "water" animals. These are *intermediate* nodes, and each of them has its daughters which are the *leaves*. In such a collection, the images belonging to an intermediate or root node are all its daughters images.

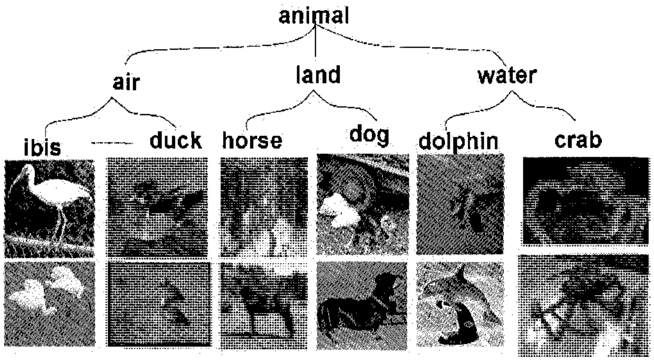


Figure 1: Hierarchical data structure

2.1 The model

Modeling the collection makes it possible to understand the data, extract missed information, and increase the accuracy of the management system. It makes the management straightforward in the sense that any management operation is directly deduced from the way the collection is modeled. In our model, the management of visual documents is based on their features. The definition and the computation of these features are based on the radiometrical content of the visual document, such as color, texture, shape, and regions of interest. Our collection is organized into several node levels, where each node is an abstraction of an object or set of objects and their relationship, providing semantic information on visual documents. Due to the uncertainty in visual data, the main idea underlying our collection modeling approach is to estimate, for each node in the collection, its probability density function (PDF) representing the appearance models of its associated visual documents.

Mathematically, consider $m = (l, c)$ a collection node, where l indexes the levels and c the node at level l. The appearance model of m is a PDF mixture of o defined by:

$$p(o \mid \theta_m) = \sum_{i=1}^{K_{lc}} p_{mi} p(o \mid \theta_{mi}) \tag{1}$$

$p(o \mid \theta_m)$ is a mixture of K_{lc} clusters with a parameter vector θ_m that indicates how the visual document o can be generated by the node m, where p_{mi} is a mixing parameter and $p(o \mid \theta_{mi})$ a PDF with parameters θ_{mi}. Modeling the collection requires the estimation of all parameters maximizing the likelihood for each node. The likelihood formulation is given as follows. Each visual document is represented by N_r regions of interest, each of which has N_f specific features. Such region may be homogenous parts of an object, inhomogeneous regions (e.g., regions containing a point of interest). It should be noted that one of these regions is the visual document itself, allowing a global representation. For example, let us assume that the visual document represents a face. The face is described by regions of interest obtained by a detector such as SIFT or Harris-Laplace. A vector of features provided by the detector is associated with each region, allowing the face to be described by a set of vectors. Furthermore, the node is formed by N_o visual documents o. The global generative likelihood for a given node m is expressed by:

$$L_m = \prod_{f=1}^{N_f} \prod_{n=1}^{N_o} \prod_{r=1}^{N_r} (\sum_{i=1}^{K_{mf}} p_{mfi} p(a_{o_n rf} \mid \theta_{mfi})) \tag{2}$$

where $a_{o_n rf}$ is the vector feature f describing the region r in the document o_n. Kherfi et al. [16] use the same idea, but they define their likelihood for a given level l, which leads to a costly maximization schema. Our likelihood is defined for each node separately, which makes it more flexible when processing an update. We maximize the likelihood of each PDF mixture separately using an EM algorithm. The accuracy of the collection modeling depends on the choice of PDFs to be used and the optimization algorithm. The PDFs must fulfill the following requirements: 1) accurately reproducing the shape of the data space; 2) allowing statistical interpretation of the data; and 3) allowing modeling in high-dimensional space. To fulfill the above requirements, we chose a Dirichlet of the second kind. This PDF has nice properties such as a flexible shape (asymmetric, symmetric) and allowing the modeling of high-dimensional data[4]. If a random vector $\overrightarrow{X} = (X_1, X_2, ..., X_d)$ follows a generalized Dirichlet distribution, the PDF is given by:

$$p(X_1, ..., X_d) = \prod_{i=1}^{d} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} X_i^{\alpha_i - 1} (1 - \sum_{j=1}^{i} X_j)^{\gamma_i} \tag{3}$$

where $\sum_{i=1}^{d} X_i < 1$, $0 < X_i < 1$ for $i = 1, ..., d$, $\gamma_i = \beta_i - \alpha_{i+1} - \beta_{i+1}$ for $i = 1, ..., d-1$, and $\gamma_d = \beta_d - 1$.

The mixture parameter estimation is done by the EM algorithm proposed in [4], where a d-dimension generalized Dirichlet mixture parameter estimation problem is reduced to the estimation of d Beta mixtures. Details on this EM are given in Appendix B. Note that in the definition of the generalized Dirichlet distribution, we have two conditions on the data: $\sum_{i=1}^{d} X_i < 1$ and $0 < X_i < 1$ for $i = 1, ..., d$. Since we can't be sure that the data we are using satisfy these conditions, we must apply a transformation to the data. Details on this transformation are given in Appendix C.

2.2 Feature weighting

For a given node, the relevance of different features is not the same. For example, the texture feature will be relevant in the case of textured images and irrelevant in non-textured images. Consequently, features cannot contribute to the retrieval process in the same manner and a weighting process must be applied. The model in Equation 1 is well known in the literature as the least semantic probabilistic model. We will make significant improvements to this model by increasing its data-clustering capabilities by using feature weighting.

The definition of $p(o \mid \theta_m)$ requires particular attention. Let us consider two features. The first feature is shared by all nodes and $p(o \mid \theta_m)$ for all l is high. For the second, $p(o \mid \theta_m)$ is high for node m and low for the other nodes. The second feature allows us to discriminate between nodes and its contribution in the clustering process is desirable, while that of the first feature is not. More generally, a feature is relevant if $p(o \mid \theta_m)$ is high and this is not the case for nodes at the same level of abstraction. Otherwise it is irrelevant. The introduction of the relevance of features reduces the confusion between nodes, thus leading to more accurate clustering. We define the relevance of a feature f for a node m as:

$$\rho_{mf} = \prod_{n=1}^{N_o} \prod_{r=1}^{N_r} \frac{p(a_{o_n rf} \mid \theta_m)}{\sum_{k \epsilon \Omega} p(a_{o_n rf} \mid \theta_k)} \tag{4}$$

where Ω is a list of nodes at the same abstraction level.

At a given abstraction level, the weighting process consists in defining feature weights. These weights enhance $\frac{p(a_{o_n rf} \mid \theta_m)}{\sum_{k \epsilon \Omega} p(a_{o_n rf} \mid \theta_k)}$ in the clustering processes for relevant features and attenuate it for irrelevant features. Consequently, the weights for a relevant feature should be high, meaning that its participation will be greater than other features, and low for an irrelevant feature, meaning that its participation will be less than that of other features. Once all node parameters are estimated for all features, these weights (σ) should maximize the following discriminative likelihood which is defined for a given list of nodes Ω:

$$L_\Omega = \prod_{m=1}^{N_\Omega} \prod_{n=1}^{N_o} \prod_{r=1}^{N_r} \prod_{f=1}^{N_f} \left(\frac{p(a_{o_n rf} \mid \theta_m)}{\sum_{k \epsilon \Omega} p(a_{o_n rf} \mid \theta_k)}\right)^{1/\sigma_{mf}} \tag{5}$$

where N_Ω is the number of nodes in Ω, and σ_{mf} the weight of feature f.

Since $\frac{p(a_{o_nrf}|\theta_m)}{\sum_{k\epsilon\Omega} p(a_{o_nrf}|\theta_k)} \in [0,1]$, then when σ_{mf} is close to 0, $1/\sigma_{mf}$ is high and the feature relevance is attenuated, while when σ_{mf} is high, $1/\sigma_{mf}$ is low and the relevance of the feature is increased. For convenience, we maximize the log of equation (5). For two given features, it is sufficient to compare the relative attenuation of the irrelevance. That is, if one of these two features is more relevant then its weight should be greater. Consequently σ_{mf} can be chosen between zero and one. We need to estimate σ_{mf} by maximizing equation (5) under the constraint $\sum_{f=1}^{N_f} \sigma_{mf} = 1$. We obtain (see Appendix A):

$$\sigma_{mf} = \frac{\sqrt{- \sum_{n=1}^{N_o} \sum_{r=1}^{N_r} log(\frac{p(a_{o_nrf}|\theta_m)}{\sum_{k\epsilon\Omega} p(a_{o_nrf}|\theta_k)})}}{\sum_{j=1}^{N_f} \sqrt{- \sum_{n=1}^{N_o} \sum_{r=1}^{N_r} log(\frac{p(a_{o_nrj}|\theta_m)}{\sum_{k\epsilon\Omega} p(a_{o_nrj}|\theta_k)})}} \qquad (6)$$

Based on the above analysis, the schema in Figure 2 summarizes our model.

2.3 Retrieval

Let us recall that retrieval is a ranking of the visual documents available in the collection according to the user query. Since in our collection, each node is an abstraction of a given semantic meaning (object name), the idea behind our model is to identify the k best nodes in the collection that most closely represent the semantic meaning of the user query, after which, the retrieval is limited to these k best nodes. We have chosen to select k nodes, because the user query can match several nodes if it contains several objects. k can be set to a given value or determined by thresholding the score. In our experiments, we set k to a given value. So our retrieval process involves two steps: 1. identifying the k best nodes in the collection and 2. performing a retrieval from these k nodes.

Note that our collection is organized hierarchically, so the node ranking process will also be hierarchical. Starting with the root's daughters, we rank the nodes and keep the k best ones. Then for each chosen node, we rank its daughters and again keep the k best ones. The process is repeated until all the chosen nodes are leaves. Finally, we rank the leaves retained. The images belonging to the k best leaves will participate in the image ranking to determine the best images. We will now detail the node ranking process for a given level of abstraction, which is used hierarchically to identify the k best leaves, as explain before. We also give the image ranking process used to identify the images most

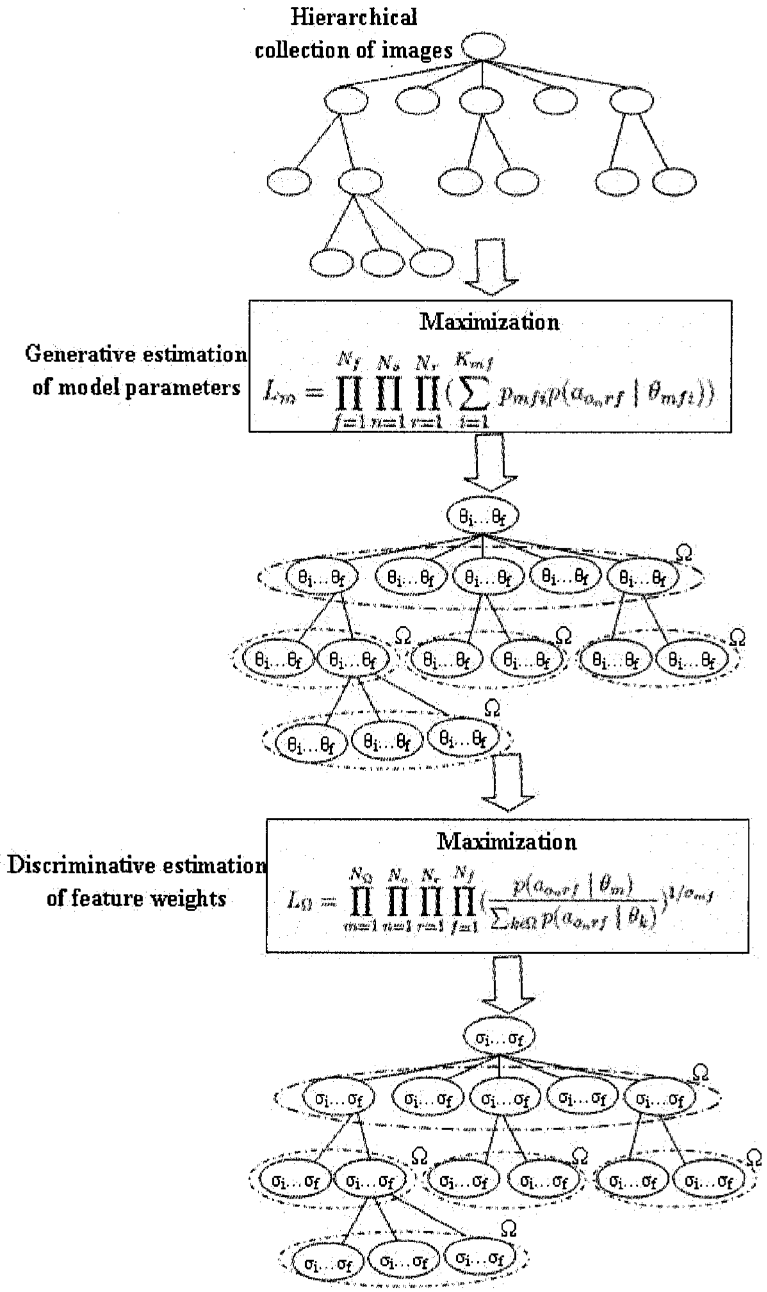

Hierarchical collection of images
Generative estimation of model parameters
Maximization
$L_m = \prod_{f=1}^{N_f} \prod_{n=1}^{N_o} \prod_{r=1}^{N_r} (\sum_{i=1}^{K_{mf}} p_{mfi} p(a_{o_n rf} \mid \theta_{mfi}))$
$\theta_i \ldots \theta_f$
Ω
Discriminative estimation of feature weights
Maximization
$L_\Omega = \prod_{m=1}^{N_\Omega} \prod_{n=1}^{N_o} \prod_{r=1}^{N_r} \prod_{f=1}^{N_f} (\frac{p(a_{o_n rf} \mid \theta_m)}{\sum_{k \in \Omega} p(a_{o_n rf} \mid \theta_k)})^{1/\sigma_{mf}}$
$\sigma_i \ldots \sigma_f$

Figure 2: Our model

similar to the query.

Consider that we have a query $Q = q_1 \wedge ... \wedge q_n$ where $\wedge$ is the "and" operator and q_i is an image. This query form which is the same used in [16] and [3], enables us to combine positive images. Note that query forms combining negative and positive images have been used in the literature: for more details see [19]. For a given level of abstraction l, the node ranking can be defined as a ranking of the probabilities that each node at l generates the query Q. These probabilities are calculated using a Bayesian rule where we combine all features using their weights. The node ranking is done by the equation:

$$argmax_{m \in l}\{\prod_{f=1}^{N_f} p(\theta_{mf} \mid Q)^{1/\sigma_{mf}}\} \tag{7}$$

where σ_{mf} is the weight of feature f calculated as described in section 2.2 and $p(\theta_{mf} \mid Q)$ indicates how probable the query is to be generated by the node m using the feature f. According to Bayes' rule, we have:

$$p(\theta_{mf} \mid Q) = \frac{p(Q \mid \theta_{mf})}{\sum_{k \in l} p(Q \mid \theta_{kf})}$$

Assuming that all q_i and a_{q_ij} are independent, we have:

$$p(Q \mid \theta_{mf}) = \prod_{i=1}^{n} \prod_{j=1}^{N_{ir}} p(a_{q_ijf} \mid \theta_{mf})$$

where n is the number of images in query Q, N_{ir} the number of regions of interest in image q_i, a_{q_ijf} is the descriptor of feature f in region j in q_i and $p(a_{q_ijf} \mid \theta_{mf})$ is the score given by the PDF representing the node m using feature f.

For image ranking, let ψ be the list of images o belonging to the k best nodes chosen according to Equation 7. When performing retrieval in ψ, the image ranking will be done using all features by combining two similarities: semantic similarity and low-level similarity. Semantic similarity indicates the semantic resemblance between query Q and the node of image o, which is the probability that the node generates the query. The low-level similarity indicates the resemblance between the query and the image o using low-level features (color, shape and texture). The following equation gives the image ranking:

$$argmax_{o \in \psi}\{\prod_{f=1}^{N_f} (p(\theta_{mf} \mid Q)^{1/\sigma_{mf}} p_f(Q \mid o))\} \tag{8}$$

where $p(\theta_{mf} \mid Q)$ (the semantic similarity) is defined as before and $p_f(Q \mid o)$ (the low-level similarity) is the similarity score between image o and query Q according to feature f. Since, in the proposed model, an image is characterized by regions of interest as seen in section 2.1, the low-level similarity measurement should be adapted to these regions of interest. $p_f(Q \mid o)$ can be used both for similarity and matching. For matching, Mikolajczyk et al [28] perform a good evaluation of different regions of interest and compare three strategies of matching. We chose to use the *Nearest Neighbor Matching* (NNM) strategy, where two regions of interest A and B are matched if the descriptor of B is the nearest one to the descriptor of A, and the distance between them is below a given threshold. Let $\phi_{q_i,o} = ((a_{q_i1f}, a_{o1f}), ..., (a_{q_ib_if}, a_{ob_if}))$ be the list of the b_i matched regions between images q_i and o according to the NNM strategy using feature f. We define $p_f(Q \mid o)$ by:

$$p_f(Q \mid o) = \prod_{i=1}^{n} p_f(q_i \mid o) = \prod_{i=1}^{n} \frac{1}{N_{ir}}(b_i - \sum_{j=1}^{b_i} d(a_{q_ijf}, a_{ojf}))$$

where $d(a_{q_ijf}, a_{ojf})$ is the distance between matched regions a_{q_ijf} and a_{ojf} using feature f. As a measure of similarity, we can use the Kullback-Leibler divergence [32] defined by:

$$p(Q \mid o_f) = 1/KL(p(Q \mid \theta_{Qf}), p(o_f \mid \theta_{of}))$$

where θ_{of} and θ_{Qf} are the PDF mixture parameters representing all regions of interest using feature f in the image o and the query Q respectively. We use the same EM algorithm [4] used in model estimation to estimate θ_{of} and θ_{Qf}. KL is a Monte Carlo approximation of the Kullback-Leibler divergence given by:

$$KL(p(x \mid \theta_i), p(x \mid \theta_j)) \approx \frac{1}{s} \sum_{m=1}^{s} \log \frac{p(x_m \mid \theta_i)}{p(x_m \mid \theta_j)}$$

where $x_1, ..., x_s$ is a sample drawn according to $p(x \mid \theta_i)$. Finally, note that in our experiments we use matching.

3 Visual features

In our CBIR system, we use several features describing color, shape and texture. We use the well-known feature SIFT, and we investigate a new feature that we call puzzle-feature, which can be applied to obtain local color, texture and shape. We will now details.

3.1 Puzzle-feature: a new feature

Our aim is to build a feature which will enable us to find even a part of an image. The idea is inspired by the familiar notion of "jigsaw puzzle" (see Figure 3 for an example). We build a picture puzzle by carving a given picture into small fragments, nearly the same shape, and then mixing them. People with inductive reasoning aptitude can resolve the puzzle (put all the pieces back together). The interesting thing about this puzzle, is that all of the pieces we can reassemble the full picture but also other pictures (see Figure 4).

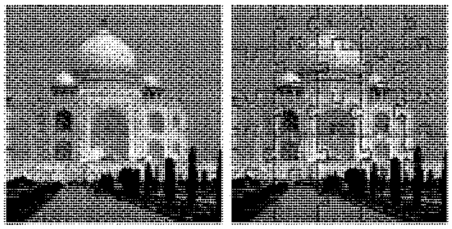


Figure 3: Puzzle example

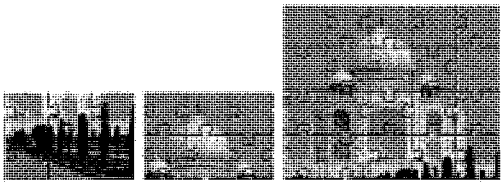


Figure 4: Picture built from the same puzzle

The content of the puzzle pieces determines the nature of the features. If one uses pieces which describe the color, it will be a color-puzzle; shape content will make it shape-puzzle; texture content, a texture-puzzle. We chose a square pieces shape because it is simple to handle, as shown in Figure 5.

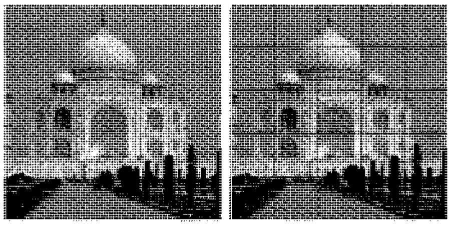


Figure 5: Square-based puzzle

The main question is what width of square to use. The answer to this question is the solution for the scale invariance feature problem. From an image we produce a series of puzzles, starting with a given width and doubling it until obtaining minimum 9 pieces for the puzzle as shown in Figure 6. Using this puzzle series, the puzzle feature will be composed by all the pieces of different width which preserve the image at different resolutions.

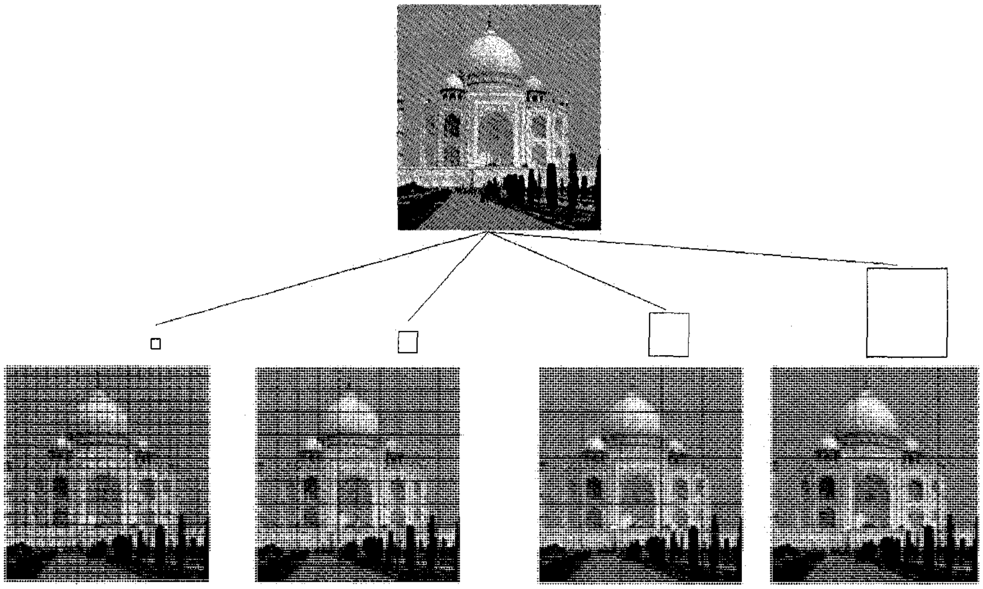


Figure 6: Square-based puzzle series with different square widths

Now, we need to maintain the neighbor relation between puzzle pieces. This relation preserves spatial information and decreases the interference between puzzle pieces. For example, pieces 1 and 2 (see Figure 7) resemble each other in color. We propose to use the following neighbor relation. Let $X_p : (x_{p1}, x_{p2}, ..., x_{pn})$ be an n-dimension vector which defines the content (color, shape, or texture) of the puzzle piece p. Let $X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8$ be the vectors corresponding to p's neighbors. We define the difference vector $D_{p,i}$ between p and the neighbor X_i by:

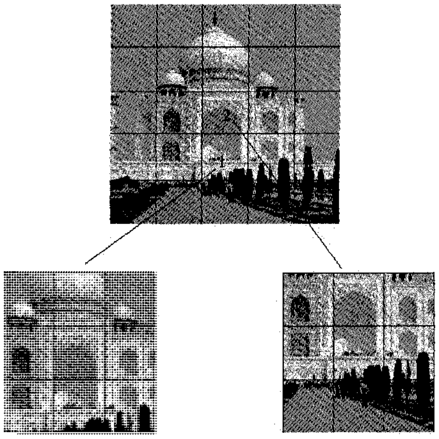


Figure 7: Interference between pieces

$$D_{p,i} = (d_{pi1}, d_{pi2}, ..., d_{pin})$$

$$= (x_{p0} - x_{i0}, x_{p1} - x_{i1}, ..., x_{pn} - x_{in})$$

The neighboring relation between piece p and its neighbors is captured by the mean and variance of all $D_{p,1}, D_{p,2}, ..., D_{p,n}$ dimensions:

$$Mean(D_p) = (M_{p1}, M_{p2}, ..., M_{pn}) = (\frac{1}{8}\sum_{i=1}^{8} d_{pi1}, ..., \frac{1}{8}\sum_{i=1}^{8} d_{pin})$$

and

$$Var(D_p) = (V_{p1}, V_{p2}, ..., V_{pn}) = (\frac{1}{8}\sum_{i=1}^{8}(M_{p1} - d_{pi1})^2, ..., \frac{1}{8}\sum_{i=1}^{8}(M_{pn} - d_{pin})^2)$$

So finally, each puzzle piece will be characterized by three vectors X_p, $Mean(D_p)$, $Var(D_p)$. We note here that we use several widths to produce a puzzle series which makes the feature robust to scale. So to make the full feature robust to scale and rotation, we should choose an X_p vector that is also rotation robust. Now we will explain the vectors (X_p) selected to build color, texture and shape puzzles.

3.1.1 Color-puzzle feature

We use the first and second moments (mean and variance) of each band in the CIE L*a*b color space. We have chosen the L*a*b space because it is a color-uniform representation. So our X_p for the color puzzle will be the 6-dimension vector $X_p = (mean(l), var(l), mean(a), var(a), mean(b), var(b))$, which produces a puzzle piece vector totalling 18 dimensions after addition of the $Mean_8(D_p)$ and $Var_8(D_p)$ vectors.

3.1.2 Texture-puzzle feature

For texture we use gray-level images. The features derived from the co-occurrence matrix have been widely used in CBIR [19]. This matrix is indexed by a single displacement (δx, δy), where as usually this matrix is calculated for several displacements. In our texture-puzzle, we propose to use a featurė derived from the co-occurrence matrix [15] indexed by a distance d instead of $(\delta x, \delta y)$. So when estimating it, we check all pixels at distance d which make it invariant to rotation. We first transform our image into gray-level image. For each puzzle piece we calculate the gray-level co-occurrence using distance *d=1pixel* and derive from it the following values: Mean, Variance, Energy, Entropy, Contrast and Homogeneity. So our X_p for the texture puzzle will be the 6-dimension vector $X_p = (Mean(p), Var(p), Ener(p), entr(p), Contr(p), Homo(p))$, which again produces a 18-dimension puzzle piece vector.

3.1.3 Shape-puzzle feature

We introduce pixel type as a novel manner to characterize shape. We start by extracting the image edge map (see Figure 8). The edge map contains shape information; we will use it to build the shape-puzzle instead of using the original image. A standard shape representation that has been often used in the literature is the *edge orientation histogram* [17]. However, this representation is rotation-variant, as a result of the global rotation

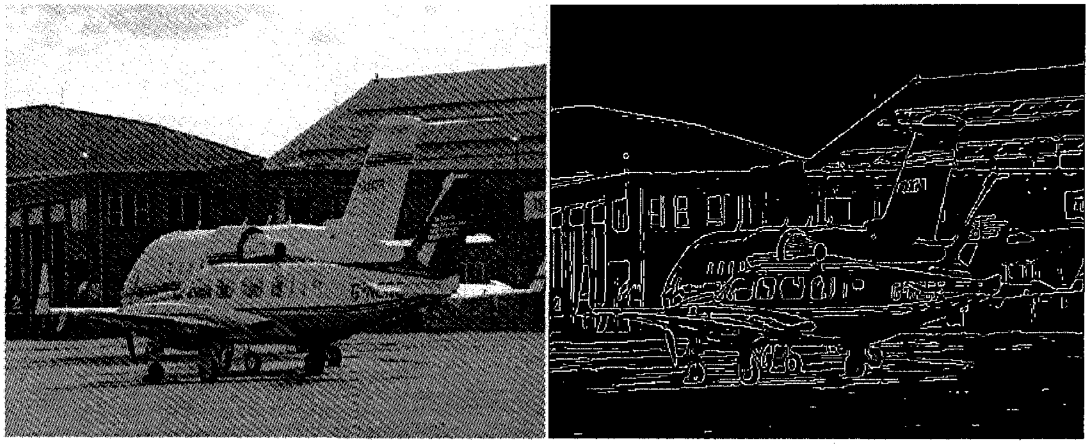


Figure 8: Image edge map

applied to produce a normalized histogram. A small error in estimating this global orientation shifts all the edge orientations. As our shape feature, we use a *differential orientation histogram.* The idea is is to characterize the edge pixel by an invariant rotation measurement. This measurement is the mean and variance of the difference in orientation between its edge pixel neighbors. Figure 9 gives a simple example of these mean and variance calculation. We use 3 bins for the mean and 3 for the variance, which results in 9 bins or edge pixel types. We use a pixel-type histogram of 9 bins as a vector

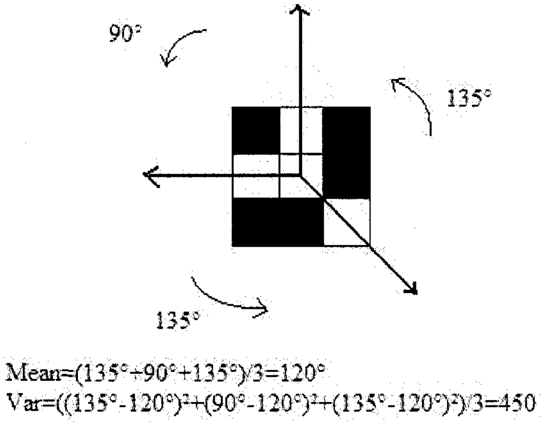

90°
135°
135°
Mean=(135°+90°+135°)/3=120°
Var=((135°-120°)²+(90°-120°)²+(135°-120°)²)/3=450

Figure 9: Example of orientation difference mean and variance estimation

X to characterize the shape in puzzle pieces. In addition to this histogram, we use the percentage of edge pixels in the puzzle piece. So our X_p for the shape puzzle will be a 10-dimension vector.

3.2 Other features used

For comparison purposes, in addition to color, texture, and shape puzzles, we also use the well-known SIFT [1] feature [23] which is identified in [28] as one of the best local features existing in the literature.

4 Experiments

To evaluate our CBIR, we used the *Microsoft Research Cambridge Image Database* [2] (retrieval, classification), which contains 4323 images (Figure 10 gives image examples). It is organized hierarchically, and has 33 leaves, as shown in Figure 11. 4158 images will be used in the model learning phase, and the remaining 165 images (5 images per leaf) will

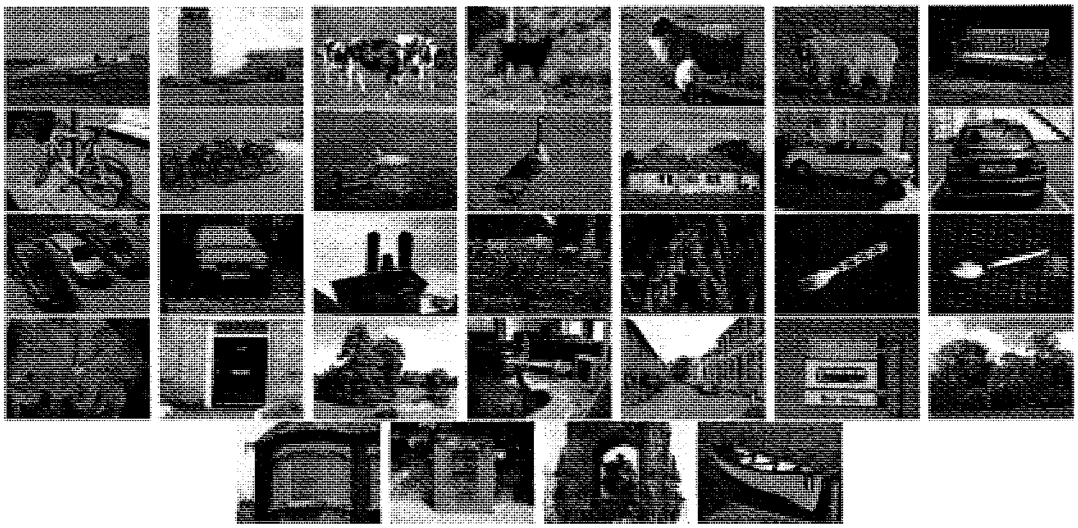


Figure 10: Examples of images from the collection

[1] http://www.cs.ubc.ca/~lowe/keypoints/

[2] http://www.research.microsoft.com/vision/cambridge/recognition/default.htm

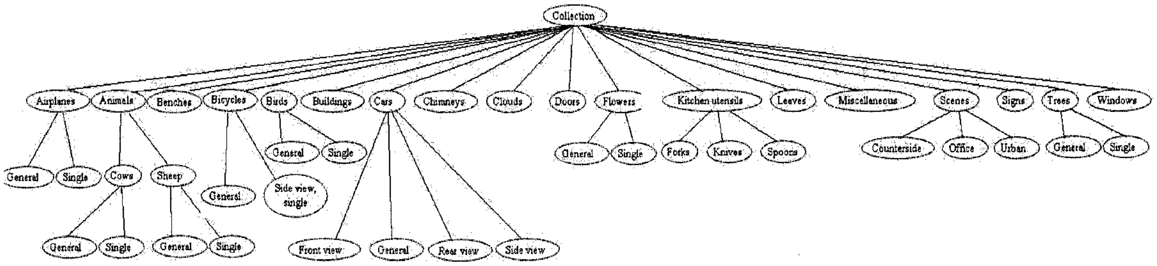


Figure 11: The hierarchical structure of the collection

be used as described below. We carried out three experiments, each intended to measure a specific aspect of our system. The first measures the clustering performance and the improvement achieved by using feature weights. The aim of the second experiment is to evaluate the system retrieval performance. The third test is intended to evaluate the retrieval performance using external images (the 165 images); we make this test more challenging by applying several transformations to the images.

First Experiment: Each image in the learning database will be a query. This test will be done for each feature alone, and after that, all features are used. Table 1 gives the precision results for all collection leaves using color-puzzle, shape-puzzle, texture-puzzle, and SFIT. Table 2 gives the precision results of combining features without weighting, and using our weighting schema. A substantial improvement in classification rate is achieved with our feature weighting schema.

Second Experiment: We randomly choose 15 images for each leaf, and use them as queries. Each time, we check the number of similar images in the 10 top ranked images and the 20 top ranked images. We do a sequential retrieval by traversing the whole collection, and a retrieval using our model. Table 3 gives the precision results for all collection nodes using all features. We find that our model improves semantic retrieval performance, relative to sequential retrieval.

Third Experiment: Using only our model, we do the same test as in the second experiment with the remaining images (the 165 images). Six transformations are applied to each images, giving us a total of 1155 images. The transformations are: zoom, rotation, zoom+rotation, deformation, image blur, and light change, as shown in Figure 12. Tables

Node name	Color-puzzle	Shape-puzzle	Texture-puzzle	SIFT
General-airplanes	87%	98%	53%	40%
Single-airplanes	80%	67%	100%	27%
General-cows	93%	80%	13%	20%
Single-cows	87%	67%	7%	13%
General-sheep	80%	67%	73%	67%
Single-sheep	87%	93%	7%	33%
Benches	80%	60%	40%	40%
General-bicycles	80%	93%	0%	67%
Side view-bicycles	73%	93%	93%	53%
General-birds	84%	57%	71%	14%
Single-birds	80%	27%	0%	47%
Buildings	63%	47%	13%	27%
Front view-cars	87%	100%	60%	60%
General-cars	67%	56%	0%	22%
Rear view-cars	80%	100%	47%	47%
Side view-cars	80 %	100%	73 %	67 %
Chimneys	93%	87%	27%	27%
Clouds	100%	80%	100%	33%
Doors	80%	87%	33%	67%
General-flowers	70%	40%	33%	47%
Single-flowers	80%	80%	67%	67%
Forks	60%	33%	20%	47%
Knives	80%	93%	47%	60%
Spoons	87%	100%	67%	73%
Leaves	90%	94%	13%	47%
Miscellaneous	67%	40%	7%	27%
Countryside-scenes	90%	47%	20%	27%
Office-scenes	87%	100%	47%	33%
Urban-scenes	80%	87%	53%	13%
Signs	8%	93%	27%	8%
General-trees	73%	87%	13%	33 %
Single-trees	87%	100%	47%	40%
Windows	74%	60%	27%	53%
Global	80%	76%	40%	43%

Table 1: Classification rate for the first experiment using features separately

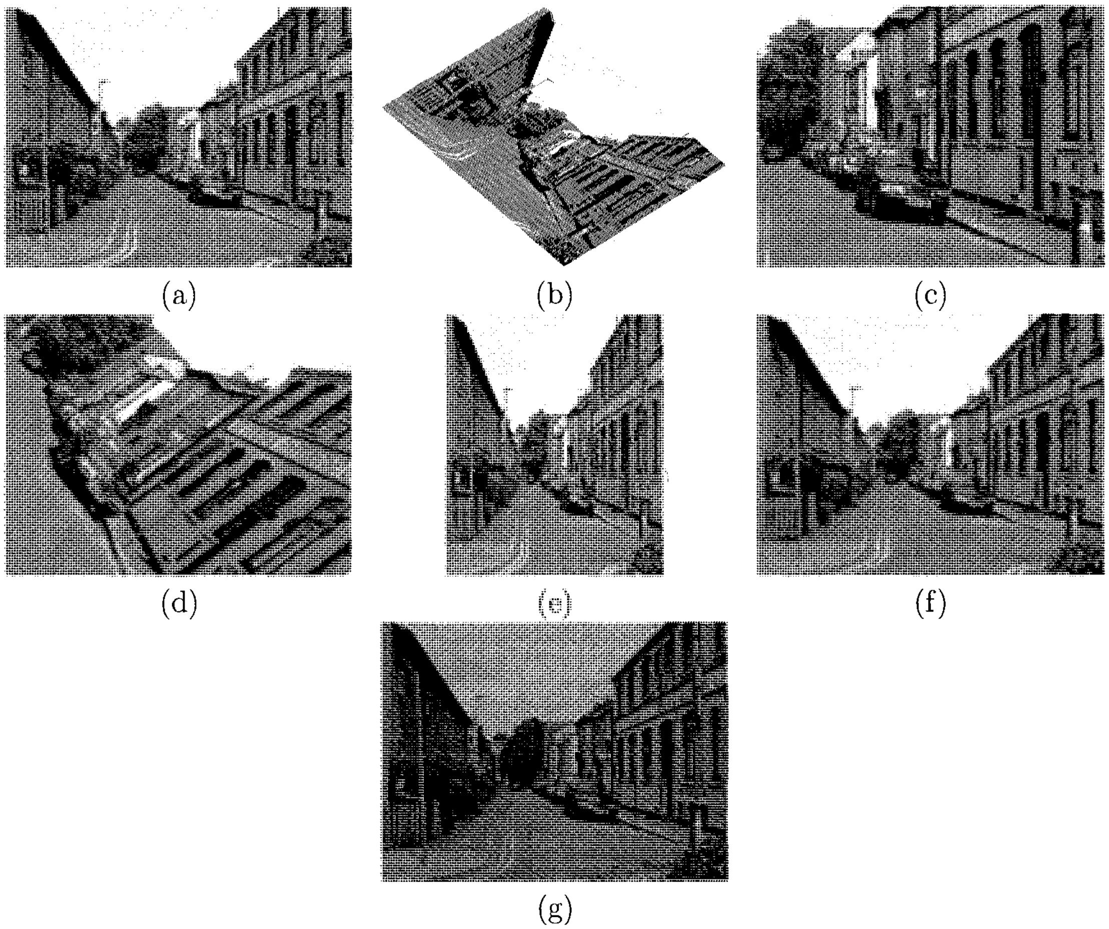
4 and 5 give the global result for all transformations. On the average, we achieve a retrieval rate of 53% similar images in the 10 top ranked images, and 47.3% in the 20 top ranked images for all transformations. This proves the robustness of our system relative to the transformations used.

Node name	Without weights	Feature weighting
General-airplanes	80%	87%
Single-airplanes	87%	100%
General-cows	73%	87%
Single-cows	80%	100%
General-sheep	67%	100%
Single-sheep	87%	87%
Benches	80%	100%
General-bicycles	50%	93%
Side view-bicycles	93%	100%
General-birds	67%	80%
Single-birds	54%	80%
Buildings	63%	100%
Front view-cars	70%	78%
General-cars	40%	100%
Rear view-cars	67%	100%
Side view-cars	87%	100%
Chimneys	80%	93%
Clouds	93%	100%
Doors	67%	87%
General-flowers	74%	100%
Single-flowers	87%	100%
Forks	44%	60%
Knives	80%	93%
Spoons	87%	100%
Leaves	93%	93%
Miscellaneou	53%	60%
Countryside-scenes	77%	93%
Office-scenes	80%	93%
Urban-scenes	87%	93%
Signs	67%	87%
General-trees	80%	93%
Single-trees	87%	93%
Windows	74%	80%
Global	74.4%	91%

Table 2: Classification rate for the first experiment using all features, with weights and without

Node name	10 top ranked images		20 top ranked images	
	Our model	Sequential retrieval	Our model	Sequential retrieval
General-airplanes	87%	37%	87%	35%
Single-airplanes	100%	57%	100%	50%
General-cows	87%	36%	87%	34%
Single-cows	100%	32%	100%	34%
General-sheep	100%	56%	100%	50%
Single-sheep	87%	53%	87%	49%
Benches	100%	46%	100%	44%
General-bicycles	93%	52%	93%	50%
side view-bicycles	100%	41%	100%	36%
General-birds	56%	14%	28%	10%
Single-birds	80%	29%	80%	24%
Buildings	100%	25%	100%	28%
Front view-cars	87%	29%	43%	26%
General-cars	90%	7%	45%	9%
Rear view-cars	100%	77%	100%	70%
Side view-cars	100%	65%	100%	50%
Chimneys	93%	55%	93%	60%
Clouds	100%	91%	100%	85%
Doors	87%	51%	87%	42%
General-flowers	100%	30%	100%	27%
Single-flowers	100%	35%	100%	30%
Forks	60%	22%	60%	21%
Knives	93%	50%	93%	49%
Spoons	100%	49%	100%	50%
Leaves	93%	41%	93%	40%
Miscellaneous	60%	13%	60%	11%
Countryside-scenes	93%	44%	93%	42%
Office-scenes	93%	56%	93%	50%
Urban-scenes	93%	58%	93%	50%
Signs	87%	30%	87%	27%
General-trees	93%	50%	93%	48%
Single-trees	93%	35%	93%	33%
Windows	80%	45%	80%	45%
Global	90.4%	42.8%	86.9%	39.7%

Table 3: Retrieval rate for all queries in the second experiment



(a) (b) (c)

(d) (e) (f)

(g)

Figure 12: Image transformations: (a) original image, (b) rotation, (c) zoom, (d)rotation+zoom, (e) deformation, (f) blur, (g) light change

Node name	Transformation					
	Original image		Rotation		Zoom	
	10 top ranked images	20 top ranked images	10 top ranked images	20 top ranked images	10 top ranked images	20 top ranked images
General-airplanes	50%	44%	47%	43%	44%	40%
Single-airplanes	64%	55%	63%	58%	54%	50%
General-cows	72%	65%	65%	60%	48%	32%
Single-cows	40%	55%	40%	44%	32%	30%
General-sheep	44%	50%	46%	45%	30%	20%
Single-sheep	38%	45%	33%	44%	36%	35%
Benches	68%	65%	64%	55%	48%	43%
General-bicycles	84%	85%	79%	75%	76%	70%
Side view-bicycles	52%	55%	55%	65%	43%	30%
General-birds	20%	27%	18%	14%	20%	15%
Single-birds	34%	30%	30%	19%	34%	21%
Buildings	76%	80%	77%	75%	50%	33%
Front view-cars	54%	50%	50%	55%	68%	60%
General-cars	60%	53%	60%	49%	56%	44%
Rear view-cars	56%	46%	50%	44%	82%	80%
Side view-cars	76%	70%	70%	60%	70%	60%
Chimneys	70%	80%	72%	75%	84%	85%
Clouds	100%	94%	100%	90%	100%	90%
Doors	94%	90%	87%	95%	88%	65%
General-flowers	48%	50%	48%	50%	46%	36%
Single-flowers	34%	44%	40%	43%	62%	50%
Forks	50%	40%	47%	35%	44%	30%
Knives	60%	63%	56%	54%	64%	41%
Spoons	66%	68%	59%	63%	60%	40%
Leaves	68%	55%	70%	65%	58%	55%
Miscellaneous	56%	55%	54%	50%	66%	60%
Countryside-scenes	66%	50%	63%	55%	44%	40%
Office-scenes	55%	47%	52%	44%	40%	43%
Urban-scenes	48%	28%	43%	35%	42%	38%
Signs	72%	70%	69%	66%	72%	70%
General-trees	50%	47%	55%	45%	28%	31%
Single-trees	55%	42%	49%	48%	30%	27%
Windows	94%	90%	92%	91%	74%	70%
Global	60%	57%	57.7%	54.8%	54.3%	46.5%

Table 4: Retrieval rate for original image, rotation, and zoom in the third experiment, using weighting features.

Node name	Transformation							
	Rotation+zoom		Deformation		Blur		Light change	
	10 top ranked images	20 top ranked images	10 top ranked images	20 top ranked images	10 top ranked images	20 top ranked images	10 top ranked images	20 top ranked images
General-airplanes	44%	36%	56%	48%	36%	34%	30%	28%
Single-airplanes	42%	28%	46%	38%	40%	40%	48%	38%
General-cows	35%	33%	40%	50%	48%	46%	22%	14%
Single-cows	56%	48%	72%	60%	36%	37%	26%	20%
General-sheep	40%	31%	48%	30%	44%	34%	46%	39%
single-sheep	42%	29%	44%	34%	50%	40%	32%	23%
Benches	44%	82%	58%	55%	66%	60%	30%	22%
General-bicycles	72%	70%	76%	70%	60%	48%	90%	90%
Side view-bicycles	64%	60%	46%	41%	52%	40%	84%	85%
General-birds	18%	17%	20%	13%	15%	13%	18%	11%
Single-birds	20%	10%	23%	18%	32%	23%	23%	12%
Buildings	45%	42%	52%	50%	88%	75%	42%	32%
Front view-cars	64%	60%	66%	55%	48%	37%	32%	22%
General-cars	34%	24%	40%	34%	40%	38%	36%	18%
Rear view-cars	82%	85%	78%	50%	64%	30%	44%	33%
Side view-cars	70%	60%	66%	60%	72%	38%	32%	32%
Chimneys	80%	66%	76%	70%	72%	70%	74%	70%
Clouds	95%	84%	96%	80%	100%	90%	52%	50%
Doors	62%	60%	80%	75%	92%	85%	52%	40%
General-flowers	54%	50%	35%	25%	44%	36%	56%	50%
Single-flowers	44%	39%	40%	36%	31%	28%	68%	65%
Forks	29%	20%	44%	31%	44%	39%	31%	19%
Knives	54%	48%	50%	27%	64%	40%	44%	33%
Spoons	47%	44%	48%	33%	60%	43%	39%	31%
Leaves	74%	65%	64%	49%	54%	48%	36%	35%
Miscellaneous	56%	60%	54%	55%	56%	52%	28%	35%
Countryside-scenes	40%	41%	44%	45%	66%	55%	48%	55%
Office-scenes	38%	30%	32%	28%	36%	31%	38%	39%
Urban-scenes	47%	41%	43%	36%	30%	23%	40%	36%
Signs	58%	60%	52%	49%	74%	75%	32%	35%
General-trees	24%	23%	48%	37%	19%	25%	20%	14%
Single-trees	35%	30%	50%	47%	40%	32%	26%	20%
Windows	58%	50%	90%	85%	94%	90%	25%	23%
Global	50.5%	46.3%	53.9%	45.9%	53.5%	45.3%	40.7%	35.4%

Table 5: Retrieval rate for rotation+zoom, deformation, and light change in the third experiment using weighting features.

5 Conclusion

In this paper, we have presented a hybrid probabilistic framework for CBIR modeling, using a generative approach for node parameter estimation, and a discriminative approach for feature weighting. We proved that defining feature weights for each node increases the systems clustering performance. We found also that processing retrieval with our model improves the semantic retrieval result, compared to sequential retrieval. So using a probabilistic framework for CBIR gives a semantic property to retrieval, narrowing the gap between human perception and low-level features. Furthermore, we proposed an algorithm to extract local color, shape, and texture features. Our experiments showed that our local shape feature performs better than the SIFT descriptor. In future work, we should apply our probabilistic framework to larger collections, and use more features. Indeed, increasing the number of images and features will give better assessment to the validity of our model.

References

[1] H.K. Ahn, N. Mamoulis, and H.M. Wong. A Survey on Multidimensional Access Methods. *Research report, Hong Kong University of Science and Technology, Hong Kong,* 1997.

[2] M. Banerjee and M.K. Kundu. Edge Based Features for Content Based Image Retrieval. *Pattern Recognition*, vol.36, no.11, p.2649-2661, November 2003.

[3] K. Barnard and D. Forsyth. Learning the Semantics of Words and Pictures. *International Conference of Computer Vision*, vol.2, p.408-415, 2001.

[4] N. Bouguila and D. Ziou. A Hybrid SEM Algorithm for High-Dimensional Unsupervised Learning Using a Finite Generalized Dirichlet Mixture. *Image Processing, IEEE Transactions on.* vol.15, no.9, p.2657- 2668, Sept 2006.

[5] G. Carneiro, A.B. Chan, P.J. Moreno, and N. Vasconcelos. Supervised Learning of Semantic Classes for Image Annotation and Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.29, no.3, p.394-410, March 2007.

[6] H. Chang and D.Y. Yeung. Kernel-Based Distance Metric Learning for Content-Based Image Retrieval. *Image and Vision Computing*, vol.25, no.5, p.695-703, 2007.

[7] M.H. Chang, J.Y. Pyun, M.B. Ahmed, J.H. Chun, and J.A. Park. Modified Color Co-Occurrence Matrix for Image Retrieval. *LNCS, Advances in Natural Computation*, vol.3611, p.43-50, 2005.

[8] T.S. Choras, T. Andrysiak, and M. Choras. Integrated Color, Texture and Shape Information for Content-Based Image Retrieval. *Pattern Analysis and Applications*, On line, first accessed, 2007.

[9] T.W.S. Chow, M.K.K. Rahman, and S. Wu. Content-Based Image Retrieval by using Tree-Structured Features and Multi-Layer Self-Organizing Map. *Pattern Analysis and Applications*, vol.9, no.1, p.1-20, 2006.

[10] G. Das, S. Ray, and C. Wilson. Feature Re-Weighting in Content-Based Image Retrieval. *LNCS, Image and Video Retrieval*, vol.4071, p.193-200, 2006.

[11] L. Fei-Fei, R. Fergus, and P. Perona. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. *CVPR , Workshop on Generative-Model Based Vision*, vol.12, p.178, 2004.

[12] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. *Proceedings of the Tenth IEEE International Conference on Computer Vision*, vol.2, p.1458-1465, 2005.

[13] A.D Holub and P. Perona . A Discriminative Framework for Modelling Object Classes. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on Publication*, voll, p.664-671, 2005.

[14] A.D. Holub, M. Welling, and P. Perona. Combining Generative Models and Fisher Kernels for Object Recognition. *Proceedings of the Tenth IEEE International Conference on Computer Vision*, vol.1, p.136-143, 2005.

[15] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih. Image Indexing using Color Correlograms. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p.762, 1997.

[16] M.L. Kherfi and D. Ziou. Image Collection Organization and Its Application to Indexing, Brosing, Summarization, and Semantic Retrieval. *IEEE Transactions On Multimedia*, vol.9, no.4, June 2007.

[17] M.L. Kherfi and D. Ziou. Image Retrieval Based on Feature Weighting and Relevance Feedback. *Image Processing, International Conference on*, vol.1, p.689-692, October 2004 .

[18] M.L. Kherfi and D. Ziou. Image Retrieval from the World Wide Web: Issues, Techniques and Systems. *ACM computing surveys*, vol.36, no.1, p.35-67, March 2004.

[19] M.L. Kherfi and D.Ziou. Relevance Feedback for CBIR: A New Approch Based on Probabilistic Feature Weighting with Positive and Negative Examples. *IEEE Transactions on Image Processing*, vol.15, no.4, p.1017-1030, April 2006.

[20] N.W. Kim, T.Y. Kim, and J.S. Choi. Edge Based Spatial Descreptor using Color Vector Angle for Effective Image Retrieval. *LNCS, Modeling Decisions for Artificial Intelligence*, vol.3558, p.365-375, 2005.

[21] D. Liang, J. Yang, J.J Lu, and Y.C Chang. Image Retrieval using Weighted Color Co-Occurrence Matrix. *LNCS*, vol.3567, p.161-165, 2005.

[22] H. Ling, W. Lingda, C. Yichao, and L. Yuchi. Indexing Structures for Content-Based Retireval of Large Image Databases: A Review. *LNCS*, vol.3689, p.626-634, 2005.

[23] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, vol.60, no.2, p.91-110, 2004.

[24] Y. Lu, Q. Zhao, J. Kong, C. Tanh, and Y. Li. A Two-stage Region-Based Image Retrieval Approach Using Combined Color and Texture Features. *LNCS*, vol.4304, p.1010-1014, 2006.

[25] A. Marakakis, N. Galatsanos, A. Likas, and A. Stafylopatis. A Relevance Feedback Approach for Content Based Image Retrieval Using Gaussian Mixture Models. *LNCS*, vol.4132, p.84-93, 2006.

[26] T. Mihran and K.J. Anil. Texture Analysis. In C.H. Chen, L.F. Pau, P.S.P. Wang(eds.), *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, p.207-248, World Scientific Publishing Co, 1998.

[27] K. Mikolajczyk and C. Schmid. Indexing Based on Scale Invariant Interest Point. *Proceedings Eighth IEEE International Conference Computer Vision*, vol.1, p.525-531, 2001.

[28] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Matchine Intelligence*, vol.27, no.10, p.1615-1630, October 2005.

[29] A. Ng and M. Jordan. On Discriminative Vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes. In T. Dietterich, S. Becker and Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems 14*, Cambridge, MA: MIT Press, 2002.

[30] G. Pass and R. Zabih. Histogram Refinement for Content-Based Image Retrieval. *Processing of the IEEE Workshop on Application of Computer Vision*, Sarasota 1996.

[31] G. Qiu. Color Image Indexing using BTC. *IEEE Transaction on Image Processing*, vol.12, no.1, p.93-101, January 2003.

[32] Y. Rubner, J. Puzicha, C. Tomasi, and J.M. Buhmann. Empirical Evaluation of Dissimilarity Measures for Color and Texture. *Computer Vision, The Proceedings of the Seventh IEEE International Conference on*, vol.2, p.1165-1172, 1999.

[33] L. Si, R. Jin, S.C.H. Hoi, and M. R. Lyu. Collaborative Image Retrieval via Regularized Metric Learning. *Multimedia Systems* vol.12, no.1, p.34-44, 2006.

[34] V. Takala, T. Ahonen, and M. Pietikäinen. Block-Based Methods for Image Retrieval using Local Binary Patters. *LNCS*, vol.3540, p.882-891, 2005.

[35] A. Vailaya, A.K. Jain, and H.J. Zhang. Image Classification for Content-Based Indexing. *IEEE Transactions on Image Processing*, vol.10, no.1, p.117-130, January 2001.

[36] R. C. Veltkamp and M. Tanase. Content-Based Image Retrieval Systems: A Survey. *Revised and extended version of Technical Report UU-CS-2000-34*, October 28, 2002.

[37] Q. Wu, C. Zhou, and C. Wang. Content-Based Affective Image Classification and Retrieval by using Support Vector Machines. *Proceedings of the Tenth IEEE International Conference on Computer Vision*, vol.1, p.136-143, 2005.

[38] I. Yahiaoui, N. Hervé, and N. Boujemaa. Shape-Based Image Retrieval in Botanical Collections. *LNCS*, vol.4261, p.357-364, 2006.

[39] H. Zhang, A.C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.2, p.2126-2136, 2006.

[40] X.S. Zhou and T.S. Huang. Edge Based Structural Features for Content Based Image Retrieval. *Pattern Recognition Letters*, vol.22, no.5, p.457-468, April 2001.

Appendices

A Feature weights

$$L_\Omega = \prod_{m=1}^{N_\Omega}\prod_{n=1}^{N_o}\prod_{r=1}^{N_r}\prod_{f=1}^{N_f}(\frac{p(a_{o_n rf} \mid \theta_m)}{\sum_{k\epsilon\Omega} p(a_{o_n rf} \mid \theta_k)})^{1/\sigma_{mf}} \tag{9}$$

where N_Ω is the number of nodes in Ω, N_o the number of visual documents of node m, N_r the number of regions of interest in visual document o, N_f the number of features used, and σ_{mf} the weight of feature f for the node m. In order to maximize the log of L_Ω under the constraints $\sum_{f=1}^{N_f}\sigma_{mf} = 1$ for $1 \leq m \leq N_\Omega$, we introduce a Lagrange multiplier λ_m for each constraint. We obtain:

$$L(\sigma_{mf}, \lambda_m)_{1\leq m\leq N_\Omega, 1\leq f\leq N_f} = \sum_{m=1}^{N_\Omega}\sum_{n=1}^{N_o}\sum_{r=1}^{N_r}\sum_{f=1}^{N_f}\frac{1}{\sigma_{mf}}log(\frac{p(a_{o_n rf} \mid \theta_m)}{\sum_{k\epsilon\Omega} p(a_{o_n rf} \mid \theta_k)})+\sum_{m=1}^{N_\Omega}\lambda_m(1-\sum_{f=1}^{N_f}\sigma_{mf})$$

To simplify notation, let us set $A_{mf} = \sum_{n=1}^{N_o}\sum_{r=1}^{N_r} log(\frac{p(a_{o_n rf}|\theta_m)}{\sum_{k\epsilon\Omega} p(a_{o_n rf}|\theta_k)})$. We obtain:

$$L(\sigma_{mf}, \lambda_m)_{1\leq m\leq N_\Omega, 1\leq f\leq N_f} = \sum_{m=1}^{N_\Omega}\sum_{f=1}^{N_f}\frac{1}{\sigma_{mf}}A_{mf} + \sum_{m=1}^{N_\Omega}\lambda_m(1-\sum_{f=1}^{N_f}\sigma_{mf})$$

We have:

$\frac{\partial L(\sigma_{mf},\lambda_m)}{\partial\sigma_{ij}} = \frac{-A_{ij}}{\sigma_{ij}^2} - \lambda_i = 0$

$$\Rightarrow \sigma_{ij} = \sqrt{-\frac{A_{ij}}{\lambda_i}} \tag{10}$$

We have also:

$\frac{\partial L(\sigma_{mf},\lambda_m)}{\partial\lambda_i} = 1 - \sum_{f=1}^{N_f}\sigma_{if} = 0 \Rightarrow \sum_{f=1}^{N_f}\sigma_{if} = 1 \Rightarrow \sum_{f=1}^{N_f}\sqrt{-\frac{A_{if}}{\lambda_i}} = 1$

$$\Rightarrow \lambda_i = (\sum_{f=1}^{N_f}\sqrt{-A_{if}})^2 \tag{11}$$

10 and 11 $\Rightarrow$

$$\sigma_{ij} = \frac{\sqrt{-A_{ij}}}{\sum_{f=1}^{N_f}\sqrt{-A_{if}}} \tag{12}$$

Since $\frac{\partial^2 L(\sigma_{mf},\lambda_m)}{\partial^2 \sigma_{ij}} = \frac{2A_{ij}}{\sigma_{ij}^3} < 0$, then Equation 12 is a maximum for Equation 9. We obtain:

$$\sigma_{ij} = \frac{\sqrt{-\sum_{n=1}^{N_o}\sum_{r=1}^{N_r} log(\frac{p(a_{o_n rj}|\theta_i)}{\sum_{k\epsilon\Omega} p(a_{o_n rj}|\theta_k)})}}{\sum_{f=1}^{N_f}\sqrt{-\sum_{n=1}^{N_o}\sum_{r=1}^{N_r} log(\frac{p(a_{o_n rf}|\theta_i)}{\sum_{k\epsilon\Omega} p(a_{o_n rf}|\theta_k)})}} \tag{13}$$

B EM algorithm

Our likelihood for each collection node is defined by:

$$L_m = \prod_{f=1}^{N_f}\prod_{n=1}^{N_o}\prod_{r=1}^{N_r}(\sum_{i=1}^{K_{mf}} p_{mfi}p(a_{o_n rf} \mid \theta_{mfi}))$$

We maximize its log under the constraint $\sum_{i=1}^{K_{mf}} p_{mfi} = 1$ for each feature f, using a Lagrange multiplier λ_f for each constraint we obtain the following log likelihood lagrangian:

$$LL_m = \sum_{f=1}^{N_f}\sum_{n=1}^{N_o}\sum_{r=1}^{N_r}\log(\sum_{i=1}^{K_{mf}} p_{mfi}p(a_{o_n rf} \mid \theta_{mfi})) + \sum_{f=1}^{N_f}\lambda_f(1 - \sum_{i=1}^{K_{mf}} p_{mfi})$$

The derivative of this log likelihood with respect to the mixture parameters for each feature is independent of the other features, so we can maximize the likelihood for each feature separately. We define the log likelihood lagrangian for a node m and a feature f by:

$$LL_{m,f} = \sum_{n=1}^{N_o}\sum_{r=1}^{N_r}\log(\sum_{i=1}^{K_{mf}} p_{mfi}p(a_{o_n rf} \mid \theta_{mfi})) + \lambda_f(1 - \sum_{i=1}^{K_{mf}} p_{mfi}) \tag{14}$$

B.1 Calculation of p_{mfj}

We have

$$\frac{\partial LL_{m,f}}{\partial p_{mfj}} = \sum_{n=1}^{N_o}\sum_{r=1}^{N_r}\frac{\partial \log(\sum_{i=1}^{K_{mf}} p_{mfi}p(a_{o_n rf} \mid \theta_{mfi}))}{\partial p_{mfj}} - \lambda_f$$

$\Rightarrow$

$$\frac{\partial LL_{m,f}}{\partial p_{mfj}} = \sum_{n=1}^{N_o}\sum_{r=1}^{N_r}\frac{1}{\sum_{i=1}^{K_{mf}} p_{mfi}p(a_{o_n rf} \mid \theta_{mfi})}p(a_{o_n rf} \mid \theta_{mfj}) - \lambda_f$$

We have $p(\theta_{mfj} \mid a_{o_n rf}) = \frac{p_{mfj}p(a_{o_n rf}|\theta_{mfj})}{\sum_{i=1}^{K_{mf}} p_{mfi}p(a_{o_n rf}|\theta_{mfi})}$

$\Rightarrow$

$$\frac{\partial LL_{m,f}}{\partial p_{mfj}} = \sum_{n=1}^{N_o}\sum_{r=1}^{N_r} p(\theta_{mfj} \mid a_{o_n rf})\frac{1}{p_{mfj}} - \lambda_f$$

$\frac{\partial LL_{m,f}}{\partial p_{mfj}} = 0 \Rightarrow$

$$p_{mfj} = \frac{1}{\lambda_f}\sum_{n=1}^{N_o}\sum_{r=1}^{N_r} p(\theta_{mfj} \mid a_{o_n rf}) \tag{15}$$

We also have

$$\frac{\partial LL_{m,f}}{\partial \lambda_f} = 1 - \sum_{i=1}^{K_{mf}} p_{mfi}$$

$\frac{\partial LL_{m,f}}{\partial \lambda_f} = 0$

$$\sum_{i=1}^{K_{mf}} p_{mfi} = 1 \tag{16}$$

Equation (15) and (16) $\Rightarrow$

$$\sum_{i=1}^{K_{mf}} \frac{1}{\lambda_f}\sum_{n=1}^{N_o}\sum_{r=1}^{N_r} p(\theta_{mfi} \mid a_{o_n rf}) = 1$$

$\Rightarrow$

$$\lambda_f = \sum_{i=1}^{K_{mf}}\sum_{n=1}^{N_o}\sum_{r=1}^{N_r} p(\theta_{mfi} \mid a_{o_n rf})$$

$\Rightarrow$

$$\lambda_f = \sum_{n=1}^{N_o}\sum_{r=1}^{N_r}\sum_{i=1}^{K_{mf}} p(\theta_{mfi} \mid a_{o_n rf})$$

$$\lambda_f = \sum_{n=1}^{N_o}\sum_{r=1}^{N_r}(1) = N_o N_R$$

Replacing λ_f value in equation 15, we obtain:

$$p_{mfj} = \frac{\sum_{n=1}^{N_o}\sum_{r=1}^{N_r} p(\theta_{mfj} \mid a_{o_n rf})}{N_o N_R} \tag{17}$$

B.2 Calculation of θ_{mfj}

We have chosen to use the generalized Dirichlet distribution, whose PDF is given by:

$$p(X_1, ..., X_d) = \prod_{i=1}^{d} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} X_i^{\alpha_i - 1}(1 - \sum_{j=1}^{i} X_j)^{\gamma_i}$$

where $\sum_{i=1}^{d} X_i < 1$, $0 < X_i < 1$ for $i = 1, ..., d$, $\gamma_i = \beta_i - \alpha_{i+1} - \beta_{i+1}$ for $i = 1, ..., d-1$, and $\gamma_d = \beta_d - 1$.

So θ_{mfj} is defined by $\{(\alpha_{mfj1}, \beta_{mfj1}), ..., (\alpha_{mfjd}, \beta_{mfjd})\}$. If a random vector $\overrightarrow{X} = (X_1, X_2, ..., X_d)$ follows a generalized Dirichlet distribution, then we can construct a vector $\overrightarrow{W} = (W_1, W_2, ..., W_d)$ using the following transformation $W_i = T(X_i)$:

$$T(X_i) = \begin{cases} X_i & \text{if } i = 1 \\ \frac{X_i}{1 - X_1 - ... - X_{i-1}} & \text{for } i = 2, 3, ..., d \end{cases}$$

where each W_i, $i = 1, .., d$ follows a Beta distribution with parameter α_i and β_i, and $\{\alpha_i, \beta_i,\ i = 1, .., d\}$ defines the generalized Dirichlet distribution which characterizes $\overrightarrow{X}$. Bouguila and Ziou [4] use this transformation to reduce the problem of estimating a d-dimension generalized Dirichlet mixtures parameters to the estimation of d Beta mixtures. The PDF of the Beta distribution is given by:

$$p_{Beta}(W_i) = \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} W_i^{\alpha_i - 1}(1 - W_i)^{\beta_i - 1}$$

We obtain:

$$p(X_1, ..., X_d) = \prod_{i=1}^{d} p_{Beta}(W_i) = \prod_{i=1}^{d} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} W_i^{\alpha_i - 1}(1 - W_i)^{\beta_i - 1}$$

Using the EM algorithm proposed by Bouguila and Ziou, to maximize Equation 14, we must maximize the log likelihood for every dimension h (for $h = 1, 2, ..., d$):

$$LL_{m,f,W_h} = \sum_{n=1}^{N_o} \sum_{r=1}^{N_r} \log\left(\sum_{i=1}^{K_{mf}} p_{mfi}\ p_{beta}(W_{nrfh} \mid \theta_{mfih})\right) \tag{18}$$

where $\theta_{mfih} = (\alpha_{mfih}, \beta_{mfih})$.

We need to calculate the partial derivatives. Since θ_{mfjh} is independent from n and r, we can replace $\sum_{n=1}^{N_o} \sum_{r=1}^{N_r}$ by $\sum_{u=1}^{N_u}$ where $N_u = N_o N_r$. We obtain:

$$LL_{m,f,W_h} = \sum_{u=1}^{N_u} \log\left(\sum_{i=1}^{K_{mf}} p_{mfi}\ p_{beta}(W_{ufh} \mid \theta_{mfih})\right) \tag{19}$$

The likelihood in equation (19) is the same developed by Bouguila and Ziou. We refer the reader to reference [4] for further details on the calculation of these derivatives and the Fisher's scoring method used to maximize the likelihood.

C Data normalization

In the definition of the generalized Dirichlet distribution, we have two conditions on the data: $\sum_{i=1}^{d} X_i < 1$ and $0 < X_i < 1$ for $i = 1, ..., d$. Since we can't be sure that the data we are using satisfy these conditions, we must apply a transformation to the data. The simplest transformation that we can use is normalization. Consider $\overrightarrow{X} = (X_1, ..., X_d)$, where $m < X_i < M$ for $i = 1, ..., d$. To satisfy the two conditions of the generalized Dirichlet, we normalize X_i by the transformation:

$$Y_i = \frac{X_i - m}{(M - m) * d} \text{ for } i = 1, ..., d$$

We obtain $Y_i \in]0, 1/d[$, so the value interval of Y_i depends on the data dimension. According to the EM proposed by [4], Y_i will also be transformed by:

$$W_i = T(Y_i) = \begin{cases} Y_i & \text{if } i = 1 \\ \frac{Y_i}{1 - Y_1 - ... - Y_{i-1}} & \text{for } i = 2, 3, ..., d \end{cases}$$

Since $Y_i \in]0, 1/d[$, applying T, we find that

$$W_i \in]0, \frac{1}{d - (i-1)}[\text{ for } i = 2, 3, ..., d$$

If we deal with data of high dimension, Y_i will belong to a small interval. For example, the SIFT vector is of dimension 128, so $Y_i \in]0, 0.0078125[$. Applying the transformation T, we obtain $W_1 \in]0, 0.0078125[$, $W_2 \in]0, 0.007874[$, ..., $W_{128} \in]0, 1[$. Since W_i follows a Beta distribution, which has only $0 \leq W_i \leq 1$ for $i = 1, ..., d$ as a condition, and to avoid working on small intervals like $]0, 0.0078125[$, we use a second transformation that still assumes the Beta condition and makes $W_i \in]0, 1[$ for all dimensions:

$$T_2(W_i) = W_i * (d - (i - 1)) \text{ for } i = 1, 2, 3, ..., d$$

Conclusion

Dans ce mémoire, nous avons présenté une approche probabiliste hybride (générative/ discriminative) pour la modélisation et la recherche d'images par le contenu. Sur une collection d'images organisée hiérarchiquement en noeuds selon la sémantique des images, nous avons utilisé une approche générative pour l'estimation des mélanges de probabilités qui représentent l'apparence visuelle de chaque noeud dans la collection. Ensuite nous avons appliqué une approche discriminative pour l'estimation des poids des caractéristiques visuelles pour améliorer la performance de la partie générative. Nous avons montré la capacité de notre approche pour la modélisation des collections d'images, et la diminution du fossé sémantique causé par les caractéristiques visuelles de bas-niveau. Nous avons montré l'importance de définir des poids pour les descripteurs visuels dépendamment de la sémantique des images et la robustesse de notre approche aux différentes transformations qu'une image peut subir tels que le zoom, le changement de lumière, la rotation. Nous avons proposé un algorithme d'extraction de descripteurs visuels locaux de couleur, de texture et de forme. Nous avons utilisé ces descripteurs pour valider notre modèle. Selon les tests effectués, ces caractéristiques performent mieux que le descripteur SIFT identifié dans la littérature parmi les meilleurs descripteurs visuels locaux proposés. En perspective, il reste à valider le modèle sur de grandes collections d'images et à l'étendre à la vidéo.

Bibliographie

[1] H.K. Ahn, N. Mamoulis, and H.M. Wong. A Survey on Multidimensional Access Methods. *Research report, Hong Kong University of Science and Technology, Hong Kong,* 1997.

[2] K. Barnard and D. Forsuth. Learning the Semantics of Words and Pictures. *International Conference of Computer Vision*, vol.2, p.408-415, 2001.

[3] G. Das, S. Ray, and C. Wilson. Feature Re-Weighting in Content-Based Image Retrieval. *LNCS, Image and Video Retrieval*, vol.4071, p.193-200, 2006.

[4] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih. Image Indexing using Color Correlograms. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* p.762, 1997.

[5] M.L. Kherfi and D. Ziou. Image Collection Organization and Its Application to Indexing, Brosing, Summarization, and Semantic Retrieval. *IEEE Transactions On Multimedia*, vol.9, no.4, June 2007.

[6] M.L. Kherfi and D. Ziou. Image Retrieval from the World Wide Web : Issues, Techniques and Systems. *ACM computing surveys*, vol.36, no.1, p.35-67, March 2004.

[7] M.L. Kherfi and D.Ziou. Relevance Feedback for CBIR : A New Approch Based on Probabilistic Feature Weighting with Positive and Negative Examples. *IEEE Transactions on Image Processing*, vol.15, no.4, p.1017-1030, April 2006.

[8] N.W. Kim, T.Y. Kim, and J.S. Choi. Edge Based Spatial Descreptor using Color Vector Angle for Effective Image Retrieval. *LNCS, Modeling Decisions for Artificial Intelligence*, vol.3558, p.365-375, 2005.

[9] H. Ling, W. Lingda, C. Yichao, and L. Yuchi. Indexing Structures for Content-Based Retireval of Large Image Databases : A Review. *LNCS*, vol.3689, p.626-634, 2005.

[10] A. Marakakis, N. Galatsanos, A. Likas, and A. Stafylopatis. A Relevance Feedback Approach for Content Based Image Retrieval Using Gaussian Mixture Models. *LNCS*, vol.4132, p.84-93, 2006

[11] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Matchine Intelligence*, vol.27, no.10, p.1615-1630, October 2005.

[12] G. Pass and R. Zabih. Histogram Refinement for Content-Based Image Retrieval. *Processing of the IEEE Workshop on Application of Computer Vision*, Sarasota 1996.

[13] L. Si, R. Jin, S.C.H. Hoi, and M. R. Lyu. Collaborative Image Retrieval via Regularized Metric Learning. *Multimedia Systems* vol.12, no.1, p.34-44, 2006.

[14] A. Vailaya, A.K. Jain, and H.J. Zhang. Image Classification for Content-Based Indexing. *IEEE Transactions on Image Processing*, vol.10, no.1, p.117-130, January 2001.

[15] R. C. Veltkamp and M. Tanase. Content-Based Image Retrieval Systems : A Survey. *revised and extended version of Technical Report UU-CS-2000-34*, October 28, 2002.