# A Quasi-random Sampling Approach to Image Retrieval

Jun Zhou[1,2] and Antonio Robles-Kelly[1,2]
[1]National ICT Australia (NICTA),[*] Locked Bag 8001, Canberra ACT 2601, Australia
[2]RSISE, Bldg. 115, Australian National University, Canberra ACT 0200, Australia

## Abstract

*In this paper, we present a novel approach to contents-based image retrieval. The method hinges in the use of quasi-random sampling to retrieve those images in a database which are related to a query image provided by the user. Departing from random sampling theory, we make use of the EM algorithm so as to organize the images in the database into compact clusters that can then be used for stratified random sampling. For the purposes of retrieval, we use the similarity between the query and the clustered images to govern the sampling process within clusters. In this way, the sampling can be viewed as a stratified sampling one which is random at the cluster level and takes into account the intra-cluster structure of the dataset. This approach leads to a measure of statistical confidence that relates to the theoretical hard-limit of the retrieval performance. We show results on the Oxford Flowers dataset.*

## 1. Introduction

Contents-based image retrieval is an important problem in the areas of pattern recognition, computer vision and robotics, which has derived into a large body of research. Moreover, despite the emergence of commercial systems such as QBIC (Query By Image Content) [17], FourEyes [20] and SQUID (Shape Queries Using Image Databases) [11], the retrieval of the best match in a dataset to a user-supplied query image based upon similarity remains an open problem.

In general, object and image retrieval and classification techniques [19, 26, 5, 8] are based upon the summarization of the image dataset using a codebook of visual words [12, 21, 18], which are used to query the dataset so as to retrieve images that best match the query. Thus, when a query image is provided by the user, the features in the image are compared with those on the codebook. Then a measure of similarity between the images in the dataset is computed so as to retrieve the closest match.

The multidimensional nature of the image features requires organizing them in order to make indexing efficient. Clustering algorithms are one of the methods that have been used to provide improved organization of multidimensional data. Chen *et al* [4] have employed unsupervised learning so as to exploit similarity information in a graph-theoretic setting by using a dynamic clustering method. Sengupta and Boyer [23] combined the geometric hashing approach [13] with a hierarchical tree organization of the dataset, reducing the time taken to match the tables. Shokounfandeh et. al. [24] proposed an indexing algorithm to map the topology of the search tree onto a low dimensional space in order to reduce the number of similar candidates during the query. These algorithms generally rely on a model-based partitioning and organization system, which is a hybrid that uses information-theoretical criteria to hierarchically structure the dataset and pattern recognition methods to match the candidates.

For purposes of retrieval, unsupervised or supervised techniques may be used. In the case of unsupervised methods, K-nearest neighbor classifiers [10] are prevalent. In the area of supervised retrieval methods, Support Vector Machines (SVMs) [6] have been used for both image classification and retrieval. These are often used in conjunction with relevance feedback techniques [1], where the user provides on-line training information, i.e. positive and negative examples, on the retrieval results so as to cross-validate the parameters of the classifier used in the query operation. In [28], Tao *et al.* have used an SVM classifier based upon asymmetric bagging and random subspace methods to overcome overfitting and stability in the retrieval operation. In [27], a relevance feedback approach is proposed to use feature subspace extraction on a Gaussian classifier. Despite effective, SVMs may be unstable when the training sample-size is small. Further, the decision boundary of the classifier may be biased if the number of positive and negative feedback samples differ greatly from one another, i.e. if the positive and negative sample-sets are asymmetric.

In either case of a supervised or unsupervised classifi-

[*]NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

cation scheme, the design of an architecture for image retrieval requires both, an image representation suitable for search and a similarity measure that can be employed to rank the images with respect to the relevance to the query [30]. In this paper, we present a quasi-random sampling approach to image retrieval which employs a maximum a posteriori (MAP) method to cluster the database for purposes of efficiency. In this manner, each cluster becomes a strata from which quasi-random sampling can be effected. Here, the number of samples randomly drawn from each strata is determined by the likelihood of the best match being in the cluster of interest. The method has a number of desirable attributes. Firstly, it avoids user intervention. Secondly, due to the probabilistic approach taken here, bounds of accuracy can be drawn and a margin of confidence can be computed. Thirdly, it permits achieving a performance comparable to intensive search with a fraction of the computational cost. Finally, its quite general in nature and permits the use of a number of image features, similarity measures and classification methods elsewhere in the literature.

## 2. Preliminaries

In this section, we depart from the random sampling setting so as to introduce a number of relations that are used throughout the paper. In order to perform stratified quasi-random sampling, we organize the images in the image dataset $V$ into clusters. From this viewpoint, each of the clusters $\omega$ is one of the strata in the sampling process.

In the following section we elaborate on our clustering scheme. For purposes of this section, we assume that the dataset is organized into clusters $w$ and examine the random sampling case. Suppose that we aim at recovering the $k$ best matches to the query image provided by the user, the probability of drawing the best match $x^* \in V$ to the query image $y$ in a single trial from the cluster-set $\Omega$ is given by

$$P(x^*|\omega, \Omega) = P(x^*|\omega)P(\omega|\Omega) = \frac{1}{|\omega|}\frac{1}{|\Omega|} \qquad (1)$$

where $P(x^*|\omega) = \frac{1}{|\omega|}$ is the probability of randomly drawing the image $x^* \in w$ and $P(\omega|\Omega) = \frac{1}{|\Omega|}$ is the probability of $w$ given the cluster-set $\Omega$.

Now, consider the case where we draw $N_w$ images from the cluster $\omega$. The probability then becomes

$$P(x^*|\phi_w, \omega, \Omega) = P(x^*|\phi_w, \omega)P(\omega|\Omega) = \frac{N_w}{|\omega|}\frac{1}{|\Omega|} \qquad (2)$$

where $\phi_w$ are the images that arose from $w$. We define $\Psi = \bigcup_{w \in \Omega} \phi_w$ as the set of samples drawn from $V$. Thus, by randomly drawing $N_w$ images from every cluster $w$ in the dataset $V$, the probability of success, i.e. that the best match $x^*$ to the query image $y$ is in $\Psi$, becomes

$$P(x^*|\Psi, \Omega) = \Gamma = \sum_{w \in \Omega} \gamma_w \qquad (3)$$

where $P(x^*|\phi_w, \omega, \Omega) = \gamma_w$.

Note that, according to the relations above, the probability $P(x^*|\Psi, \Omega)$ tends to unity as number of clusters decrease and the number of samples per cluster increase. This implies that unity is reached when $\Psi = V$. In other words, the theoretical limit on the probability of recovering the best match $x^*$ to the query image is given by intensive search. This is an important observation since it provides an upper limit for the accuracy of any metric used for the purpose of assessing the similarity between the query image and those candidates in the dataset. The theoretical limit on the accuracy of the retrieval process is then given by the intensive search over the dataset $V$. In the following sections, we elaborate on the organization of the dataset into clusters and the proposed iterative random-sampling scheme.

## 3. Dataset Organization

In this section, we focus our attention in the assignment of images to the clusters in the database. Following [22], the dataset clustering problem is characterized by the set of images $V$ and a $|V| \times |V|$ matrix of pairwise affinities $A$. The element $A_{i,j}$ of the pairwise affinity matrix represents the degree of similarity between the images indexed $i$ and $j$. We will work with pairwise affinities which are constructed to fall in the interval $[0, 1]$. When the affinity is close to one, then there is a strong association between the image pair. If the affinity is close to zero then the association is weak. The aim in grouping is to partition the image-set $V$ into disjoint subsets. If $V_\omega$ represents one of these subsets and $\Omega$ is the index-set of different partitions (i.e. the different clusters), then $V = \bigcup_{\omega \in \Omega} V_\omega$ and $V_{\omega'} \cap V_{\omega''} = \emptyset$ if $\omega' \neq \omega''$.

To represent the assignment of images to clusters, we introduce a cluster membership indicator $s_{i\omega}$. This quantity measures the degree of affinity of the image indexed $i$ to the cluster $\omega \in \Omega$ and is in the interval $[0, 1]$. When the cluster membership is close to unity then there is a strong association of the image to the cluster.

### 3.1. Expected Log-likelihood Function

Our grouping process aims at estimating the cluster membership indicators $S$ based upon the pairwise affinity matrix $A$. We pose the problem in terms of the conditional likelihood $P(S|A)$. Since we are interested in the joint dependence of the affinities $A_{ij}$ and the cluster membership indicators $S$, we turn our attention instead to the maximization of the log-likelihood function for the observed pattern of pairwise affinities. Further, since we assume that the affinities corresponding to each cluster are independent of one another, we can write

$$\mathcal{L}(A, S) = \sum_{\omega \in \Omega} \sum_{(i,j) \in V} \ln p(s_{i\omega}, s_{j\omega}|A_{i,j}) \qquad (4)$$

To proceed, we require a model of the probability distribution for the pairwise affinities. Here we adopt a model in which the observed dataset cluster structure arises through a Bernoulli distribution. The parameter of this distribution is the affinity $A_{i,j}$. The idea behind this model is that any pair of images indexed $i$ and $j$ may belong to the same cluster. This affinity is treated as a Bernoulli variable. To test for cluster-consistency we make use of the quantity $s_{i\omega}s_{j\omega}$. This is unity if both images belong to the same cluster and is zero otherwise. The Bernoulli distribution becomes

$$p(A_{i,j}|\omega) = A_{i,j}^{s_{i\omega}s_{j\omega}}(1-A_{i,j})^{1-s_{i\omega}s_{j\omega}} \tag{5}$$

This distribution takes on its largest values when either the pairwise affinity $A_{i,j}$ is unity and $s_{i\omega} = s_{j\omega} = 1$, or if the affinity $A_{i,j} = 0$ and $s_{i\omega} = s_{j\omega} = 0$.

Here, we locate the maximum likelihood estimates of the cluster memberships using the apparatus of the EM algorithm [9]. The reason for doing this is that the cluster-membership variables $s_{i\omega}$ can be regarded as latent variables whose distribution is governed by the affinities $A_{i,j}$. Therefore, we use the EM algorithm to estimate them.

For the likelihood function above, the expected log-likelihood function is given by

$$Q(A^{(n+1)}|A^{(n)}) = \sum_{\omega\in\Omega}\sum_{(i,j)\in V} P(w|A_{i,j}^{(n)}) \ln p(A_{i,j}^{(n+1)}|\omega) \tag{6}$$

where $p(A_{i,j}^{(n+1)}|\omega)$ is the probability distribution for the pairwise affinities at iteration $n+1$ of the EM algorithm and $P(w|A_{i,j}^{(n)})$ is the a posteriori probability that the pair of images with affinity $A_{i,j}$ belong to the cluster indexed $\omega$ at iteration $n$. When the probability distribution function from Equation (5) is substituted, and after some algebra, the expected log-likelihood function becomes

$$Q(A^{(n+1)}|A^{(n)}) = \sum_{\omega\in\Omega}\sum_{(i,j)\in V} \zeta_{i,j,\omega}^{(n)} \Bigg\{ s_{i\omega}^{(n+1)} s_{j\omega}^{(n+1)} \ln \frac{A_{i,j}}{1-A_{i,j}^{(n+1)}} + \ln(1-A_{i,j}) \Bigg\} \tag{7}$$

where we have used the shorthand $\zeta_{i,j,\omega}^{(n)} = P(w|A_{i,j}^{(n)})$ for the a posteriori cluster membership probabilities.

### 3.1.1 Maximization

To compute the cluster membership variables, we compute the derivative of the expected log-likelihood function

$$\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial s_{i\omega}^{(n+1)}} = \sum_{j\in V} \zeta_{i,j,\omega}^{(n)} s_{j\omega}^{(n+1)} \ln \frac{A_{i,j}}{1-A_{i,j}} \tag{8}$$

However, the associated equations are not tractable in closed form. Instead, we use the soft-assign ansatz of Bridle [3] to update the cluster membership assignment variables. This involves exponentiating the partial derivatives of the expected log-likelihood function in the following manner

$$s_{i\omega}^{(n+1)} = \frac{\exp\left[\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial s_{i\omega}^{(n+1)}}\right]}{\sum_{\omega\in\Omega}\exp\left[\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial s_{i\omega}^{(n+1)}}\right]} \tag{9}$$

As a result the update equation for the cluster membership indicator variables is

$$s_{i\omega}^{(n+1)} = \frac{\exp\left[\sum_{j\in V}\zeta_{i,j,\omega}^{(n)} s_{j\omega}^{(n)} \ln\frac{A_{i,j}}{1-A_{i,j}}\right]}{\sum_{\omega\in\Omega}\exp\left[\sum_{j\in V}\zeta_{i,j,\omega}^{(n)} s_{j\omega}^{(n)} \ln\frac{A_{i,j}}{1-A_{i,j}}\right]} \tag{10}$$

### 3.1.2 Expectation

The a posteriori probabilities are updated in the expectation step of the algorithm. The current estimates of the cluster memberships $s_{i\omega}^{(n)}$ are used to compute the probability densities $p(A_{i,j}^{(n)}|\omega)$ and the a posteriori probabilities are updated using the formula

$$P(\omega|A_{i,j}^{(n)}) = \frac{p(A_{i,j}^{(n)}|\omega)\alpha^{(n)}(\omega)}{\sum_{\omega\in\Omega} p(A_{i,j}^{(n)}|\omega)\alpha^{(n)}(\omega)} \tag{11}$$

where $\alpha^{(n)}(\omega)$ is the available estimate of the class-prior $P(\omega)$. This is computed using the formula

$$\alpha^{(n)}(\omega) = \frac{1}{|V|^2}\sum_{(i,j)\in V} P(\omega|A_{i,j}^{(n)}) \tag{12}$$

Upon substituting for the probability density from Equation (6), the updated a posteriori probabilities are given by

$$P(\omega|A_{i,j}^{(n+1)}) = \frac{A_{i,j}{}^{s_{i\omega}^{(n)} s_{j\omega}^{(n)}}(1-A_{i,j})^{1-s_{i\omega}^{(n)} s_{j\omega}^{(n)}}\alpha^{(n)}(\omega)}{\sum_{(i,j)\in\Phi} A_{i,j}{}^{s_{i\omega}^{(n)} s_{j\omega}^{(n)}}(1-A_{i,j})^{1-s_{i\omega}^{(n)} s_{j\omega}^{(n)}}\alpha^{(n)}(\omega)} \tag{13}$$

where $P(\omega|A_{i,j}^{(n+1)}) = \zeta_{i,j,\omega}^{(n+1)}$.

## 3.2. Cluster Assignment

With the cluster membership variables at hand, the assignment of the images to the clusters in the dataset is implemented as follows. We commence by noting that, in the ideal case, the cluster membership variables $s_{iw}$ are equal to unity if the image indexed $i$ belongs to the cluster $w$ and zero otherwise. Further, the cluster assignments can be viewed as a matrix which slots over the cluster membership variable indexes and selects its largest values.

Moreover, we note that at the commencement of the sampling process it is desirable that the probability $P(x^*|\omega, \Omega)$ be uniform across the cluster-set $\Omega$. The reason is that we begin with a random uniform sampling over $V$ at the start-up of the retrieval process. As a result, and following Equation 1, we organize the dataset such that $\varrho =| \omega |=| \Omega |$.

Following are the steps to recover the cluster assignments. Let the matrix assignment matrix be denoted $\mathbf{L}$. We commence by clearing $\mathbf{L}$ and, recursively, do

1.- $\mathbf{L}(j, \tau) = 1$, where $\{j, \tau \mid s_{j\tau} = \max_{i \in V, w \in \Omega}(s_{iw})\}$.

2.- $s_{iw} = 0 \, \forall \{i = j\}$.

3.- $s_{iw} = 0 \, \forall \{| w |= \varrho\}$.

4.- $s_{iw} = 0 \, \forall \{w \notin \Omega\}$, where $\{| \Omega |= \varrho\}$.

until $S \equiv 0$. The image indexed $i$ is then a member of the cluster $w$ *if and only if* $\mathbf{L}(i, w) = 1$.

At this point, it is worth noting that the step sequence above slots over $S$ selecting the cluster memberships $s_{iw}$ such that the image whose cluster membership is maximum is assigned to the corresponding cluster $w$ subject to the size constraint given by $\varrho$. Since the membership variables for the images in $w$ are maximum across the dataset, the resulting clusters are as compact as possible. This, in turn, permits stratified random sampling to be effected.

## 4. Image Retrieval

For purposes of image retrieval, we adopt an iterative sampling scheme. To this end, we note that the retrieval process can be viewed as a quasi-random sampling one without substitutions. That is, at every iteration, a number of images are drawn from the dataset. Since we aim at retrieving the best $k$ matches to the query image $y$ on the dataset, the sampled images can be removed from further consideration. We, therefore, separate the problem into two parts. First, we aim at controlling the number of images sampled randomly from each cluster at every iteration of the retrieval algorithm. Second, we link the number of sampled images to a confidence measure relating the algorithm to the hard limit on accuracy given by intensive search.

Given a query image $y$, let the overall number of sampled images at iteration $n$ be $N = \sum_{\omega \in \Omega} N_w^{(n)}$, where $N_w^{(n)}$ is the number of sampled images per cluster $w$ indexed to the iteration number. Moreover, consider the probability of $P(w \mid y, \Omega)$ be equivalent to $\frac{N_w^{(n)}}{N}$. Note that this is consistent with the uniform random sampling case in which

$$P(w \mid y, \Omega) = \frac{1}{| \Omega |} = \frac{N_w^{(n)}}{N} \tag{14}$$

This observation is important since it opens up the possibility of using kernel methods to recover the value of $P(w \mid y, \Omega)$ from sampled images.

For purposes of computing $P(\omega \mid y, \Omega)$, we consider the class of kernels given by

$$\mathcal{K}(w, y) = P(w \mid y, \Omega) = \sum_{x \in w} P(x \mid y) P(y \mid \Omega) P(\Omega) \tag{15}$$

We can expand these kernels by taking sums over products of weighted probability distributions [2]. In this manner, the kernel $\mathcal{K}(x, y)$ can be viewed as a function which is proportional to a mixture distribution of the form $P(w \mid y, \Omega) = \sum_{x \in w} \pi_x P(x \mid y)$, where $\pi_w$ is the mixture weight given by $P(y \mid \Omega)P(\Omega)$. Thus, the marginal for the distribution above with respect to the image cluster $w$ is

$$P(w, y \mid \Omega) = \frac{\sum_{x \in w} \pi_x P(x \mid y)}{\sum_{\Omega} \sum_{x \in w} \pi_x P(x \mid y)} \tag{16}$$

The equation above can be rewritten making use of the proportionality between $\mathcal{K}(w, y)$ and $P(w \mid y, \Omega)$, as follows

$$P(w \mid y, \Omega) = \frac{\mathcal{K}(w, y)}{\sum_{\Omega} \mathcal{K}(w, y)} \tag{17}$$

Due to the use of a maximum likelihood approach in the previous section, the dataset is comprised by compact clusters. Hence, the kernel $\mathcal{K}(w, y)$ can be approximated as a function of the average distances of the query image to the sampled images at the current iteration, i.e.

$$\mathcal{K}(w, y) \approx \mathscr{K}(w, y)^{(n)} = \frac{1}{N_w} \sum_{x \in \mathcal{X}_w} f_{\kappa^{(n)}}\big(d(x, y)\big) \tag{18}$$

where $\mathcal{X}_w$ is the set of images drawn from the cluster $w$ at the current iteration, $d(x, y)$ is the similarity between the images $x$ and $y$, $\kappa^{(n)}$ is the bandwidth parameter of the kernel at iteration $n$ and $\mathscr{K}(w, y)^{(n)}$ is the approximation of $\mathcal{K}(w, y)$ indexed to iteration number.

In the following section, we will give further discussion on the bandwidth parameter $\kappa^{(n)}$ and its effect in the sampling process. For now, we continue our analysis and note that, by using the quantity $\mathscr{K}(w, y)^{(n)}$ as an alternative to $\mathcal{K}(w, y)$ so as to measure the overall similarity of the query image to the images drawn from the cluster $w$ we can control the number of images sampled from each cluster at every iteration of the algorithm. By substituting the equation above into Equation 14 and manipulating terms, we get

$$N_w^{(n)} = N \frac{\mathscr{K}(w, y)^{(n)}}{\sum_{\Omega} \mathscr{K}(w, y)^{(n)}} = N \frac{\sum_{x \in \mathcal{X}_w} f_{\kappa^{(n)}}\big(d(x, y)\big)}{\sum_{\Omega} \sum_{x \in \mathcal{X}_w} f_{\kappa^{(n)}}\big(d(x, y)\big)} \tag{19}$$

At each iteration $n$ of our quasi-random sampling scheme, the number of samples per cluster is governed by $\mathscr{K}(w, y)^{(n)}$, whereas the $N_w^{(n)}$ samples per $w$ are selected randomly from the remaining images in the dataset, i.e. those that have not been drawn at previous iterations. Since the number of remaining images per cluster to be drawn decreases with respect to iteration number, the probability $P(x^* \mid \phi_w, w, \Omega)$ is no longer constant at every iteration.

Now we turn our attention to the computation of the confidence in recovering the best match $x^*$ to the query image $y$. To this end, we recover the probability $P(x^* \mid \Phi, \Omega)$ by indexing $P(x^* \mid \phi_w, w, \Omega) = \gamma_w$ to iteration number. Following Equations 2 and 3, we can write

$$\Gamma^{(n)} = \frac{1}{|\Omega^{(n)}|} \sum_{\Omega^{(n)}} \frac{N_w^{(n)}}{|\omega^{(n)}|} \tag{20}$$

where $\Omega^{(n)}$ is the set of the non-empty clusters $\omega^{(n)}$ at iteration $n$. This is an important observation since it provides not only a criterion to stop the iterative search based upon the confidence in recovering the best match, but also a means to compute the lower bound for the expected accuracy of the search procedure.

Recall that we have organized the dataset using a maximum likelihood approach so as to recover compact clusters $\omega$. Thus, the quasi-random search above is expected to perform better than random search. As a result, given the accuracy or performance of intensive search $\vartheta$, the lower bound on the search accuracy is given by the expression for random search, i.e.

$$\vartheta^{(n)} = \vartheta\Gamma^{(n)} = \vartheta \frac{1}{|\Omega^{(n)}|} \sum_{\Omega^{(n)}} \frac{N_w^{(n)}}{|\omega^{(n)}|} \tag{21}$$

In other words, the accuracy of the retrieval operation tends to that yielded by intensive search as the confidence on the retrieval, i.e. the probability of recovering the best match $x^*$ to the query image $y$, tends to unity.

## 5. Discussion and Implementation Issues

Having described the theoretical foundation of the algorithm above, we now turn to its step sequence. We organize the dataset $V$ off-line using a measure of similarity or metric $d(x, y)$. There are a number of metrics that have been proposed elsewhere in the literature. In our experiments, we follow Nilsback and Zisserman [18] and use SIFT descriptors [16], color and MR-filters [25]. The feature vocabulary is computed using the method presented in [18], where each vector is quantized to recover visual words and optimized as described in [7]. As a result, the intensive search performance $\vartheta$ for our experiments is given by a nearest-neighbor classifier, where the distance $d(x, y)$ is given by the Euclidean metric between the frequency histograms for the visual words corresponding to $x$ and $y$. It is important to stress that other distance measures, code books and image descriptors can be used for purposes of recovering $d(x, y)$. Moreover, pairwise distances between the query and the dataset images can be based upon string and tree-kernels [15], edit distance [14], etc.

In Section 3, we cast the dataset organization in terms of a maximum likelihood formulation governed by the image pairwise affinities. In our experiments, we compute the corresponding entries in the affinity matrix using the expression $A_{i,j} = \exp[-\ell d(x, y)^2]$, where $\ell$ is a constant. This is an extremely simple approach that can be substituted without any loss of generality on the search algorithm by other methods which yield an affinity measure bounded between zero and unity. We have followed [18] and used a nearest-neighbor classifier.

Also, recall that in the previous section we introduced the kernel $\mathcal{K}(w, y)^{(n)}$ governed by the function $f_\kappa(\cdot)$. Following our choice of $A_{i,j}$, we used a kernel

$$\mathcal{K}(x_i, y)^{(n)} = \exp\left(-\frac{d(x, y)^2}{\kappa^{(n)}}\right) \tag{22}$$

where $d(x, y)$ is the dissimilarity between the image $x$ in the dataset and the query image $y$.

In the equation above, we set $\kappa^{(n)} = \tau\sigma^{(n)}$, where $\tau$ is a constant and $\sigma^{(n)}$ is the variance of the distances corresponding to the images sampled at the current iteration. In this way, the bandwidth of the kernel varies in accordance with the distances between the images drawn from the sample set and the query. Note that, if $\kappa^{(n)}$ is very large or very small, the samples per cluster as given in Equation 19 become uniform across $\omega$. It means that the sampling process becomes random. Thus, we can interpret $\kappa^{(n)}$ as a variable that governs the tradeoff between the cost of intensive search and the accuracy of the classifier. That is, if $\kappa^{(n)} \to 0$ or $\kappa^{(n)} \to \infty$, the results yielded by the classifier are ignored by the algorithm. If $\kappa^{(n)} = 1$ then the behavior of the algorithm is governed solely by the dissimilarities $d(x, y)$. Because the $\sigma$ in each iteration is automatically computed, the value of $\kappa^{(n)}$ is decided by $\tau$, which is a decisive control factor.

With these ingredients and once the dataset is organized as described in Section 3, the step sequence of the algorithm is as follows:

1.- Randomly select $N_w$ for each cluster $\omega \in \Omega$ as described in the previous section making use of the probability $P(\omega \mid y, \Omega)$.

2.- Compute the confidence $\Gamma^{(n)}$.

3.- Update the list of $K$ best matches to the query image $y$ from the sampled images.

4.- If the desired confidence $\Gamma^*$ has been reached or every image has been sampled then exit.

5.- From the sampled images, compute $\mathcal{K}(w, \Omega)^{(n)}$ using Equation 18 so as to recover $P(\omega \mid y, \Omega)$. Discard the sampled images and go to Step 1.

|  |  | 1-NN | 2-NN | 3-NN | 4-NN | 5-NN | Sampling percentage |
|---|---|---|---|---|---|---|---|
| intensive search | shape | 46.8% | 57.6% | 63.5% | 68.5% | 73.5% | 100% |
|  | color | 32.4% | 47.9% | 55.0% | 60.3% | 65.9% |  |
|  | texture | 21.1% | 31.8% | 41.5% | 49.7% | 57.1% |  |
| quasi-random sampling | shape | 39.7% | 49.5% | 56.1% | 60.7% | 64.0% | 25% |
|  | color | 27.7% | 36.5% | 43.0% | 47.2% | 50.3% |  |
|  | texture | 18.0% | 29.1% | 37.1% | 43.7% | 48.5% |  |

Table 1. Performance comparison between intensive search and quasi-random sampling on the Oxford flowers dataset. For the quasi-random sampling, the confidence is set to 99% and $\tau = 2$.

Figure 1. Sample images for the Oxford flowers dataset. On the left side of the figure is a query image. The right side contains 12 images in the training set, one of which is the ideal match to the query.

## 6. Experiments

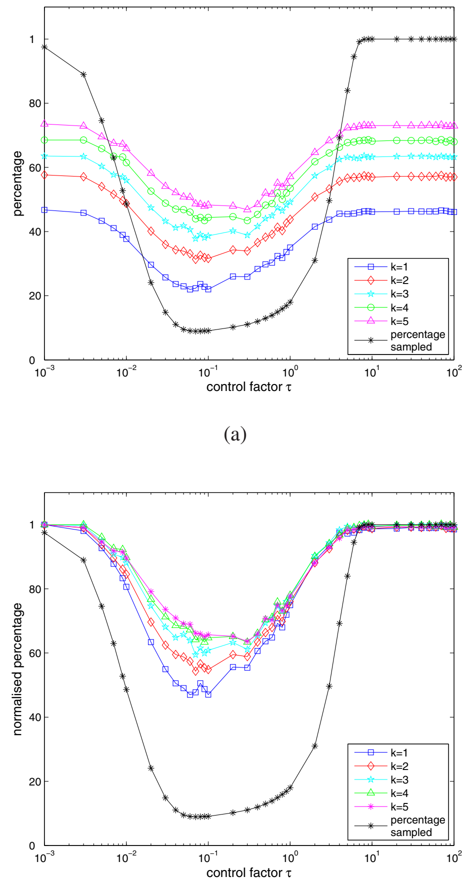
In this section, we present results on the Oxford flowers dataset [18] and discuss the influence of the parameter $\tau$ on the image retrieval behavior. In our experiments, we store, at each iteration, a list of the top $k$ matching results. This is equivalent to a k-nearest neighbor (k-NN) classifier on the sampled images. We aim at using the recognition rate for the nearest neighbor (NN) classifier to evaluate the retrieval accuracy as a function of both confidence and sampling percentage with respect to the whole of the dataset. Thus, we can define the likelihood of the best match being in the top $k$ nearest neighbors sampled up to iteration number.

The Oxford flowers dataset contains unsegmented flower images in 17 species against difference background, some clean and some cluttered. Images in the dataset are divided into a training set of 680 images, i.e. 40 images for each species, a validation and a testing set of 340 images each (20 images per species, respectively). In our experiments, we used the training set as the target dataset and used images in the testing set as the query images. All our quasi-random sampling results are averaged over 10 experiments. Example images on the dataset are shown in Figure 1.

Following [18], we have implemented three NN classifiers using codebooks of visual words comprised by shape, color, and texture features. To recover a baseline accuracy for each of the codebooks, we performed an intensive search over all the images in the training set and recover the best $k$ matches ranked upon their pairwise distances. The recognition rates of the k-NN classifiers are displayed in Table 1. Note that image classification results reported in [18] and [29] using NN and SVMs classifiers are better than those shown in the table. The reason is that segmented flower images are used for their experiments. Note that the SVM-based method is aimed at image classification, which is only one step towards image retrieval. If we define the target as the image(s) that is (are) closest to the query image, a within-class search step is needed after classification. The difference also lies in that the SVM is a supervised approach, whereas ours is unsupervised and, thus, requires much less human involvement.

We applied our quasi-random sampling method to the dataset organized as described in Section 3 using pairwise EM clustering. The clustering created 27 image clusters with nearly the same number of images in each cluster. Using quasi-random sampling, we can achieve a retrieval accuracy close to that yielded by intensive search with far less images sampled per query. For instance, when shape features are used we can achieve a confidence of 99% by a sampling percentage, i.e. the percentage of the dataset sampled in the query operation, of 25% and an accuracy of 39.7% ($k$-NN with $k = 1$). Results for intensive search and our method with $k = \{1, 2, 3, 4, 5\}$ with respect to the three features under study are shown in Table 1.

As mentioned earlier, the parameter $\kappa^{(n)}$ in Equation 22 governs the trade-off between the number of images sampled from the dataset and the accuracy of the classifier. The confidence $\Gamma^{(n)}$ in equation 20 and the sampling rate can be then viewed as functions of $\kappa^{(n)}$. Moreover, recall that the constant $\tau$ does not depend on iteration number, therefore, we can adjust it to favor the classifier over to the randomized search or viceversa. In Figure 2 we show plots for a confidence of 99% corresponding to the fraction of the dataset sampled, i.e. sampling percentage, and the retrieval accuracy for each $k$-NN classifier as a function of $\tau$. We do this in order to illustrate the effect of $\kappa$ in the retrieval process. In figure 2(a), when $\tau$ tends to 0.001 or grows beyond 7, the image retrieval process is close to intensive search. It requires almost all images to be sampled and achieves an



(a)

(b)

Figure 2. Absolute and relative accuracies and sampling percentage as a function of $\tau$. Top panel: Absolute recognition and sampling percentage; Bottom panel: Accuracy and sampling percentage normalized with respect to the upper bounds in Table 1.

accuracy close to the upper bound. When the control factor is between these two values, less sampling is required given a fixed confidence level. For example, when $\tau = 1$, we are $99\%$ confident that the accuracy is $35\%$ for the best image retrieved being the true match having sampled $18\%$ of the images in the dataset. Moreover, note that the plots are quite "flat" in the range $\tau \in [0.1, 2]$. Thus, the sampling method here is quite stable to variations in $\tau$. The proposed method is more efficient on very large databases. As shown in equation 20, the sampling percentage is controlled by the confidence value, which is closely related to the number of clusters and the number of images in each cluster. This guarantees that, the larger the database, the more efficient the method becomes.

Figure 2(a) also suggests a means to select $\tau$ based upon the relationship between the accuracy and the percentage of the dataset sampled for a given confidence value. This is based on the observation that, for some values of $\tau$, the gain in accuracy becomes negligible with respect to the cost of sampling a larger amount of the images in the dataset. To illustrate this, in Figure 2(b), we display the plots in Figure 2(a) normalized with respect to their upper bounds. Thus, in the plots, a sampling percentage close to $100\%$ implies that almost the whole of the dataset needs to be sampled to achieve a confidence of $99\%$. The largest differences between the curves corresponding to the normalized accuracy and sampling percentage is obtained when $\tau = 2$, as shown in Figure 3. As a result, the value of $\tau$ for which the accuracy is best given a small sampling percentage for a given confidence in our dataset is $\tau = 2$.

Finally, we would like to illustrate how the organization of dataset influences the efficiency of the image retrieval. Figure 4 shows the accuracy with respect to sampling percentage when different features are used to cluster the dataset. Note that, for a random sampling case, these plots are expected to be linear with respect to the percentage of dataset images sampled for purposes of retrieval. The plots show that, far from a linear tendency, the shape feature provides an accuracy of almost $40\%$ for $k = 1$ with a sampling percentage of $25\%$. Moreover, with $15\%$ sampled, the accuracy is close to $35\%$. Thus, our dataset organization scheme improves efficiency by reducing the sampling percentage required to achieve retrieval performances comparable to random search.

## 7. Conclusions

In this paper, we have presented a method for image retrieval that employs a maximum likelihood method to organize the dataset and a quasi-random stratified sampling for the query operation. The method is quite general in nature and allows the use of a variety of metrics between image pairs and descriptors. It also provides a measure of confidence on the retrieval process and bounds on accuracy as a function of the results yielded by intensive search. We have shown experiments to illustrate the utility of the algorithm.

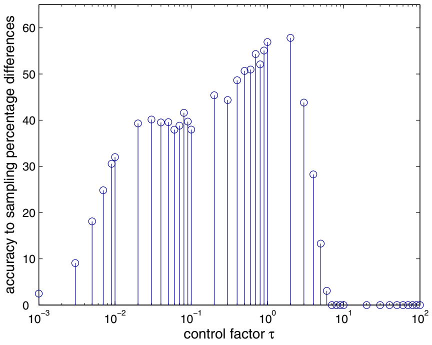


Figure 3. Differences between relative recognition rate and sampling rate.

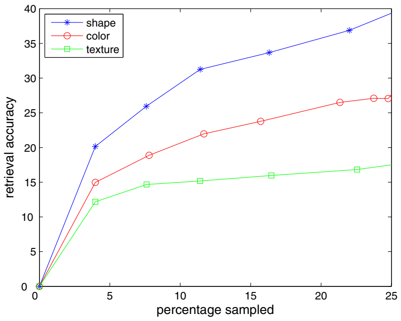


Figure 4. Influence to efficiency when organizing dataset using different features. The confidence level is set to 95% and $\tau = 2$.

## References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999. 1

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 4

[3] J. S. Bridle. Training stochastic model recognition algorithms can lead to maximum mutual information estimation of parameters. In *NIPS 2*, pages 211–217, 1990. 3

[4] Y. Chen, J. Z. Wang, and R. Krovetz. Clue: Cluster-based retrieval of images by unsupervised learning. *IEEE Trans. on Image Processing*, 14(8):1187–1201, 2005. 1

[5] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Int. Conference on Computer Vision*, 2007. 1

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000. 1

[7] N. Dalai and B. Triggs. Histogram of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, pages I:886 – 893, 2005. 5

[8] M. Das, R. Manmatha, and E. Riseman. Indexing flower patent images using domain knowledge. *IEEE Transactions on Intelligent Systems and Their Applications*, pages 1094–7167, 1999. 1

[9] A. Dempster, N. Laird, and D. Rubin. Maximum-likehood from incomplete data via the EM algorithm. *J. Royal Statistical Soc. Ser. B (methodological)*, 39:1–38, 1977. 3

[10] R. O. Duda and P. E. Hart. *Pattern Classification*. Wiley, 2000. 1

[11] S. A. M. Farzin and J. Kittler. Robust and efficient shape indexing through curvature scale space. In *Proceedings of the 7th British Machine Vision Conference*, volume 1, pages 53–62, 1996. 1

[12] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Comp. Vision and Pattern Recognition*, pages II:524–531, 2005. 1

[13] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. On recognition of 3d objects from 2d images. In *IEEE International Conference of Robotics and Automation*, 1988. 1

[14] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl.*, 6:707–710, 1966. 5

[15] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and W. C. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002. 5

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 5

[17] W. Niblack et al. The qbic project: Querying images by content using color, texture and shape. In *Proc. SPIE Conference on Storage and Retrieval of Image and Video Databases*, 1908, pages 173–187, 1993. 1

[18] M. E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Comp. Vision and Pattern Recognition*, pages II:1447–1454, 2006. 1, 5, 6

[19] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Comp. Vision and Pattern Recognition*, pages II:2161–2168, 2006. 1

[20] R. W. Picard. Light-years from lena: Video and image libraries ind the furture. In *International Conference on Image Processing*, volume 1, pages 310–313, 1995. 1

[21] P. Quelhas, F. Monay, J. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. V. Gool. Modelling scenes with local descriptors and latent aspects. In *Int. Conference on Computer Vision*, pages I:883–890, 2005. 1

[22] A. Robles-Kelly and E. R. Hancock. A probabilistic spectral framework for spectral clustering and grouping. *Pattern Recognition*, 37(7):1387–1405, 2004. 2

[23] K. Sengupta and K. L. Boyer. Using geometric hashing with information theoretic clustering for fast recognition from a large cad modelbase. In *IEEE International Symposium on Computer Vision*, pages 151–156, 1995. 1

[24] A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, and S. W. Zucker. Indexing using a spectral encoding of topological structure. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 491–497, 1998. 1

[25] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *Int. Conf. on Comp. Vision*, pages I:370 – 377, 2005. 5

[26] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Int. Conference on Computer Vision*, pages II:1470–1477, 2003. 1

[27] Z. Su, H. Zhang, S. Li, and S. Ma. Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning. *IEEE Trans. on Image Processing*, 12(8), 2003. 1

[28] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. on Pattern Anal. and Machine Intelligence*, 28(7):1088–1099, 2006. 1

[29] M. Varma and D. Ray. Learning the discriminative powerinvariance trade-off. In *Int. Conference on Computer Vision*, 2007. 6

[30] N. Vasconcelos. On the efficient evaluation of probabilistic similarity functions for image retrieval. *IEEE Trans. on Informaiton Theory*, 50(7):1482–1496, 2004. 2