Issues Concerning Dimensionality and Similarity Search

Jelena Tešić, Sitaram Bhagavathy, and B. S. Manjunath
Electrical and Computer Engineering Department
University of California, Santa Barbara, CA 93106-9560
{*jelena, sitaram, manj*}*@ece.ucsb.edu*

Abstract

Effectiveness and efficiency are two important concerns in using multimedia descriptors to search and access database items. Both are affected by the dimensionality of the descriptors. While higher dimensionality generally increases effectiveness, it drastically reduces efficiency of storage and searching. With regard to effectiveness, relevance feedback is known to be a useful tool to squeeze information from a descriptor. However, not much has been done toward enabling relevance feedback computation using high-dimensional descriptors over a large multimedia dataset. In this context, we have developed new methods that enable us to a) reduce the dimensionality of Gabor texture descriptors without losing on effectiveness, and b) perform fast nearest neighbor search based on the information available during each iteration of a relevance feedback step. Experimental results are presented on real datasets.

1. Introduction

In content based retrieval, the main task is to find entries in a multimedia database that are most similar to a given query object. The volume of the data is typically huge, and the feature vectors are of high dimensionality. Therefore, it is in general impractical to store all the extracted feature vectors in the main memory. Since I/O operations for storage devices are slow, the time spent accessing the feature vectors overwhelmingly dominates the time complexity of the search. In databases, indexing is used to narrow the scope of the search and increase its efficiency. There has been a considerable amount of work on high-dimensional indexing [1]. The search procedure is subject to the notorious "curse of dimensionality", i.e., the search space grows exponentially with the number of dimensions.

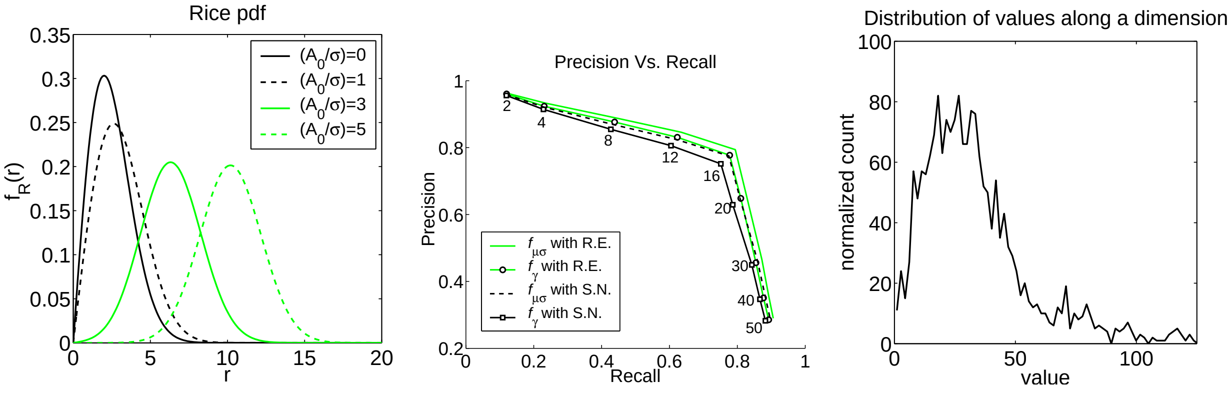
An equally important issue is the effectiveness of the multimedia descriptors. This has been addressed by the content based retrieval community extensively [2]. While the low-level features are quite effective in "similarity" retrieval, they do not capture well the high level semantics. Therefore, new concepts were introduced to capture image information: new similarity metrics, learning similarity metrics, and on-line learning systems that use relevance feedback mechanisms to modify the query and/or the similarity computations. A practical system should address concerns about both efficiency and effectiveness. However, not much has been done toward enabling, for example, relevance feedback computations using high-dimensional index structures. In this context, we have recently developed new methods that can support the system implementation.

A widely used texture descriptor, which is part of the MPEG-7 standard [2], is that computed using multiscale Gabor filters. However, the high dimensionality and computational complexity of this descriptor adversely affect the performance as well as the storage, computation, and indexing requirements of a content-based retrieval system. We developed a modified texture descriptor that has comparable performance, but nearly half the dimensionality of the MPEG-7 descriptor. An adaptive nearest neighbor computation method that makes use of the information available during each iteration of a relevance feedback step, is proposed to speed up the search. The adaptive nearest neighbor computation enables retrieval systems to scale to a large number (few hundred thousands and higher) of items without compromising on search effectiveness, and allows data mining applications.

2. Dimensionality Reduction of MPEG-7 Texture Features

An image can be considered to be a mosaic of textures and texture features associated with the regions can be used to index the image data for searching and browsing. A Gabor-based homogeneous texture descriptor [3] has been adopted by the MPEG-7 standard for its effectiveness and efficiency. Gabor filters can be considered to be orientation and scale tunable edge and line detectors, and the statistics of these micro-features can be used to characterize the un-

This research was supported in part by the following grants and awards: ISCR-LLNL #W-7405-ENG-48, NSF-DLI #IIS-49817432, and ONR #N00014-01-1-0391.



(a) (b) (c)

Figure 1: (a) Rice pdf; (b) Precision vs. Recall curves over the Brodatz dataset; (c) Histogram of the values along arbitrary dimension of $f_{\mu\sigma}$.

derlying texture. The texture descriptor for s scales and k orientations is given by

$$f_{\mu\sigma} = [\mu_{00},\ \sigma_{00}, ...,\ \mu_{s-1,k-1},\ \sigma_{s-1,k-1},\ \mu_I,\ \sigma_I] \quad (1)$$

where μ_{mn} and σ_{mn} are the mean and standard deviation of the Gabor filter outputs (taking absolute values, by default) at scale m and orientation n. For an input texture image I, μ_I and σ_I are the mean and standard deviation of the pixel intensities of the image. Note that the dimensionality of $f_{\mu\sigma}$ is $2sk + 2$. The format of (1) for the texture descriptor is driven by the implicit assumption that the filter outputs have Gaussian-like distributions.

It is shown that the Gabor filter outputs have a Rice distribution [4], given by $f_R(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2+A_0^2}{2\sigma^2}\right) I_0\left(\frac{A_0 r}{\sigma^2}\right)$, where A_0 and σ are the parameters and $I_0(x)$ is the zero-order modified Bessel function of the first kind. The Rice pdf can vary from a Rayleigh pdf for small A_0 ($A_0 \approx 0$) to an approximate Gaussian pdf for large A_0 ($A_0 \gg \sigma$), as shown in Figure 1(a). The latter case occurs when the texture is well-defined and periodic, with a highly peaked frequency component at ω_0. The filter outputs tend to have a Rayleigh pdf when the frequency components of the texture are weak in the vicinity of ω_0. The filter bank used for computing the texture descriptor has predefined center frequencies. Over a wide range of textures, the probability that a given texture has a strong component at a specified center frequency, is small. Hence, we claim [5] that the Rayleigh pdf model for filter output distributions is valid with a higher probability than the Gaussian pdf model that inspires the descriptor in (1).

The Rayleigh pdf, $p(z) = \frac{z}{\gamma^2} \exp\left(-\frac{z^2}{2\gamma^2}\right)$ has only one parameter γ. Therefore, instead of (1), we propose the following texture descriptor with dimensionality $sk + 2$,

$$f_\gamma = [\gamma_{00}, \gamma_{01}, ..., \gamma_{s-1,k-1}, \mu_I, \sigma_I] \quad (2)$$

where γ_{mn} is the Rayleigh parameter of the Gabor filter output distribution. If z_i are corresponding filter outputs, and N is the number of output coefficients, the ML estimate of the Rayleigh parameter is given by: $\gamma_{mn}^2 = \frac{1}{2N} \sum_{i=1}^{N} |z_i|^2$. Based on the estimates of the Gaussian parameters in (1), $\mu_{mn} = \frac{1}{N} \sum_{i=1}^{N} |z_i|$ and $\sigma_{mn}^2 = \frac{1}{N} \sum_{i=1}^{N} |z_i|^2 - \mu_{mn}^2$, it is it is easy to show that $\gamma_{mn}^2 = \frac{1}{2}\left(\mu_{mn}^2 + \sigma_{mn}^2\right)$. Thus we can compute the new features from the original Gabor features without having to repeat the computationally expensive filtering step. Also, we can see that, for large values of N, f_γ needs 50% fewer additions in its computation than $f_{\mu\sigma}$.

2.1 Experiments

We compare the similarity retrieval performance of $f_{\mu\sigma}$ and f_γ on the Brodatz texture dataset. The dimensionality of $f_{\mu\sigma}$ is 62 and that of f_γ is 32. The dataset consists of 1856 images (16 from each of 116 texture classes). Let $A(q)$ denote the set of T retrievals (based on the smallest L_1 distances from texture q in the descriptor space) and $R(q)$, the set of images in the dataset relevant to q. The *precision* is defined by $P(q) = \frac{|A(q) \cap R(q)|}{|A(q)|}$, and the *recall* by $C(q) = \frac{|A(q) \cap R(q)|}{|R(q)|}$, where $|\cdot|$ denotes cardinality. Fig. 1(b) shows the precision vs. recall curves for each descriptor and normalization method. The curves are plotted by averaging precision and recall over all q, for different values of T (shown below the curves). While the dimensionality of f_γ is smaller by almost 50% (for large sk), the drop in precision (equivalently, the increase in error rate) is below 3% for a wide range of T. This supports our claim that the Rayleigh pdf assumption for filter outputs is valid with high probability when we consider a wide range of textures.

The distance metric used for measuring visual dissimilarity is more sensitive in dimensions with larger dynamic ranges. Since this is undesirable, the descriptors are normalized so that each dimension has the same dynamic range. Over a large dataset, the values along each dimension (other than μ_I and σ_I) of both $f_{\mu\sigma}$ and f_γ follow a skewed distribution that can be well modelled by a Rayleigh pdf (see Fig. 1(c)). *Rayleigh equalization (R.E.)* along the dimension d forces the distribution to have a uniform distribution $U(0,1)$, i.e. $X_d^{(n)} = 1 - \exp\left(-\frac{X_d^2}{2\gamma_d^2}\right)$, where $\gamma_d^2 = \frac{1}{2M}\sum_{i=1}^{M}|(f_\gamma)_d|^2$ if M is the number of features in a given dataset.

We now compare the indexing performance of *R.E.* with that of *standard normalization (S.N.)* (where each dimension is made to have zero mean and unit variance). We use an aerial image dataset with M=90,744 subimages of 128×128 pixels. f_γ is computed for each subimage using previously computed MPEG-7 features, and a standard VA file index [6] (with S bits per dimension) is constructed for the dataset. The indexing efficiency is quantified by the number of candidates (the smaller the better) obtained after VA filtering for K nearest neighbors. Table 1 shows these numbers (averaged over 20 queries) for *S.N.* and *R.E.*, for $K = 20$, and different values of S. Note that *R.E* results in fewer candidates, thus indicating a more efficient index structure.

Table 1: Number of candidates obtained after a 20-NN VA filtering (using VA-file index of f_γ, with S bits per dimension).

S	**8**	**7**	**6**	**5**	**4**
S.N.	74	189	863	3466	9176
R.E.	66	147	440	1460	3481

3. Adaptive Nearest Neighbor Search for Relevance Feedback

In content-based retrieval, to retrieve images that are similar in texture or color, usually one computes the nearest neighbors of the query feature vector in the corresponding feature space. Nearest neighbor computations over a large number of dataset items is expensive. This is further aggravated by the high dimensionality of image descriptors, and the need to perform this search repetitively in relevance feedback. This can be a limiting factor on the overall effectiveness of using relevance feedback for similarity retrieval. We address the problem of nearest neighbor search for relevance feedback applications. Our method takes advantage of the correlation between two sets of nearest neighbors from the consecutive iterations in pruning the search space.

3.1. Indexing High-dimensional Spaces

Basically, the purpose of the index is to narrow the scope of the search and hence avoid accessing irrelevant feature vectors. Recently, there has been much work on indexing structures to support high-dimensional feature spaces. However, as reported in [1], the effectiveness of many of these indexing structures is highly data-dependent and, in general, difficult to predict. Often a simple linear scan of all the items in the database is cheaper than using an index based search in high dimensions. An alternative to high-dimensional index structures is a compressed representation of the database items, like the vector approximation structure known as VA-File [6].

The construction of the approximation is based on the feature values in each of the dimensions independently. As a result, the compressed domain representation can support query search if the distance is quadratic. Secondly, the construction of approximation can be made adaptive to the dimensionality of data.

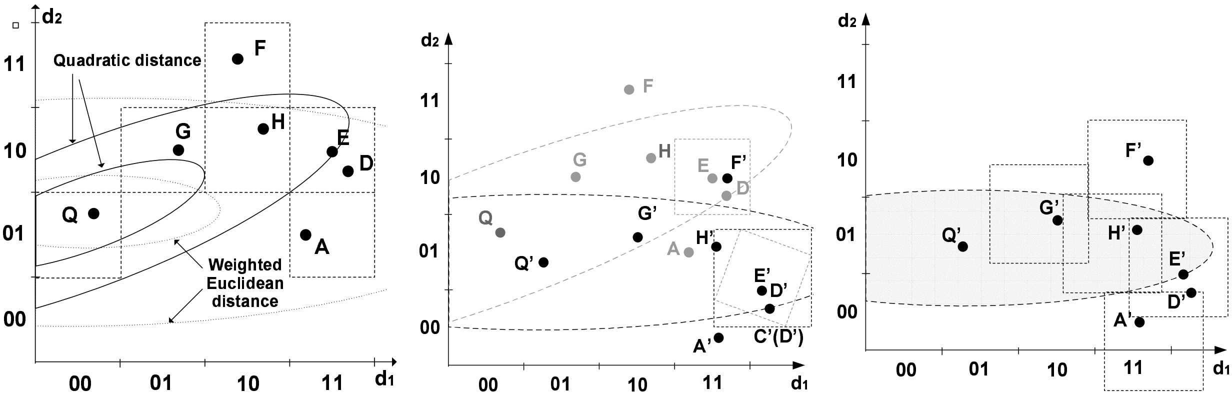
Consider a database $\Phi = \{F_i \mid i \in [1,N]\}$ of N elements, where F_i is an M-dimensional feature vector.

$$F_i = [f_{i1}, f_{i2}, \ldots, f_{iM}]^T \quad (3)$$

Each of the feature vector dimensions is partitioned into non overlapping segments. Generally, the number of segments is 2^{B_j}, $j \in [1,M]$. B_j is the number of bits allocated to dimension j. Denote the boundary points that determine the 2^{B_j} partitions to be b_{lj}, $l = 0,1,2,\ldots,(2^{B_j}-1)$. For a feature vector F_i, its approximation $C(F_i)$ is an index to the cell containing F_i. If F_i is in partition l along the j^{th} dimension, then: $b_{lj} \leq f_{ij} < b_{(l+1)j}$.

Approximation based nearest neighbor search can be considered to be a two-phase filtering process [6]. Phase I is an approximation-level filtering. In this phase, the set of all vector approximations is scanned sequentially and lower and upper bounds are computed on the distances of each object in the database to the query object. During the scan, a buffer is used to keep track of the K^{th} largest upper bound ρ found from the scanned approximations. If an approximation is encountered such that its lower bound is larger than the K^{th} largest upper bound found so far, the corresponding feature vector can be skipped since at least better candidates exist. Otherwise, the approximation will be selected as a candidate and its upper bound will be used to update the buffer. The resulting set of candidate objects at this stage is $N_1(Q,W)$. Phase II finds K nearest neighbors from the feature vectors contained by approximations filtered in Phase I. The feature vectors are visited in increasing order of their lower bounds and the exact distances to the query vector are computed. If a lower bound is reached that is larger than the K^{th} actual nearest neighbor distance encountered so far, there is no need to visit the remaining candidates. Finally, the nearest neighbors are found by sorting the distances.

In database searches, the disk/page access is an expensive process. The number of candidates from Phase I filtering determines the cost of disk/page access. Our focus is on improving Phase I filtering in the presence of relevance feedback.



(a) VA Construction (b) Rotational Mapping (c) Search Space

Figure 2: a) Construction of 2D VA-file approximations and bound computation for (1) weighted Euclidean and (2)quadratic distance; b) Rotational mapping of feature space: $A \rightarrow A'$:$A' = PA$; c) Illustration of using r^u_{t+1} to limit the search space in Phase I adaptive filtering.

3.2. Relevance Feedback

A way to compute the new set of retrievals closer to the user's expectation is to modify the similarity metric used in computing the distances between the query and the database items. If Q is a query vector, F_i is a database feature vector, and W_t is a symmetric, real, positive semi-definite matrix, the distance $d(Q, F_i, W)$ between two feature vectors is typically calculated as a quadratic function:

$$d^2(Q, F_i, W) = (Q - F_i)^T W (Q - F_i) \qquad (4)$$

During each iteration, the weight matrix is updated based on the user's feedback. Given the updated weight matrix, the next set of nearest neighbors is then computed. Let W_t be the weight matrix used in iteration t, and R_t be the set of K nearest neighbors to the query object Q, using (4) to compute the distances. At iteration t, define the k^{th} positive example vector ($k \in [1, K']$) as: $X_k^{(t)} = [x_{k1}^{(t)}, x_{k2}^{(t)}, \ldots, x_{kM}^{(t)}]^T$, $X_k^{(t)} \in R_t$. K' is the number of relevant objects identified by the user. These K' examples are used to modify the weight matrix W_t to W_{t+1}. We consider an optimized learning technique proposed in [7] that merges two existing well-known updating schemes MARS [8] and MindReader [9].

The adaptive nearest neighbor search problem can be formulated as follows: *given R_t, W_t, and K', the weight matrix W_{t+1} is derived from W_t using some update scheme. Compute the next set of K nearest neighbors R_{t+1} using W_{t+1} using minimum number of computations.*

Given a query Q and a feature vector F_i, the lower and upper bounds of $d(Q, F_i, W^t)$ are defined as $L_i(Q, W_t)$ and $U_i(Q, W_t)$, so that $L_i(Q, W_t) \le d(Q, F_i, W_t) \le U_i(Q, W_t)$. The computation of lower and upper bound of distance $d(Q, F_i, \Lambda_t)$ using a VA-file index is straightforward for a diagonal Λ_t [10]. For the case of general quadratic distance metric, nearest neighbor query becomes an ellipsoid query.

The locus of all points F_i having a distance $d(Q, F_i, W_t) \le \epsilon$ is an ellipsoid centered around query point Q. It is difficult to determine whether a general ellipsoid intersects a cell in the original feature space. Figure 2(a) illustrates the lower and upper bounds in the case of weighted Euclidean distance and quadratic distance. For quadratic metric, exact distance computation between the query object Q and a rectangle $C(F_i)$ requires numerically extensive quadratic programming approach. That would undermine the advantages of using an indexing structure.

Conservative bounds on rectangular approximations introduced in [11, 12] allow us to avoid exact distance computation between the query object Q and every approximation cell $C(F_i)$. However, for restrictive bounds, the distance computation stays quadratic with number of dimensions M. In [12], spatial transformation of the feature vector space significantly reduces CPU cost. We adopt a similar approach, further reducing computational cost and improving efficiency. Assuming that distance matrix W_t is real, symmetric, and positive definite, we can factorize W_t as $W_t = P_t^T \Lambda_t P_t$, $P_t * P_t^T = I$, and from (4):

$$d^2(Q, F_i, W_t) = (P_t(Q - F_i))^T \Lambda_t (P_t(Q - F_i)) \qquad (5)$$

Define a rotational mapping of a point D as $D \rightarrow D'$, where $D' = P_t D$. All quadratic distances in the original space transform to weighted Euclidean distances in the mapped space. Cell $C(D)$ that approximates feature point D is transformed into a hyper parallelogram $C(D')$ in the mapped space, as illustrated in Figure 2(b). The parallelogram $C(D')$ can be approximated with bounding hyper rectangular cell $C'(D')$. The approximation $C(D)$ only specifies the bounding rectangle position in the mapped space. The size of relative bounding rectangle depends only on the cell size in the original space, and the rotation matrix P. Relative bounding rectangle in the mapped space can be computed before Phase I filtering.

Note that the W_t update is computed before approximation level filtering. The weight matrix in the mapped space is Λ_t, and the quadratic distance becomes a weighted Euclidean distance. Lower and upper bounds, $L_i(Q, W_t)$ and $U_i(Q, W_t)$, respectively, are approximated in the mapped space with $L_i(Q', \Lambda_t)$ and $U_i(Q', \Lambda_t)$, using the weighted Euclidean matrix Λ_t, as in (5) (see Figure 2(c)). Also, $L_i(Q', \Lambda_t) \leq L_i(Q, W_t)$, and $U_i(Q, W_t) \leq U_i(Q', \Lambda_t)$, and therefore:

$$L_i(Q', \Lambda_t) \leq d(Q, F_i, W_t) \leq U_i(Q', \Lambda_t) \quad (6)$$

3.3. Adaptive Nearest Neighbor Search

Let $R_{t-1} = \{F_k^{(t-1)}\}$ be the set of K nearest neighbors of query Q at iteration $t-1$ under weight matrix W_{t-1}, and $r_t(Q)$ be the maximum distance between Q and the items in R_t. Define $r_t^u(Q) = max\{d(Q, F_k^{(t-1)}, W_t)\}, k \in [1, K]$. When W_{t-1} is updated to W_t, we can establish an upper bound on $r_t(Q)$ as: $r_t(Q) \leq r_t^u(Q)$, i.e. maximum of the distances between the query Q and objects in R_t computed using W_t, can not be larger than the maximum distance between the query Q and the objects in R_{t-1} computed using W_t. This is intuitively clear, since R_t is the set of K nearest neighbors from Φ to Q, under W_t.

Phase I filtering determines a subset of approximations from which the K nearest neighbors can be retrieved. Let $N_1^{opt}(Q, W_t)$ be the minimal set of approximations that contain K nearest neighbors. The best case scenario for Phase I filtering is to identify exactly this subset $N_1(Q, W_t) = N_1^{opt}(Q, W_t)$. For approximation $C(F_i)$ to be a qualified one in $N_1^{opt}(Q, W_t)$, its lower bound $L_i(Q, W_t)$ must satisfy:

$$L_i(Q, W_t) < r_t^u(Q) \quad (7)$$

Let ρ be the K^{th} largest upper bound encountered so far during a sequential scan of approximations. In the standard approaches [6], the approximation $C(F_i)$ is included in $N_1(Q, W_t)$ only if $L_i(Q, W_t) < \rho$, and the ρ is updated if $U_i(Q, W_t) < \rho$. Only the K^{th} smallest upper bound from the scanned approximations is available and used for filtering. The best filtering bound in Phase I is $min(r_t^u(Q), \rho)$. The relationship $r_t^u(Q, W_t) \leq \rho$ is satisfied for the larger part of database scanning. In general, fewer candidates need to be examined in Phase I filtering if we use $r_t^u(Q)$ as a filtering bound.

There are two essential differences between the the existing approaches and the proposed approach: (1) Only the absolute position of the approximation rectangle is computed during the Phase I filtering. We can compute the grid mapping in advance. Lower bound $L_i(Q, W_t)$ computation during Phase I is linear with number of dimensions for every VA-file index. (2) The proposed constraint in approximation filtering is based on relevance feedback results, and it gives us a smaller set of false candidates.

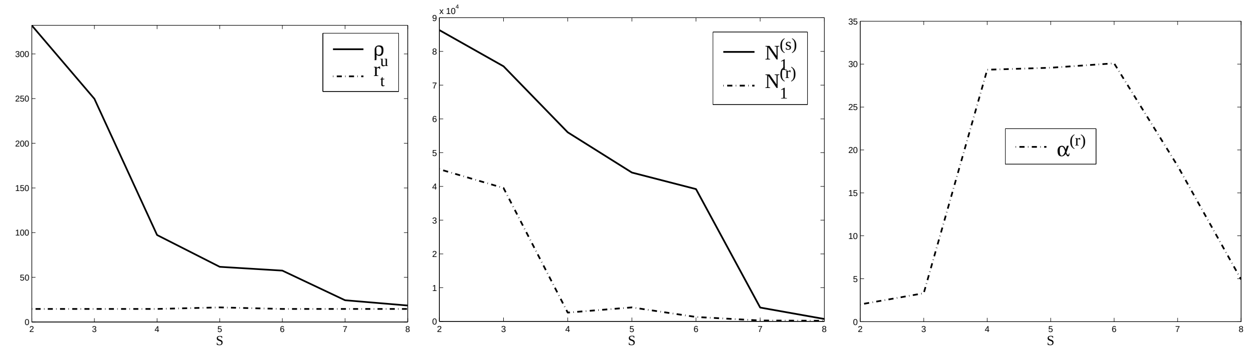
3.4. Experiments

We demonstrate the effectiveness of the proposed approach over a dataset of $N = 90774$ texture feature vectors, $M = 60$ dimensions each. Experiments are carried out for different resolutions S used for constructing standard VA, $S \in [2, 8]$. 2^S bits are assigned to each of the M uniformly partitioned feature dimensions. We compare the standard VA approach to K-NN search to our adaptive for different S, in relevance feedback presence.

Queries Q_i are selected from the dataset to cover both dense cluster representatives and outliers in the feature space. For each query Q_i, $K = 70$ nearest neighbors are retrieved during each iteration. The feedback from the user is based on texture relevance only. For a specific query, the user selects $K' = 65$ relevant retrievals to update the distance measure. The distance metric is updated before every iteration.

The approximations are constructed using the standard VA index. We choose to assign 2^S bits to all dimensions. Each of the feature dimensions is uniformly partitioned, at different resolutions. Experiments are carried out for different numbers of bits assigned to every dimension, $S \in [2, 7]$. Larger value corresponds to the approximation constructed at a finer resolution. We compare the standard VA approach to computing the K nearest neighbors to our proposed method for different S, in relevance feedback presence. Computation of upper bounds for the standard approach adds marginal complexity, since $U_i(Q, W_t)$ is found in a mapped space. Filter bound ρ rapidly increases when the resolution is smaller, due to the larger sizes of the hyper rectangles used in the corresponding approximations. A larger difference between r_t^u and ρ should impose a significant improvement for the proposed method. Thus, in the presence of relevance feedback we can either save some memory for approximation storage or reduce the number of disc access for the same resolution. Figure 3(a) shows that the difference between $r_t^u(Q)$ and ρ increases for lower values of S, for weighted Euclidean distance and quadratic.

For a given resolution S, and the query vector Q_i, number of candidates from Phase I filtering is noted as $N_1^{(s)}(Q_i)$ for standard approach and $N_1^{(r)}(Q_i)$ for adaptive approach. Define an average number of Phase I candidates over the example queries as $N_1^{(s)}$ for standard approach and $N_1^{(r)}$ for adaptive approach. Also, define an effectiveness measure of the proposed method: $\alpha^{(r)} = \frac{1}{I}\sum_{i=1}^{I} \frac{N_1^{(s)}(Q_i)}{N_1^{(r)}(Q_i)}$; We average over $I = 20$ query vectors. The number of candidates resulting from the standard and adaptive Phase I filtering, using quadratic distance metric, is given in Figure 3(b). The effectiveness $\alpha^{(r)}$ is significant, especially for coarser approximations, see Figure 3(c). This is a real dataset, and α is not monotonic over S, since the results are



(a) Phase I Filter bounds (b) Average number of Phase I candidates (c) Effectiveness measure α

Figure 3: Quadratic distance metric: a) Phase I selectivity bound ρ for standard filtering and r_t^u for adaptive one; b) Average number of cells accessed in Phase I for standard $N_1^{(s)}$ and adaptive approach $N_1^{(r)}$; c) Effectiveness measure $\alpha^{(r)}$ for Phase I adaptive method.

strongly correlated with distribution of the feature points in 60-dimensional space. We can conclude that the minimum gain of the proposed adaptive filtering is significant at every resolution.

4. Summary

We have shown that, when texture images are passed through Gabor filters, the outputs have a strong tendency to follow a Rayleigh distribution. Based on this, we have modified the MPEG-7 texture descriptor to have lower dimensionality and computational complexity. This benefits content-based retrieval systems by significantly reducing storage, computational expense, indexing overhead, and retrieval time. We support this approach by demonstrating that the new descriptor performs comparably with the MPEG-7 descriptor.

The paper also presented a framework that supports efficient retrieval of relevance feedback results, even when the similarity metric is quadratic. Based on the user's input, the weight matrix of the feature space is modified in every iteration, and a new set of nearest neighbors is computed. The cost of nearest neighbor computation in each iteration is quadratic in the number of dimensions and linear in the number of items. The proposed scheme allows us to use rectangular approximations for nearest neighbor search under a quadratic distance metric and exploits correlations between two consecutive nearest neighbor sets. The proposed approach significantly reduces the overall search complexity.

References

[1] C. Böhm, S. Berchtold, and D. A. Keim, "Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases," *ACM Computing Surveys*, vol. 33, no. 3, pp. 322–373, Sep 2001.

[2] B. S. Manjunath, P. Salembier, and T. Sikora, Eds., *Introduction to MPEG7: Multimedia Content Description Interface*, John Wiley & Sons Ltd., 2002.

[3] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. PAMI*, vol. 18, no. 8, pp. 837–842, Aug 1996.

[4] D. Dunn and W. E. Higgins, "Optimal gabor filters for texture segmentation," *IEEE Trans. Image Proc.*, vol. 4, no. 7, pp. 947–964, Jul 1995.

[5] S. Bhagavathy, J. Tešić, and B. S. Manjunath, "On the Rayleigh nature of Gabor filter outputs," in *Intl. Conf. on Image Processing (ICIP)*, Sep 2003.

[6] R. Weber, H. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *Proc. Very Large Databases (VLDB)*, Aug 1998, pp. 194–205.

[7] Y. Rui and T. Huang, "Optimizing learning in image retrieval," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Jun 2000, vol. 1, pp. 236–243.

[8] H. Y. Rui and T. Huang, "Content-based image retrieval with relevance feedback in MARS," in *Proc. Intl. Conf. on Image Processing (ICIP)*, Oct 1997, vol. 2, pp. 815–818.

[9] Y. Ishikawa, R. Subramanya, and C. Faloutsos, "Mindreader: Query databases through multiple examples," in *Proc. Very Large Databases (VLDB)*, Aug 1997, pp. 218–227.

[10] J. Tešić and B. S. Manjunath, "Nearest neighbor search for relevance feedback," in *CVPR*, 2003.

[11] M. Ankerst, B. Braunmüller, H.-P. Kriegel, and T. Seidl, "Improving adaptable similarity query processing by using approximations," in *Proc. Very Large Databases (VLDB)*, Aug 1998, pp. 206–217.

[12] Y. Sakurai, M. Yoshikawa, R. Kataoka, and S. Uemura, "Similarity search for adaptive ellipsoid queries using spatial transformation," in *Proc. Very Large Databases (VLDB)*, Sep 2001, pp. 231–240.