ELSEVIER

Pattern Recognition Letters 22 (2001) 63–74

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Independent feature analysis for image retrieval

Jing Peng [a], Bir Bhanu [b,*]

[a] Department of Computer Science, Oklahoma State University, Stillwater, OK 74078, USA

[b] Center for Research in Intelligent Systems, University of California, Bourns Hall A220, Riverside, CA 92521, USA

Abstract

Content-based image retrieval methods based on the Euclidean metric expect the feature space to be isotropic. They suffer from unequal differential relevance of features in computing the similarity between images in the input feature space. We propose a learning method that attempts to overcome this limitation by capturing local differential relevance of features based on user feedback. This feedback, in the form of accept or reject examples generated in response to a query image, is used to locally estimate the strength of features along each dimension while taking into consideration the correlation between features. This results in local neighborhoods that are constricted along feature dimensions and that are most relevant, while elongated along less relevant ones. In addition to exploring and exploiting local principal information, the system seeks a global space for efficient independent feature analysis by combining such local information. We provide experimental results that demonstrate the efficacy of our technique using both simulated and real-world data. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Feature analysis; Image retrieval

1. Introduction

The rapid advance in digital imaging technology makes possible the widespread use of image libraries and databases. This in turn demands effective means for access to such databases. It has been well documented that simple textual annotations for images are often ambiguous and inadequate for image database search. Thus, retrieval based on image "content" becomes very attractive (Cox et al., 1996; Flickner et al., 1995; Ma and Manjunath, 1996; Minka and Picard, 1997; Peng et al., 1999; Rui et al., 1997). Generally, a set of features (color, shape, texture, etc.) are extracted from an image to represent its content. As such, image database retrieval becomes a K nearest neighbor (K-NN) search in a multidimensional space defined by these features under a given similarity metric.

Simple K nearest neighbor search, as an image retrieval procedure, returns the K images closest to the query. Obviously this involves the issue of measuring the closeness or similarity between two images. The most common measure of the similarity between two images, represented by their feature vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, is the distance between them. If the Euclidean distance $D(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{q}(x_i - y_i)^2}$ is used, then the K closest images to the query $\boldsymbol{x}_Q$ are computed according to $\{\boldsymbol{x} \mid D(\boldsymbol{x},\boldsymbol{x}_Q) \leqslant d_K\}$, where d_K is the Kth order statistic of $\{D(\boldsymbol{x}_i,\boldsymbol{x}_Q)\}_{i=1}^{N}$. Here N is the number of

* Corresponding author. Tel.: +1-909-787-3954; fax: +1-909-787-2425.

E-mail addresses: jpeng@cs.okstate.edu (J. Peng), bhanu@cris.ucr.edu (B. Bhanu).

PII: S 0 1 6 7 - 8 6 5 5 (0 0) 0 0 1 0 0 - 8

images in the database. The major appeal for simple K-NN search methods resides in their ability to produce continuous and overlapping rather than fixed neighborhoods, and to use a different neighborhood for each individual query so that, to the extent possible, all points in the neighborhood are close to the query.

One problem with the Euclidean metric, however, is that it does not take into account the influence of the scale of each feature variable in the distance computation. Changing the scale of a feature dimension in different amounts alters the overall contribution of that dimension to the distance computation, hence its influence in the nearest neighbors retrieved. This is usually considered undesirable. An additional limitation is that the use of Euclidean distance, while simple computationally, implies that the input feature space is isotropic. However, isotropy is often invalid and irrelevant features might hurt retrieval performance. Finally, feature relevance depends on the location where the query is made in the input feature space. Capturing such relevance information is a prerequisite for constructing successful retrieval procedures for image databases.

In this paper, we propose a novel method that provides a solution to the problems discussed above. With this method image retrieval system is able to learn differential feature relevance in an efficient manner that takes into account the correlation between features, and that is highly adaptive to query locations. In addition, by accumulating experience obtained at each iteration the system is capable of, to the extent possible, continuously improving its retrieval performance.

2. Previous work

Minka and Picard (1997) select features based on feedback from user but all features are treated with equal importance. Ma and Manjunath (1996) use hybrid neural networks to partition the feature space into clusters but no user feedback is used to refine retrievals and all features are treated as equally important. Rui et al. (1997) in their MARS system use a simple query shifting mechanism that attempts to improve retrieval performance by adaptively moving the input query toward relevant retrievals and, at the same time, away from irrelevant ones. Similarity computation remains fixed throughout the retrieval process. While MARS has been shown to improve retrieval performance in simple tasks, it is clear that in many problems the mere shifting of the query is insufficient to achieve desired goals. PicHunter (Cox et al., 1996) is a system based on Bayesian relevance feedback. PicHunter defines a set of actions that a user might take and attempts to estimate the probabilities of the actions the user will take. Based on actual actions taken, the system estimates the probability of each image in the database being the target. A major concern with PicHunter is that probability estimates rely heavily on simplifying assumptions that are often invalid in practice. Furthermore, its image similarity measure is nonadaptive. Peng et al. (1999) use probabilistic feature relevance learning for content-based image retrieval that computes flexible metrics for producing retrieval neighborhoods that are elongated along less relevant feature dimensions and constricted along most influential ones. The technique has shown promise in a number of image database applications (Peng and Bhanu, 1999). It, however, becomes less appealing in situations where feature relevance can only be captured by examining several feature variables simultaneously. In this paper, we develop techniques that attempt to overcome this limitation.

3. Local feature relevance

In a two class (1/0) classification problem, the class label $y \in \{0, 1\}$ at query $\boldsymbol{x}$ is treated as a random variable from a distribution with the probabilities $\{Pr(1 \mid \boldsymbol{x}), Pr(0 \mid \boldsymbol{x})\}$. We then have $f(\boldsymbol{x}) \doteq Pr(y = 1 \mid \boldsymbol{x}) = E(y|\boldsymbol{x})$. To predict y at $\boldsymbol{x}$, $f(\boldsymbol{x})$ is first estimated from a set of training data using techniques based on regression, such as the least-squares estimate. The Bayes classifier can thus be applied to achieve optimal classification performance. In image retrieval, however, the "label" of $\boldsymbol{x}$ is known, which is 1 (positive image) in terms of the notation given above. All that is required is to exploit differential relevance of the input features to image retrieval. In the absence of any variable assignments, the least-squares estimate for $f(\boldsymbol{x})$ is $E[f] = \int f(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, where $p(\boldsymbol{x})$ is the joint density. Now given only that $\boldsymbol{x}$ is known at dimension $x_i = z_i$. The least-squares estimate becomes $E[f \mid x_i = z_i] = \int f(\boldsymbol{x})p(\boldsymbol{x} \mid x_i = z_i)\,\mathrm{d}\boldsymbol{x}$. Here $p(\boldsymbol{x} \mid x_i = z_i)$ is the conditional density of the other input variables.

In image retrieval, $f(\boldsymbol{z}) = 1$, where $\boldsymbol{z}$ is the query. Then $[(f(\boldsymbol{z}) - 0) - (f(\boldsymbol{z}) - E[f \mid x_i = z_i])] = E[f \mid x_i = z_i]$ represents a reduction in error between the two predictions. We can now define a measure of feature relevance at query $\boldsymbol{z}$ as $r_i(\boldsymbol{z}) = E[f \mid x_i = z_i]$. The relative relevance can be used as a weighting scheme. The following exponential weighting scheme:

$$w_i(\boldsymbol{z}) = \exp(Tr_i(\boldsymbol{z})) \Big/ \sum_{l=1}^{q} \exp(Tr_l(\boldsymbol{z})) \tag{1}$$

is employed in this paper. Here T is a parameter that can be chosen to maximize (minimize) the influence of r_i on w_i. A discussion on the choice of a value for T can be found in (Peng et al., 1999). From (1), we obtain the weighted Euclidean distance $D(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{q} w_i(x_i - y_i)^2}$.

In order to estimate (1), one must first compute $E[f \mid x_i = z_i]$. The retrieved images with relevance feedback from the user can be used as training data to obtain estimates for $E[f \mid x_i = z_i]$, hence (1). Let $\{\boldsymbol{x}_j, y_j\}_1^K$ be the training data, where $\boldsymbol{x}_j$ denotes the feature vector representing the jth retrieved image, and y_j is either 1 (positive image) or 0 (negative image) marked by the user as the class label associated with $\boldsymbol{x}_j$. Since there may not be any data at $x_i = z_i$, the data from the vicinity of x_i at z_i are used to estimate $E[y \mid x_i = z_i]$, a strategy suggested in (Friedman, 1994). That is,

$$\hat{E}[y \mid x_i = z_i] = \frac{\sum_{j=1}^{K} y_j 1\big(\mid x_{ji} - z_i \mid \leqslant \Omega\big)}{\sum_{j=1}^{K} 1\big(\mid x_{ji} - z_i \mid \leqslant \Omega\big)}, \tag{2}$$

where $1(\cdot)$ is 1 if its argument is true, and 0 otherwise. Ω can be chosen so that there are sufficient data for the estimation of $E[f \mid x_i = z_i]$. In this paper, Ω is chosen such that

$$\sum_{j=1}^{K} 1\big(\mid x_{ji} - z_i \mid \leqslant \Omega\big) = C, \tag{3}$$

where $C \leqslant K$ is a constant. It represents a trade-off between bias and variance and should be chosen to lie between 1 and K.

4. Feature decorrelation using local principal information

In order for Eq. (1) to be effective, features must be independent. However, this condition can hardly be met in practice. Often, there is a degree of correlation among the input features. Our goal here is to seek a space into which to project data so that the feature dimensions coincide with the eigenvectors of the space, whereby feature relevance can be estimated at the query along individual dimensions independently. The novelty here is in choosing the space using pooled local principal information.

We begin with linear multivariate analysis. Given a set of q-dimensional data $\{\boldsymbol{x}_j\}_{j=1}^{n}$, the eigenspace is the space spanned by eigenvectors onto which the feature dimensions representing the projected data are statistically uncorrelated. Let $\bar{\boldsymbol{x}}$ denote the mean for the observed data. Then this space corresponds to the eigenspace of the data points $\boldsymbol{x}_j - \bar{\boldsymbol{x}}$, or the local covariance matrix at the query.

In the context of image retrieval, the basic idea is to compute $\boldsymbol{x}_j - \bar{\boldsymbol{x}}$ at a given query using only local information, and then perform the principal component analysis for the local scatter matrix. Specifically, let $\{\boldsymbol{x}_j(i)\}_1^n$ be the set of n nearest neighbors obtained at query i, and $\bar{\boldsymbol{x}}(i)$ be the associated mean vector. Also let

$$\boldsymbol{S}(i) = \frac{1}{n} \sum_{j=0}^{n} \Big(\boldsymbol{x}_j(i) - \bar{\boldsymbol{x}}(i)\Big)\Big(\boldsymbol{x}_j(i) - \bar{\boldsymbol{x}}(i)\Big)^{\mathrm{t}} \tag{4}$$

be the scatter matrix at query i, where t denotes transpose. We compute the space spanned by the eigenvectors of $\boldsymbol{S}(i)$ (for query i) onto which the feature relevance analysis of the projected data can be performed independently along each dimension.

1. Let i be current query; initialize weight vector $\mathbf{w}$ to $\{1/q\}_1^q$; training set: $Tset = nil$.
2. Compute $\{\mathbf{x}_j\}_1^n$, set of n nearest neighbors using $\mathbf{w}$; return the first K nearest neighbors in $\{\mathbf{x}_j\}_1^n$ to the user.
3. User marks the K retrieved images as positive (1) or negative (0).
4. Calculate the local scatter matrix $\mathbf{S}(i)$ using $\{\mathbf{x}_j\}_1^n$.
5. Project M nearest neighbors onto the eigenspace of $\mathbf{S}(i)$.
6. While $precision < \theta$ Do
   (a) $Tset = Tset \cup \{K\ points\}$.
   (b) Update $\mathbf{w}$ (Eq. 1) in the transformed space using training data in $Tset$.
   (c) Compute K nearest neighbors using $\mathbf{w}$ in the transformed space.
   (d) User marks the K points as positive (1) or negative (0).

Fig. 1. The AFRE algorithm.

The overall algorithm is summarized in Fig. 1. Here K denotes the number of images returned to the user, and M is an adjustable procedural parameter. In general, $M \gg K$. We call this algorithm adaptive feature relevance estimation (AFRE). Note that n is the number of data points used to compute the local scatter matrix, whereas M represents the number of data points projected onto the eigenspace within which feature relevance computation is carried out.

The algorithm just described computes the local scatter matrix $\boldsymbol{S}(i)$ for each given query i. As such, it is capable of computing a neighborhood that is highly adaptive to query locations, thereby significantly improving retrieval performance. On the other hand, a potential weakness of the method is that the estimated covariance matrix centered at the given query might be far different from the true covariance matrix that characterizes actual data distributions due to the nature of computation, which is based solely on limited local samples. We seek a method here to overcome this limitation by finding a space that is close to the eigenspace of the average local scatter matrices, $\boldsymbol{S}(i)s$, over all queries.

If we denote by $\boldsymbol{U}$ an orthonormal basis for the q-dimensional space, we can obtain the space by minimizing the following total residual sum of squares

$$R(\boldsymbol{U}) = \sum_{i=1}^{N} \sum_{j=1}^{n} (\tilde{\boldsymbol{x}}_j^{\mathrm{t}}(i)\tilde{\boldsymbol{x}}_j(i) - \tilde{\boldsymbol{x}}_j^{\mathrm{t}}(i)\boldsymbol{U}\boldsymbol{U}^{\mathrm{t}}\tilde{\boldsymbol{x}}_j(i)),$$

where $\tilde{\boldsymbol{x}}_j(i) = (\boldsymbol{x}_j(i) - \bar{\boldsymbol{x}}(i))$, n is the number of local retrievals, and N is the number of queries. We can obtain the minimum of the above equation by maximizing $\mathrm{tr}\boldsymbol{U}^{\mathrm{t}}\{\sum_{i=1}^{N} \boldsymbol{S}(i)\}\boldsymbol{U}$, where tr represents the trace operation of a matrix. This can be solved by finding the eigenvectors of the following matrix $\bar{\boldsymbol{S}} = \sum_{i=1}^{N} \boldsymbol{S}(i)/N$, which is the average scatter matrices over all queries. The main benefit of averaging is that it reduces variance due to skewed local information, thereby improving overall estimation accuracy. If we add $\bar{\boldsymbol{S}} = \bar{\boldsymbol{S}} + (\boldsymbol{S}(i) - \bar{\boldsymbol{S}})/(l+1)$; $l = l + 1$ to Fig. 1, immediately after step 4, and replace $\boldsymbol{S}(i)$ by $\bar{\boldsymbol{S}}$ in step 5, we obtain the learning feature relevance estimation (LFRE). Here both $\bar{\boldsymbol{S}}$ and l are initialized to zero.

While LFRE demonstrates improvements over AFRE on the problems we examine here, it is still

4. If **not** $(\nu(\bar{\mathrm{S}} - \bar{\mathrm{S}}_o) < \delta)$ then

  a. $\bar{\mathrm{S}}_o = \bar{\mathrm{S}}$.

  b. Calculate the local scatter matrix $\mathrm{S}(i)$ using the K points.

  c. $\bar{\mathrm{S}} = \bar{\mathrm{S}} + (\mathrm{S}(i) - \bar{\mathrm{S}})/(l+1)$; $l = l + 1$.

Fig. 2. The A-LFRE algorithm.

a rather computationally intensive process. We propose to approximate the average scatter matrix without compromising the level of achievable performance. The basic assumption is that $\bar{S}$ can be reasonably estimated from a few representative local $S(i)$s. Specifically, we approximate $\bar{S}$ by incrementally combining the local $S(i)$'s computed for the queries seen so far, similar to the way it is computed in LFRE. However, the estimation process stops when $\bar{S}$ becomes sufficiently accurate. Subsequent feature relevance estimates are carried out in the space spanned by the eigenvectors of $\bar{S}$. In this paper, we measure the accuracy of $\bar{S}$ using a matrix norm v. That is, we say that $\bar{S}$ is accurate if $v(\bar{S}_{t+1} - \bar{S}_t)$ is sufficiently small. While other measures exist, we do not intend to address this issue further in the rest of this paper.

If we replace line 4 in AFRE (Fig. 1) by the following, and $S(i)$ by $\bar{S}$ in line 5, we arrive at the approximate learning feature relevance estimation (A-LFRE) algorithm (Fig. 2). Here $\bar{S}_0$ is initialized to 0, $v(\cdot)$ represents the matrix norm operator, and δ is a constant parameter input to the algorithm.

5. Empirical results

In the following we compare three competing retrieval methods using both real and simulated data. (a): The probabilistic feature relevance learning (PFRL) algorithm (Peng et al., 1999) coupled with the exponential weighting scheme (1). Note that this algorithm, unlike the ones described above, computes local feature relevance in the

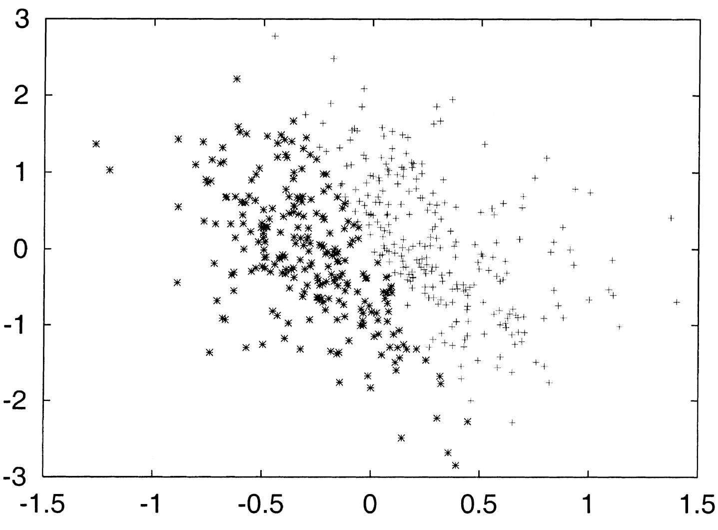


Fig. 3. A simple two class problem with substantial within class covariance between the two input features.

original feature space. It does not perform feature decorrelation. (b): The AFRE algorithm described above. Again, the algorithm is coupled with the exponential weighting scheme (1). (c): The LFRE algorithm described above. Similar to (a) and (b) above the algorithm is coupled with the exponential weighting scheme (1). Note that there is a fourth method, the simple (unweighted) K-NN method, that is being compared against implicitly. The first retrieval by all the three methods is based on the unweighted K-NN method. Also, in all the experiments, the performance is measured using the following average retrieval precision (Peng et al., 1999; Rui et al., 1997):

$$\text{precision} = \frac{\text{Positive retrievals}}{\text{Total retrievals}} \times 100\%.$$

In all the experiments the input features are first normalized. The normalization is performed along each feature dimension over the entire data set in such a way that the normalized feature values lie between 0 and 1. It simply removes some of the artifacts due to different scales of variables that are generally considered undesirable in the absence of any additional information. The number of retrieved images, K, at each iteration is set to 20 that provide relevance feedback. As far as the relative performance of AFRE, LFRE and PFRL is concerned, the choice of K will not alter the qualitative behaviors of the three techniques observed in the databases, provided that K is not too small.

5.1. Simulated data experiments

5.1.1. The problem

In the simulated data experiments we use a two class problem with substantial within class covariance between the two features, as shown in Fig. 3. There are 250 data points in each class. This problem clearly favors algorithms that first rotate the feature dimensions so that they coincide with the eigenvectors of a sample covariance matrix, and then perform feature relevance analysis. In these experiments, each data point is selected as a query, and the average retrieval precisions by the three competing methods are reported.

5.1.2. Results

In these experiments, the procedural parameters T (1), C (3), n and M input to the algorithms under comparison were determined empirically that achieved the best performance. They were set to 12 and 25 (PFRL, parameters n and M are not applicable to PFRL), 12, 22, 80 and 400 (AFRE), and 14, 24, 80 and 400 (LFRE), respectively. Fig. 4(a) plots the performance of the three algorithms: PFRL, AFRE and LFRE as a function of iteration. Both AFRE and LFRE demonstrate performance improvement over PFRL, as expected, and more so by LFRE because of its averaging effect.

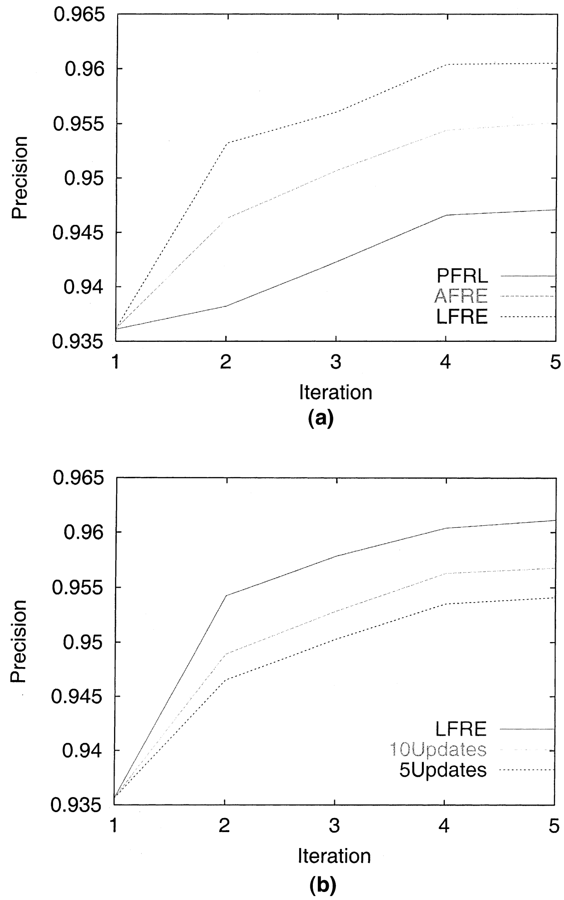


Fig. 4. Effect of feature decorrelation on retrieval. (a) Performance of PFRL, AFRE and LFRE. (b) Performance of A-LFRE.

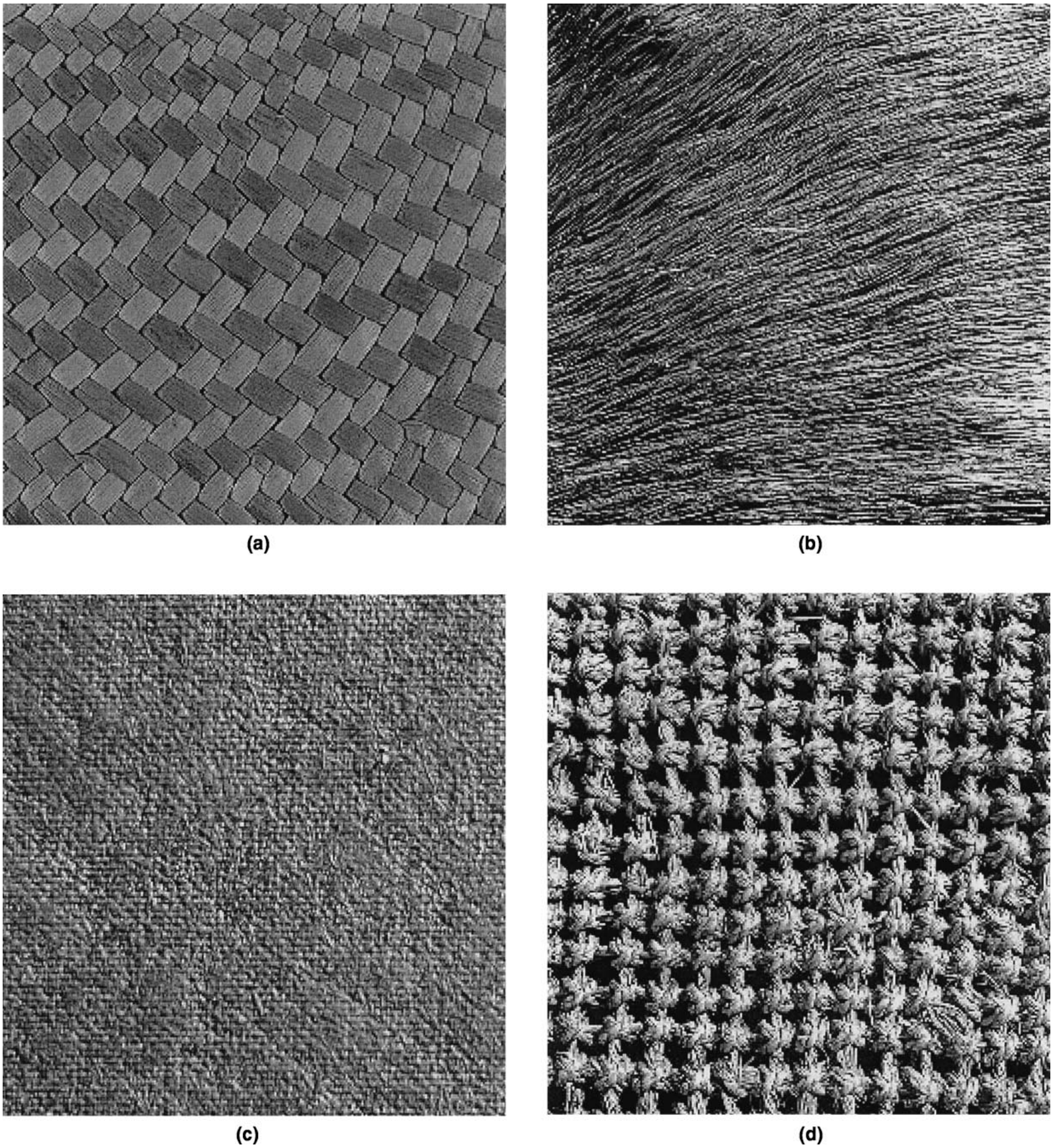
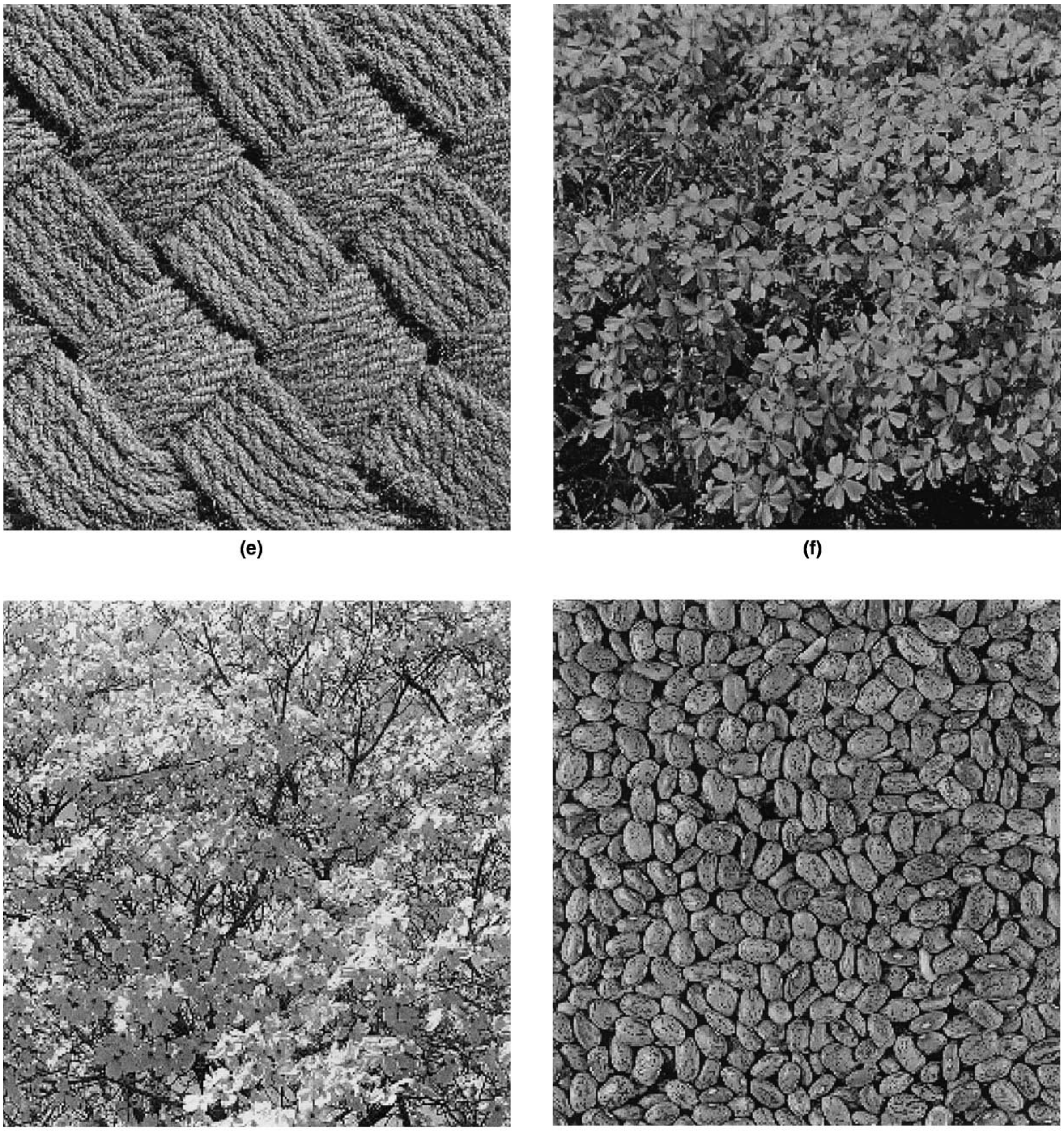


Fig. 5. Sample images from MIT database.

We carried out an additional experiment to examine the performance of the A-LFRE algorithm (Fig. 2) on the same problem shown in Fig. 3. In this experiment, 400 data points were uniformly randomly selected, each of which was used as a query. This process was repeated 10 times, and the average retrieval precisions by LFRE and A-LFRE were reported in Fig. 4(b). Note that in these first

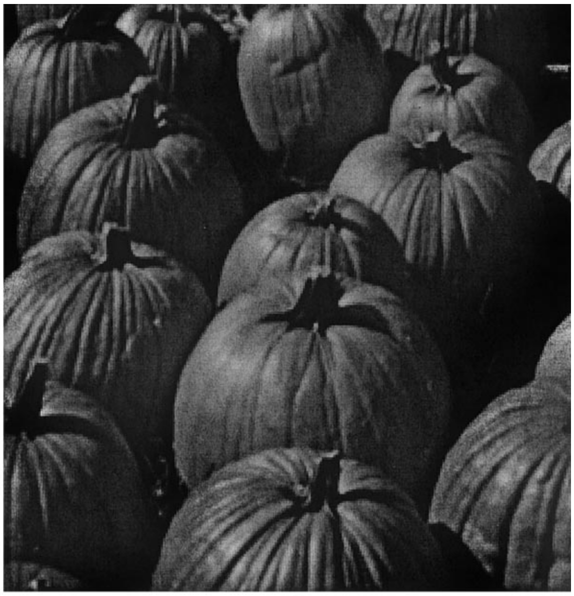


(e) (f) (g) (h)

Fig. 5. (Continued)

experiments, we compared the results achieved by a fixed number of updates to $\bar{\boldsymbol{S}}$, as in the A-LFRE algorithm, against that achieved by a total update, as in the LFRE algorithm. The results show that the significant level of performance can be achieved by A-LFRE with fewer $\bar{\boldsymbol{S}}$ updates. The procedural parameters used in this experiment were 14 (T), 24 (C), 80 (n) and 400 (M) for both algorithms.

(i)



(j)

Fig. 5. (Continued)

5.2. Real data experiments

5.2.1. Database 1

The data, taken from the UCI repository (Murphy and Aha, 1995), consist of images that were drawn randomly from a database of seven outdoor images. The images were hand segmented by the creators of the database to classify each pixel. Each image is a region. There are seven classes: *brickface*, *sky*, *foliage*, *cement*, *window*, *path* and *grass*, each having 330 instances. Thus, there are totally 2310 images in the database. These images are represented by 19 real valued attributes. These features are basically statistical moments and line counts. For further details, see Murphy and Aha (1995).

5.2.2. Database 2

The data are obtained from MIT Media Lab. [1] There are total 640 images of 128×128 in the database with 15 classes. The number of images in each class varies from 16 to 80. The images in this database are represented by 8 Gabor filters (2 scales and 4 orientations), giving rise to a 16-dimension feature space. The mean and S.D. of the magnitude of the transform coefficients are used as feature components after being normalized by the standard deviations of the respective features over the images in the entire database. Fig. 5 shows sample images from the MIT dataset.

5.2.3. Results

For both the problems, each image in the database is selected as a query and top 20 (corresponding to parameter K in the algorithms described above) nearest neighbors are returned that provide necessary relevance feedback. Note that only negative images (that are different from the query) need to be marked in practice. Also, M was set to 400 in AFRE and LFRE in these experiments. The average retrieval performance by the three competing algorithms is plotted in Fig. 6.

The first iteration in Fig. 6 shows the average retrieval precision obtained without any relevance feedback. That is, it is the result of applying the simple K-NN method using unweighted Euclidean metric. The second iteration and beyond show the

[1] Whitechapel.media.mit.edu/pub/VisTex.

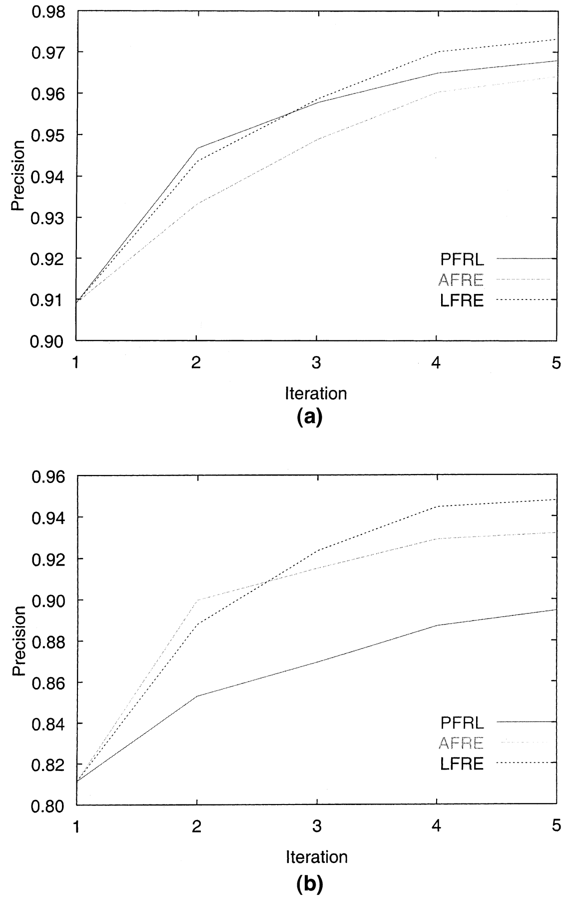


Fig. 6. Effect of feature decorrelation on retrieval. (a) Performance of PFRL, AFRE and LFRE on the UCI data set. (b) Performance of PFRL, AFRE and LFRE on the MIT data set.

average retrieval precision obtained after learning has taken place. That is, relevance feedback obtained from the previous retrieval is used to estimate local feature relevance, hence a new weighting. The procedural parameters T (1), C (3) and n input to the algorithms under comparison were determined empirically that achieved the best performance. They are by no means exhaustive. For the UCI image database, they were set to 13 and 19 (PFRL, parameter n is not applicable to PFRL), 13, 21 and 200 (AFRE), and 13, 27 and 200 (LFRE), respectively; while for the MIT image database, they were set to 13 and 20 (PFRL, again n is not applicable to PFRL), 15, 19 and 50 (AFRE), and 14, 19 and 70 (LFRE), respectively.

It can be seen from Fig. 6 that LFRE demonstrates performance improvement across the tasks and over both PFRL and AFRE. However, the improvement is most pronounced on the MIT data set. The reason is that features based on Gabor wavelet filters exhibit a degree of correlation because these filters partially overlap. On the other hand, the features representing the UCI data set are less correlated. Overall, the results show convincingly that feature decorrelation plays an important role in improving feature relevance estimates.

An additional experiment was also carried out to examine the performance of the A-LFRE algorithm (Fig. 2). In this experiment, 600 images are randomly chosen from the MIT database as query images. We ran both LFRE and A-LFRE on this database and obtained the average retrieval precisions on the query sets. We repeated this process for 10 times, and plotted the average precisions over the 10 runs at iterations 1, 2, 3, 4 and 5 in Fig. 7. Note that instead of computing a matrix norm to measure the accuracy of $\bar{S}$, as a first approximation A-LFRE simply computes a fixed number of updates to $\bar{S}$, after which $\bar{S}$ is fixed throughout.

The plots in Fig. 7 show that after only a few updates of the average scatter matrix A-LFRE approached the level of performance obtained by LFRE. Furthermore, A-LFRE did so with far less computation than that required by LFRE, thereby demonstrating its computational efficiency and advantage in practical applications. We performed similar experiments on the UCI database, where 2000 images are randomly selected as query images. We omit the details of the experiments except stating that similar results to that of the MIT database were obtained.

5.2.4. Discussions

One might contemplate to use a covariance matrix computed from a set of samples to decorrelate the entire database off-line, and then to perform feature relevance estimate in the transformed feature space using the techniques presented in this paper. There may be several reasons against such an idea. The most important one is when the database is dynamic, then an off-line operation may not be feasible. In one experiment

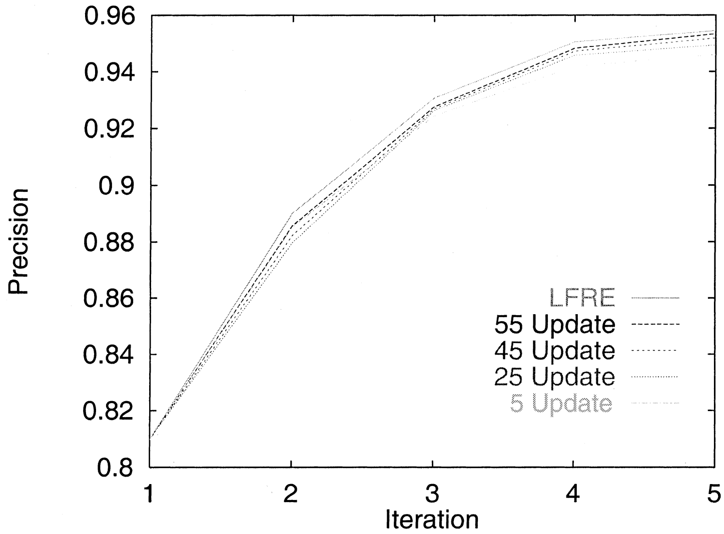


Fig. 7. Performance of the A-LFRE algorithm on the MIT database.

the entire UCI data set is used to compute the scatter matrix. We then project the original data into the eigenspace of the matrix and perform corresponding feature relevance estimates. We obtained the following average retrieval precisions at iterations 1, 2, 3, 4 and 5: 91.14, 94.99, 96.07, 96.70 and 96.86, respectively. These results and those obtained earlier (the UCI data set in Fig. 6(a)) clearly favor the on-line techniques proposed in this paper. We obtained similar results on the MIT database.

Two databases used here have a relatively small number of data in each class. The effectiveness of AFRE and LFRE in databases where each class may contain a large number of images depends critically on the effectiveness of PFRL (Peng et al., 1999) in such databases, since both AFRE and LFRE can be viewed as a special case of PFRL that computes feature relevance in the eigenspace of a scatter matrix. For a detailed empirical evaluation of PFRL in such databases, please see Peng and Bhanu (1999).

6. Conclusions

This paper presents a novel method for learning the differential feature relevance for a given query that takes into consideration the correlation between features. In addition to exploring and exploiting local discriminant information, the system seeks a global space for efficient independent feature analysis by combining such local information. Furthermore, by accumulating experience obtained at each iteration the system is capable of, to the extent possible, continuously improving its retrieval performance. The experimental results presented demonstrate the potential for substantial improvements in both the technique presented here and simple K-NN search.

Acknowledgements

This work was supported in part by DARPA/AFOSR grant F49620-97-1-0184 at the University of California at Riverside. The contents of the information do not necessarily reflect the position or the policy of the US Government.

References

Cox, I., Miller, M., Omohundro, S., Yianilos, P., 1996. PicHunter: Bayesian relevance feedback for image retrieval. In: Proceedings of the International Conference on Pattern Recognition. Vienna, Austria, pp. 361–369.

Flickner, M., et al., 1995. Query by image and video content: the QBIC system. IEEE Computer, pp. 23–31, September 1995.

Friedman, J.H., 1994. Flexible metric nearest neighbor classification. Technical Report, Department of Statistics, Stanford University.

Ma, W.Y., Manjunath, B.S., 1996. Texture features and learning similarity. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Francisco, pp. 425–430, June 1996.

Minka, T.P., Picard, R.W., 1997. Interactive learning with a "society of models". Pattern Recognition 30 (4), 565–581.

Murphy, P.M., Aha, D.W., 1995. UCI repository of machine learning databases. www.cs.uci.edu/~mlearn/MLRepository.html..

Peng, J., Bhanu, B., Qing, S., 1999. Probabilistic feature relevance learning for content-based image retrieval. Computer Vision and Image Understanding 75 (1/2), 150–164.

Peng, J., Bhanu, B., 1999. Feature relevance estimation for image databases. In: Proceedings of the Fifth International Workshop on Multimedia Information Systems, pp. 12–19.

Rui, Y., Huang, T.S., Mehrotra, S., 1997. Content-based image retrieval with relevance feedback in MARS. In: Proceedings of the IEEE International Conference on Image Processing. Santa Barbara, California, October 1997, pp. 815–818.