Dimensionality Reduction Using Multi-Dimensional Scaling for Content-Based Retrieval

Morris Beatty and B. S. Manjunath

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
E-mail: mlbeattty@iplab.ece.ucsb.edu, manj@ece.ucsb.edu

Abstract

There has been much interest recently in image content based retrieval, with applications to digital libraries and image database accessing. One approach to this problem is to base retrieval from the database upon feature vectors which characterize the image texture. Since feature vectors are often high dimensional, Multi-Dimensional Scaling, or Non-Linear Principal Components Analysis (PCA) may be useful in reducing feature vector size, and therefore computation time. We have investigated a variant of the non-linear PCA algorithm described in [6] and its usefulness in the database retrieval problem. The results are quite impressive; in an experiment using an aerial photo database, feature vector length was reduced by a factor of 10 without significantly reducing retrieval performance.

1 INTRODUCTION

Systems for content based retrieval from large image databases have been the focus of considerable recent effort[4][1]. It has been shown in [2][3] that a bank of Gabor filters can be used to generate texture feature vectors which capture the frequency content of an image such as a texture sample or a tile from an aerial photograph. These feature vectors describe the appearance of image tiles so well that retrieval from the database can be done by matching feature vectors of images in the database with the feature vector of the query image. Of course, since real image databases are quite large, efficient search techniques must be developed for practical systems. In [3] is described a working system which uses texture feature vectors to retrieve images from a large aerial photo database, based upon a given query image. To do this efficiently, a neural network is trained to divide the database vectors into groups which have a similar appearance. This training has a supervised stage, where class information about the training vectors is provided to fine tune the neural net. The result is a texture thesaurus of representative feature vectors, used as follows. Given a query image, one first computes its feature vector. Then the thesaurus is searched for the closest thesaurus vector, or code word. This code word represents a class of images from the database. Once the closest code word to the query image feature vector is found, only those database feature vectors belonging to the code word's class are searched for the best matches to the query image. Use of the thesaurus cuts down on the search time greatly without significantly reducing retrieval performance.

However, even greater improvements in retrieval efficiency are desirable. Because the database to be searched is large and the feature vectors are of high dimension, search complexity is still high. Promising results suggest that non-linear PCA may be useful in reducing the dimension of the feature vectors without destroying too much of the information they contain. Experiments show that with a database of 1333 64x64 image tiles from 10 different classes, non-linear PCA can project 60 dimensional feature vectors to just 6 dimensions, while maintaining very high retrieval rates. Thus an efficient retrieval system architecture might use neural nets to initially direct query vectors to subclasses as described above, and then apply a non-linear projection to a lower dimension before searching the subclass for the best matches.

In the following section we describe briefly how the texture feature vectors used in our experiments are computed. See [3] for a more detailed description. After that, we summarize how we make use of the algorithm described in [6]. Finally, we present our results and provide a few examples comparing different retrieval methods. We close with a description of future research possibilities.

2 FEATURE VECTORS

Image tiles of size 64x64 pixels were taken from an aerial photograph, yielding a total database of 1333 image tiles. These tiles were judged by eye to contain 10 similarity classes.

As described in [3], a bank of 60 Gabor filters was used to generate a feature vector for each image tile roughly as follows. Define

$$w_{mn}(x, y) = \int\int I(x, y) G_{mn}(x - x_1, y - y_1) dx_1 dy_1$$

where $I(x,y)$ is the image whose feature vector we are in the process of finding and $G_{mn}(x,y)$ is a Gabor kernel with scale parameter m and orientation parameter n. In our experiments, we used 5 scales and 6 orientations. The components of the feature vector are then the mean and variance of the function $w_{mn}(x,y)$ defined above:

$$\mu_{mn} = \int\int w_{mn}(x, y) dx dy$$

and

$$\sigma_{mn} = \sqrt{\int\int (w_{mn} - \mu_{mn})^2 dx dy}$$

Finally, the 60 dimensional feature vectors are $[\mu_{00}, \sigma_{00}, ..., \mu_{45}, \sigma_{45}]$.

3 ITERATIVE MAJORIZATION

To reduce the dimension of the feature vectors we use the non-linear PCA technique described in [6]. The idea is to assume that the desired projection has the form

$$v_p = W\Phi(v)$$

where v_p is the projected vector of dimension m, W is a constant m by n matrix whose value is to be determined, Φ is an n by 1 matrix whose entries are multivariate Gaussians, and v is the feature vector to be projected. We want to find a value of W which minimizes the cost function defined by

$$\sigma(W) = \Sigma \alpha_{ij}(q_{ij} - d_{ij})^2$$

where q_{ij} is the distance between v_{pi} and v_{pj}, the projections of training points i and j, d_{ij} is the distance between v_i and v_j, the raw feature vectors, α_{ij} is a weighting factor,

and the sum is over all pairs of training points. We see that σ becomes smaller as the distances between pairs of points in the projected space approximates the distances in the feature vector space.

It can be shown that the loss function σ above is majorized by a quadratic form:

$$\sigma(W) <= \sigma_m(W, V).$$

This majorizing quadratic form has the property that equality holds when $V=W$. The iterative procedure begins by a random choice of W. Then we set $V=W$ to get equality in the equation above. It is possible, by solving a linear system of equations, to choose a new value of W, W', which is guaranteed to reduce the majorizing function σ_m.

Therefore we also know that W' must reduce the value of σ. This completes an iteration; we set $V=W'$, and again solve for the next value of W to reduce the cost function further. New W values are iteratively computed until the cost function does not diminish appreciably between successive iterations.

There are a few aspects of the algorithm which should be mentioned. First, there are several parameters to adjust. First we must select n, the dimension of Φ, which is the number of basis functions that we are summing to produce our projection map. Next we must choose what basis functions to use. If using Gaussians, there are parameters mean μ and variance σ^2 to select. Finally there is the choice of α_{ij}. In [6] is suggested a definition of α_{ij} which incorporates some pre-defined information about the class membership of the training data points; if v_i and v_j are known in advance to belong to the same class, then α_{ij} is defined in such a way that the loss function tends to increase when the distance between the projections, $|v_{pi}-v_{pj}|$, is increased. The loss function becomes a blend of two loss functions, one which measures the degree to which the projection map is not an isometry, and one which measures how spread out the projected classes are. This definition of σ tends to yield a projection map in which the vectors from the same class cluster more tightly.

In a different effort to get such clustering, we experimented with an alternate definition of the cost function:

$$\sigma = \Sigma \alpha_{ij}(q_{ij} - \phi(d_{ij}))^2$$

where the function ϕ is chosen to increase the distance between feature vectors from distinct classes. Specifically, we found good results defining

$$\phi(d_{ij})= \begin{cases} L + d_{ij}, \text{ if i, j from different classes} \\ \sqrt{d_{ij}}, \text{ if i, j from same class,} \end{cases}$$

where L stands for some large value, 1.0 in our experiments.

There is also the issue of computational complexity of the algorithm, which is $O(l^2 n^2)$, where l is the number of training points. For our work, we found that choosing $l=400$ and $n=100$ gave good results in a reasonable amount of time.

4 RESULTS

To find a projection map, we randomly selected, for use in training, 400 feature vectors from the database of 1333. We also randomly selected 100 feature vectors to serve as the means μ_i of the Gaussian basis functions making up the column vector Φ in equation . After finding W, which together with Φ defines the projection map, we projected all 1333 vectors from the database. Our projected vectors were 6 dimensional.

To evaluate the quality of the projection map, each of the database images that was not used in training was used as a query image. For each query, the 10 closest vectors from the database were retrieved. Ideally, the retrieved vectors should belong to the same image class as the query vector. We found that the average correct retrieval percentage considering only those images not used in training was 87%. For comparison, we also evaluated retrieval performance using the full 60 dimensional feature vectors; we found 90% correct retrieval. Thus a factor of 10 reduction in the dimension did not greatly reduce retrieval performance. For further comparison, a traditional linear PCA was used to project the vectors to 6 dimensions. For this linear projection we found only 28% retrieval. The results are summarized in figure 1.
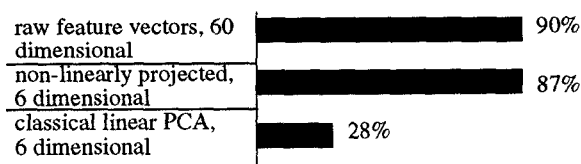
raw feature vectors, 60 dimensional	90%
non-linearly projected, 6 dimensional	87%
classical linear PCA, 6 dimensional	28%

Figure 1: Comparison of retrieval rates.

Three examples of retrieval from the database are shown in figure 2. The images are from an airphoto database. Each airphoto is about 5K x 5K pixels and the data is analyzed for each 64x64 pixel tile. In each example, the query tile is shown first. Then in (a) is shown the top five retrievals using the full 60 dimensional vectors. This can be compared with the top five retrievals using the 6 dimensional vectors resulting from the non-linear PCA, which is shown in (b).

5 DISCUSSIONS

The system for content based retrieval described in [2] has two stages. The first stage matches the feature vector of the query image with one of the code words from a thesaurus, and the second stage then does a nearest neighbor search of the database vectors represented by the codeword. This system provides good retrieval, but increased speed is always desirable. The second stage with nearest neighbor search would be accelerated if the dimension of the feature vectors could be safely reduced. Since our experiments indicate that a ten-fold reduction in feature vector length can still yield good retrieval performance, this is one application that should be investigated.Another possibility is to use the projection on the thesaurus codewords, reducing their size and improving retrieval speed accordingly.

It is also worth investigating the possibility that the non-linear projection map can substitute for the neural network classification stage of the system in [2]. To investigate this possibility, initial experiments were performed on a database of 10299 images from 60 classes. The retrieval percentage for this initial experiment was 41%. This is not high enough to replace the neural network classification. However, this lower retrieval rate may be due to the fact that only 420 vectors were used for training (the complexity of the algorithm in [6] forced this small training set).

Yet another possibility is an interactive, adaptive browsing application. The user might browse the database, specifying by mouse clicks the training images, at the same time supplying the group classification. Then a custom projection map would be computed based on the user's input, and retrieval would be according to the user's tastes. The algorithm in [6] is too complex for such real time browsing on a PC. But simpler algorithms or more powerful computers may be used in such an application. A system using relevance feedback from the user to improve search/retrieval results has been discussed in [7]; some type of multi-dimensional scaling may play a helpful role.

query image: from an orchard region.

1(a)

1(b)

query image: section of a highway.
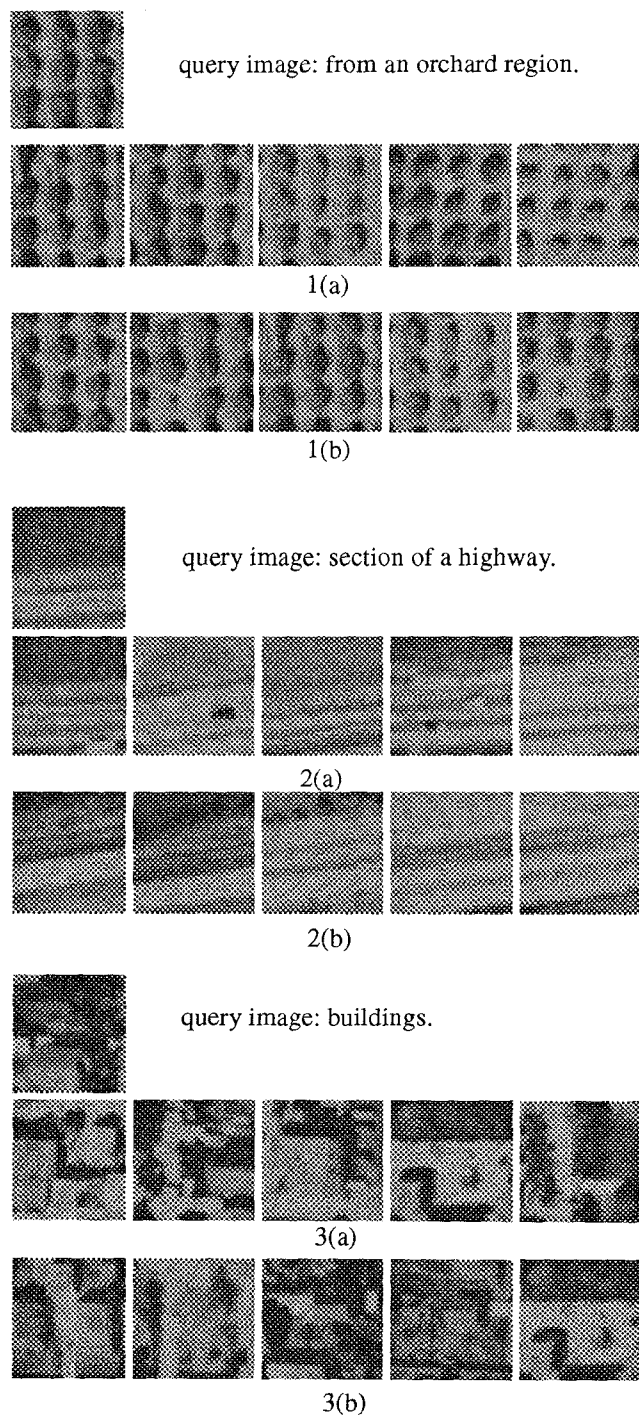
2(a)

2(b)

query image: buildings.

3(a)

3(b)

Figure 2: Examples of image retrievals. Images labeled (a) show the retrieval based upon the raw feature vectors, while the images labeled (b) show retrievals using the non-linear projected space of dimension 6. The difference in retrievals is usually not noticeable, and in some cases there is better retrieval in projected space due to class information used in training.

6 REFERENCES

[1] W.Y. Ma, Y. Deng, and B.S. Manjunath, "Tools for Texture / Color Based Search of Images," *Proc. SPIE Conf. on Human Vision and Electronic Imaging II*, San Jose, CA, Vol. 3106, pp. 496-507, February 1997.

[2] W.Y. Ma, and B.S. Manjunath, "A Texture Thesaurus for Browsing Large Aerial Photographs," to appear in *the Journal of the American Society for Information Science, 1997*.

[3] W.Y. Ma and B.S. Manjunath, "Texture Features for Browsing and Retrieval of Image Data," *IEEE T-PAMI Special Issue on Digital Libraries*, vol. 18(8), august 1996, pp. 837-842

[4] The Alexandria Digital Library Project, http://alexandria.sdc.ucsb.edu.

[5] W.Y. Ma and B.S. Manjunath, "Texture Features and Learning Similarity," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 425-430, June 1996.

[6] Andrew R. Webb, "Multidimensional Scaling by Iterative Majorization Using Radial Basis Functions", *Pattern Recognition*, Vol. 28, No. 5, pp753-759, 1995.

[7] Y. Rui, T. Huang, and S. Mehrotra, *Content based image retrieval with relevance feedback in MARS*, to appear in the Proceed dings of ICIP'97.