Laplacian Eigenmap for Image Retrieval

Xiaofei He *Partha Niyogi*

Department of Computer Science

The University of Chicago, 1100 E 58th Street, Chicago, IL 60637

ABSTRACT

Dimensionality reduction has been received much attention in image retrieval. Even though the dimension of image feature vectors is normally very high, the embedded dimension is much lower. Before we utilize any learning technique, it is beneficial to first perform dimensionality reduction. Principal component analysis is one of the popular methods used, and can be shown to be optimal when the underlying structure of data points is linear. However, it fails to discover the nonlinear structure if the data lie on a low-dimensional manifold embedded in high-dimensional feature space. In this paper, we apply a geometrically motivated algorithm for representing the high dimensional data, which has locality preserving properties. Consequently, a novel relevance feedback scheme on manifold is proposed. Experiment results on real-world image collections have shown the effectiveness and robustness of our proposed approaches.

Keywords: image retrieval, laplacian eigenmap, dimensionality reduction, relevance feedback

1. Introduction

Due to the rapid growth in the volume of digit images, there is an increasing demand for effective image management tools. Consequently, content-based image retrieval (CBIR) is receiving widespread research interest (Cox et al., 1996; Smith and Chang, 1996; Ma and Manjunath, 1997; Rui et al., 1997; Tong and Chang, 2001; Su et al, 2001). Content based image retrieval uses features automatically extracted from the images themselves, rather than manually provided annotations, to facilitate the retrieval of images relevant to a user's query. However, there are still many open research issues to be solved before such retrieval system can be put into practice.

In recent years, much research has been done to deal with the problem caused by the high dimensionality of image feature space (Tenenbaum et al., 2000; Roweis and Saul, 2000). Typically, the dimensions of feature vector range from few tens to several hundreds. For example, a color histogram may contain 256 bins. High dimensionality creates several problems for image retrieval. First, learning from examples is computationally infeasible if it has to rely on high-dimensional representations. The reason for this is known as *curse of dimension*: the number of examples necessary for reliable generalization grows exponentially with the number of dimensions (Stone, 1982). Learnability thus necessitates dimensionality reduction. Second, in large multimedia databases, high-dimensional representation is computational intensive and most users do not wait around to provide a great deal of feedbacks. Hence for storage and speed concern, dimensionality reduction is needed.

Two classical techniques for dimensionality reduction are Principle Component Analysis (PCA) and Multidimensional Scaling (MDS). PCA performs dimensionality reduction by projecting the original n-dimensional data onto the $m(<n)$ dimensional linear subspace spanned by the leading eigenvectors of the data's covariance matrix. Thus PCA builds a global linear model of the data (an m dimensional hyperplane).

For linearly embedded manifolds, PCA is guaranteed to discover the dimensionality of the manifold and produce a compact representation in the form of an orthonormal basis. However, PCA fails to discover the underlying structure, if the data lie on a nonlinear submanifold of the feature space. For example, the covariance matrix of data sampled from a helix in R^3 will have full-rank and thus three principle components. However, the helix is a one-dimensional manifold and can be parameterized with a single number. Classical MDS finds an embedding that preserves pairwise distances between data points. It is equivalent to PCA when those distances are Euclidean.

A key question in image retrieval is that, *how do we judge similarity*? The choice of the similarity measure for the inputs is a deep question that lies at the core of the field of machine learning. If the images lie on or close to a low-dimensional nonlinear manifold embedded in high-dimensional feature space, then the Euclidean distance in high-dimensional feature space may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold. In addition, due to the gap between semantic concepts and low-level image features, the global topology of images can not match human perception. Thus, local topology of images is much more important and reliable. When two images are irrelevant, the absolute value of their distance in feature space gives us little information about their similarity. For example, it is meaningless to say that, a tiger is more like a dog than a horse. In this paper, we apply a simple nonlinear dimensionality reduction algorithm, which has locality preserving properties. A linear dimensionality reduction algorithm is proposed as its linear extension. Consequently, a geometrically motivated relevance feedback scheme is proposed for image ranking, which is conducted on the image manifold, rather than Euclidean space. To model the geodesic paths of the image manifold, we find the shortest-path distances between images in the database, which is in turn used as similarity measure. Our goal is to discover, given only the unordered high-dimensional inputs, low-dimensional representations that capture the intrinsic degrees of freedom of an image set.

The rest of this paper is organized as follows. Section 2 relates a list of previous works to our work. Section 3 describes laplacian eigenmap for nonlinear dimensionality reduction. Section 4 describes its connections to principle component analysis. In section 5, we propose a linear dimensionality reduction algorithm as linear extensions of laplacian eigenmap. Section 6 describes the proposed method for relevance feedback on manifold. The experimental results are shown in section 7. Finally, we give conclusions and future work in section 8.

2. Previous Work

One of the most popular models used in information retrieval is the vector space model (Salton and McGill, 1983). Various retrieval techniques have been developed for this model, including the method of relevance feedback. Most previous researches on relevance feedback have fallen into the following three categories: retrieval based on query point movement (Rui et al., 1997), retrieval based on re-weighting of different feature dimension (Ishikawa et al., 1998) and retrieval based on updating the probability distribution of images in the database (Cox et al., 2000).

In recent years, some learning-based approaches are proposed. Wu et al. (2000) proposed a Discriminant-EM algorithm within the transductive learning framework in which both labeled and unlabeled images are used. Tieu et. al (2000) presented a framework for image retrieval based on representing images with a very large set of highly selective features. Queries are interactively learned online with a simple boosting algorithm. Tong et. al (2001) proposed the use of a support vector machine active learning algorithm for conducting effective relevance feedback for image retrieval. While most machine learning algorithms are passive in the sense that they are generally applied using a randomly selected training set, the SVM active learning algorithm chooses the most informative images within the database to ask the user to label. All these approaches have achieved good empirical results. However, a common limitation of them is that they do not consider the underlying structure of image set in high-dimensional feature space.

There have been several studies of dimensionality reduction for image retrieval. In (Su et al., 2001), principle component analysis is used to reduce both noise contained in the original image features and dimension of feature space. The PCA process is incorporated into the relevance feedback framework to extract feature subspaces in order to represent the subjective class implied in the positive examples. Different types of features (color, texture, etc.) are allowed to have different dimensions according their significances and distributions as implied in user's feedbacks. Cohen et al. (2002) proposes a novel method, called principle feature analysis, for dimension reduction of a feature set by choosing a subset of the original features that contains most of the essential information, using the same criteria as the PCA. In (Wu, 2000), weighted multi-dimensional scaling is used for dimensionality reduction. All of these methods do not consider the structure of the image manifold on which the images are likely to reside.

3. Laplacian Eigenmap for Image Representation

Recently, there has been some renewed interest in the problem of developing low dimensional representations when data lies on a manifold (Tenenbaum et al., 2000; Roweis and Saul, 2000). In image retrieval, it is desirable to map the image feature vectors into a reduced space, due to the consideration of learnability, memory storage, and computational speed. In addition, due to the gap between semantic concepts and low-level image features, the global topology of images can not always match human perception. Thus, local topology of images is much more important and reliable. If the distance of two images is large enough, we regard them as irrelevant. When two images are irrelevant, the absolute value of their distance gives us little information about their similarity. For example, it is meaningless to say that, a tiger is more like a dog than a horse. Thus, for image retrieval, a criterion should be satisfied that, the mapping must have locality preserving properties. That is, the nearest neighbors of an image in the original feature space should be mapped to nearest neighbors of that image in the reduced space.

Recall that principal component analysis (PCA) is a linear mapping algorithm which maximizes the variance. Let $\mathbf{y} = \{y_1, y_2, \cdots, y_n\}^T$ be a one-dimensional map. The objective function of PCA is as follows

$$\max_y \sum_i (y_i - \overline{y})^2$$

where $\overline{y}$ is the sample mean.

In this paper, we apply laplacian eigenmap (Belkin and Niyogi, 2001) to map the images into a low-dimensional space. Consider the problem of mapping the weighted graph G to a line so that connected points stay as close as possible. Let $\mathbf{y} = \{y_1, y_2, \cdots, y_n\}^T$ be such a map. A reasonable criterion for choosing a "good" map is to minimize the following objective function

$$\sum_{ij} (y_i - y_j)^2 W_{ij}$$

The minimization problem reduces to finding

$$\underset{\substack{y \\ y^T Dy=1}}{\arg\min}\, y^T Ly$$

where $L = D - W$ is the Laplacian matrix. D is diagonal weight matrix; its entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ji}$. Laplacian is symmetric, positive semidefinite matrix which can be thought of as an operator on functions defined on vertices of G. W is the weight matrix such that

$$W_{ij} = \begin{cases} 1 & \text{if points } \mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{j} \text{ are connected} \\ 0 & \text{otherwise} \end{cases}$$

The objective function with our choice of weights W_{ij} incurs a heavy penalty if neighboring points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are mapped far apart. Therefore, minimizing it is an attempt to ensure that if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are "close" then y_i and y_j are close as well.

The algorithmic procedure is formally stated below:

Step 1 [Constructing the adjacency graph] Nodes i and j are connected by an edge if i is among k nearest neighbors of j or j is among k nearest neighbors of i.

Step 2 [Choosing the weights] $W_{ij} = 1$ if and only if vertices i and j are connected by an edge.

Step 3 [Eigenmaps] Assume the graph G, constructed above, is connected, otherwise proceed with Step 3 for each connected component. Compute eigenvalues and eigenvectors for the generalized eigenvector problem:

$$L\mathbf{y} = \lambda D\mathbf{y} \tag{1}$$

Let $\mathbf{y}_0, ..., \mathbf{y}_{k-1}$ be the solutions of equation 1, ordered according to their eigenvalues,

$$\begin{aligned}
&L\mathbf{y}_0 = \lambda_0 D\mathbf{y}_0 \\
&L\mathbf{y}_1 = \lambda_1 D\mathbf{y}_1 \\
&\cdots\cdots \\
&L\mathbf{y}_{k-1} = \lambda_{k-1} D\mathbf{y}_{k-1} \\
&0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{k-1}
\end{aligned}$$

We leave out the eigenvector $\mathbf{y}_0$ corresponding to eigenvalue 0 and use the next *m* eigenvectors for embedding in *m*-dimensional Euclidean space.

$$\mathbf{x}_i \to (\mathbf{y}_1(i), \cdots, \mathbf{y}_m(i))$$

4. Connections to Principal component analysis (PCA)

4.1 Principal component analysis

Principal component analysis is a widely used statistical tool for data analysis. Given a set of feature vectors in high-dimensional space, the purpose is to find a low-dimensional mapping in reduced space with less redundancy, that would give as good a representation as possible. The redundancy is measured by correlations between data elements.

In the PCA transform, the feature vector **x** is first centered by substracting its mean:

$$\mathbf{x} \leftarrow \mathbf{x} - E(\mathbf{x})$$

The mean is estimated from the available samples $\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(m) \in R^n$. Let us assume in the following that the centering has been done and thus *E(x)=0*. Let **X** denote the feature vector matrix, that is, the i^{th} row of **X** is the feature vector **x**(i). So, **X** is a $m \times n$ matrix. Thus, we can obtain the covariance matrix as follows:

$$C = \mathbf{X}^T \mathbf{X}$$

By eigenvector decomposition, *C* can be decomposed into the product of three matrices:

$$C = V \Sigma V^T$$

where $\Sigma = diag\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ are the eigenvalues with descending order and *V* is a orthogonal matrix, whose column vector is the corresponding eigenvector of *C*. Thus, $T_{PCA} = V$ is the transformation matrix of PCA. If we want to map these *m*-dimensional feature vectors to a *k*-dimensional *(k<m)* space, it could be done by simply set the *m-k* least eigenvalues to be zero, i.e., $\lambda_{k+1} \leftarrow 0, \cdots, \lambda_m \leftarrow 0$.

By applying Singular Value Decomposition (SVD), we can reformulate this problem as follows:

$$\mathbf{X} = U \Sigma V^T$$

where *U* and *V* are two orthogonal matrices, whose column vectors are the eigenvector of $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T \mathbf{X}$ (the covariance matrix *C*), respectively, and $\Sigma = diag\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ are the singular values of **X** (also the eigenvalues of $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T \mathbf{X}$). Now, we obtain the transformed feature vector in new low-dimensional space:

$$\mathbf{X}' = \mathbf{X} T_{PCA} = U \Sigma$$

This indicates that the low-dimensional representation obtained by PCA is just the product of two matrices: (1) the matrix *U* whose column vectors are the leading *eigenvectors* of weight matrix $W_{PCA} = \mathbf{X}\mathbf{X}^T$:

$$W_{PCA} \mathbf{y} = \lambda_{PCA} \mathbf{y} \qquad (2)$$

$$U = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\}$$

and (2) the diagonal matrix Σ whose diagonal elements are the leading *eigenvalues* of W_{PCA} , $\Sigma = diag\{\lambda_1, \lambda_2, \cdots, \lambda_k\}$. Note that the weight matrix W_{PCA} is measured by inner product of two image vectors.

4.2 Connections to PCA

In section 4, we have show that the optimal embedding for non-linear case is obtained by solving a general eigenvector problem below:

$$
\begin{aligned}
& L\mathbf{y} = \lambda D\mathbf{y} \\
\Rightarrow\ & (D - W)\mathbf{y} = \lambda D\mathbf{y} \\
\Rightarrow\ & W\mathbf{y} = (1 - \lambda) D\mathbf{y} \\
\Rightarrow\ & D^{-1}W\mathbf{y} = (1 - \lambda)\mathbf{y}
\end{aligned}
$$

Let $W_{Laplacian} = D^{-1}W$, $W_{Laplacian}$ is essentially a normalized weight matrix preserving locality. (Note that the matrix D provides a natural measure on the vertices of the graph. The bigger the value D_{ii} is, the more "important" is that vertex). We rewrite the above formula as follows:

$$W_{Laplacian}\mathbf{y} = (1 - \lambda_{Laplacian})\mathbf{y} \qquad (3)$$

As we can see, the equations (2) and (3) have the same form. This observation shows that, with the same weight matrix, Laplacian Eigenmap and PCA will yield the same result, and $\lambda_{PCA} + \lambda_{Laplacian} = 1$. Therefore, the eigenvector of W_{PCA} with the largest eigenvalue is just the eigenvector of $W_{Laplacian}$ with the smallest eigenvalue.

The connections also give us a way to determine the dimensionality of the low-dimensional manifold, analogous to PCA.

$$J(k) = \frac{\sum_{i=1}^{k}(1 - \lambda_i)}{\sum_{i=1}^{rank(W)-1}(1 - \lambda_i)} \times 100\%$$

where $0=\lambda_0<\lambda_1\le\lambda_2\le\cdots\le\lambda_{rank(W)-1}$. If we want to keep 90% information, we simply choose such k that $J(k)\approx 90\%$.

Based on above analysis, we conclude that, the essential difference between laplacian eigenmap and PCA is that they choose different weight matrices. PCA uses inner product as a linear similarity measure, while laplacian eigenmap uses a non-linear similarity measure which preserves locality. For PCA, its advantage over laplacian eigenmap is that it can produce a transform matrix T_{PCA}. Thus, for a new point, it can be easily map to the new space. The disadvantage is that, it fails to discover the underlying nonlinear structure of data set.

5. Linear Laplacian Eigenmap

As we described previously, one disadvantage of laplacian eigenmap is that it cannot produce a transformation function. That is, if the query image is not in database, there is no way to map it into the reduced space. To overcome this problem, we propose linear laplacian eigenmap, which is a linear approximation of laplacian eigenmap.

Suppose $A=(\boldsymbol{a}_1,\boldsymbol{a}_2,\ldots,\boldsymbol{a}_k)$ is a transformation matrix of linear laplacian eigenmap, that is, $Y=XA$, where the row vectors of matrix Y are the image representations in reduced space. We rewrite the minimization problem as follows:

$$\underset{\substack{A\\(XA)^T D(XA)=I}}{\arg\min}\quad (XA)^T L(XA)$$

The vectors $\boldsymbol{a}_i$ that minimize the above function is given by the minimum eigenvalue solutions to the generalized eigenvalue problem:

$$L'\mathbf{a}=\lambda D'\mathbf{a}$$

where $L'=X^T LX$, $D'=X^T DX$.

Note that L' and D' are two $n \times n$ matrices, where n is the dimensionality of the original space. While in the nonlinear case, L and D are two $m \times m$ matrices, where m is the number of images in database. In the real world applications, $m >> n$. This property makes linear laplacian eigenmap much more promising than nonlinear laplacian eigenmap.

6. Relevance Feedback on Image Manifold

An image retrieval system should be able to rank the images according to the query image and relevance feedbacks. Traditionally, the most widely used distance function is Euclidean distance. This is based on an assumption that the images lie on a linear high-dimensional space. However, Euclidean distance is not always a good choice, if the images lie on or close to a nonlinear manifold with low-dimensionality. Following laplacian eigenmap discussed in section 3 and linear laplacian eigenmap described in section 5, we propose a geometrically motivated relevance feedback scheme for image ranking, which is conducted on manifold, rather than Euclidean space.

Let R denote the set of query image and relevant images provided by the user. Our algorithm can be described as follows:

1. **Candidate generation.** For each image $x_i \in R$, we find its k-nearest neighbors $C_i = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\}$ (those images in R are excluded from selection). Let $C = C_1 \cup C_2 \cup \cdots \cup C_{|R|}$. We call C candidate image set. Note that $R \cap C = \phi$.

2. **Construct subgraph.** We build a subgraph H(V) with weight matrix W_H, where $V = R \cup C$ .The distance of any two vertices x_i, x_j is measured as follows:

$$dist(x_i, x_j) = \begin{cases} \left\| x_i - x_j \right\|_2 & \text{if } x_i \text{ is among } x_j\text{'s } k\text{' neareset neighbor,} \\ & \text{or } x_j \text{ is among } x_i\text{'s } k\text{' neareset neighbor} \\ \infty & \text{otherwise} \end{cases}$$

Since the images in R are supposed to have some common semantics, we set their distances to be zero. That is, $dist(x_i, x_j) = 0, \forall x_i, x_j \in R$. Note that, *k'<k*.

3. **Manifold distance measure.** To model the geodesic paths of the manifold, we find the shortest-path distances between vertices in H. The length of a path in H is defined to be the sum of link weights along that path. We then compute the geodesic distance $dist_H(x_i, x_j)$ (i.e. shortest path length) between all pairs of vertices of *i* and *j* in H, using Floyd's $O(r^3)$ algorithm.

4. **Retrieval based on geodesic distance.** To retrieve the images most similar to the query, we simply sort them according to their geodesic distance to the query. The selected images are presented to the user.

5. **Update query example set.** Add the relevant images provided by the user into R. Go back to step 1 until the user is satisfied.

From the above algorithm, we can see that the similarity between the image in database and the query image is determined by their geodesic distance on image manifold. The query examples (including the original query image and relevant images provided by the user) are merged into one point on the manifold. Thus, the geodesic distance between query image *q* and image **x** can also be computed as follows:

$$dist_H(\mathbf{q}, \mathbf{x}) = \min_{\mathbf{r} \in R}\{dist_C(\mathbf{r}, \mathbf{x})\}$$

where $dist_C(\mathbf{r}, \mathbf{x})$ is the length of the shortest path in *C* between image **r** and image **x**. That is, the vertices in the shortest path can only contain vertices in *C*. (Note that, $dist_H(\mathbf{q}, \mathbf{r}) = 0, \mathbf{r} \in R$) Thus, our algorithm is essentially different from Rocchio's relevance feedback scheme (1971).

Rocchio's formula for query refinement can be illustrated as follows:

$$Q' = \alpha Q_0 + \beta \sum_{i=1}^{n_1} \frac{R_i}{n_1} - \gamma \sum_{i=1}^{n_2} \frac{S_i}{n_2}$$

where Q_0 is the initial query vector, Q' is the refined query vector, R_i is the vector of relevant examples, S_i is the vector of irrelevant examples, n_1 and n_2 are the number of relevant examples and irrelevant examples, respectively; β and γ are suitable constants. Rocchio's algorithm is essentially a perceptron-like learning algorithm when the inner product similarity is used. This technique is implemented in the MARS system (Rui et al., 1997).

Rocchio's similarity-based relevance feedback algorithm has been widely used in information retrieval, and has archived good empirical results in text retrieval domain. However, it may not be suitable for image retrieval, since it fails to preserve locality. Due to the gap between semantic concepts and low-level image features, the global topology of images can not match human perception. Thus, local distribution of images is much more important and reliable. Hence the local information should be preserved while retrieval. Figure 1 illustrate a simple example: A stands for the initial query image; B is a relevant example provided by the user. C is the refined query by Rocchio's algorithm ($\alpha = \beta = 0.5$). Thus, image D will be retrieved. However, by intuition, images E and F are preferred to image D. In our proposed algorithm, A and B are merged into a single point. Thus, the geodesic distance between A(B) and D is larger than the geodesic distance between A(B) and E(F). So image E and F will be retrieved, rather than image D.

7. Experimental Results

We performed several experiments to evaluate the effectiveness of the proposed approaches over a large image database. The database we use consists of 10,000 images of 100 semantic categories from the Corel dataset. It is a large and heterogeneous image set. A retrieved image is considered correct if it belongs to the same category of the query image. Three types of color features and three types of texture features are used in our system, which are listed in Table 1. The dimension of the feature space is 435.

We designed an automatic feedback scheme to model the retrieval process. At each iteration, the system makes the first three incorrect images from the top 100 matches as irrelevant examples, and also selects at most 3 correct images as relevant examples (relevant examples in the previous iterations are excluded from the selection). These automatic generated feedbacks are added into the query example set to refine the retrieval. To evaluate the performance of our algorithms, we define the retrieval accuracy as follows:

$$Accuray = \frac{\text{relevant images retrieved in top N returns}}{N}$$

7.1 Three examples for 2-D data visualization

To compare the performance of laplacian eigenmap, linear laplacian eigenmap and PCA, we show two examples. In the first example, the dataset contains two classes. One class contains image data of human eyes, and the other class contains image data of non-human eyes. Each class contains 1500 images in 400-dimensional space. Each image is an 8 bits (256 grey) image of size 20×20. In the second example, a multi-class dataset of handwritten digits (‘0’-‘9’) (Blake and Merz 1998). is used. The experimental results are shown in figure 4-5.

7.2 Image Retrieval in 2-dimensional Reduced Space

The image database we use for this experiment consists of 10,000 images of 100 semantic categories from the Corel dataset. Each semantic category consists of 100 images. We compare the retrieval result in PCA space, kernel PCA space, laplacian space and linear laplacian space with 2 dimensions. In PCA space, the traditional Rocchio’s relevance feedback scheme is used. In laplacian space and linear laplacian space, our relevance feedback scheme on manifold is used. Fig 2 shows the experiment results. As can be seen, our proposed relevance feedback scheme in laplacian space performs much better than Rocchio’s relevance feedback scheme in PCA space.

7.3 Image Retrieval in Reduced Space with different dimensions

In this section, we compared the performance of image retrieval in reduced space with different dimensions. Figure 3 shows the experiment results. As the feature space is drastically reduced by PCA, the performance decreases fast. In PCA space with 2 dimensions, only about 10% accuracy is achieved after three iterations. While in 2-dimensional laplacian space, we can still achieve more than 40% accuracy after *one* iteration. This observation shows that, our proposed algorithm is especially suitable for the case when drastic dimensionality reduction needs to be performed.

8. Conclusions and Future Work

While nonlinear dimensionality reduction on manifold has received much attention, to the best of our knowledge we are not aware of any application to image representation and retrieval yet. In this paper, we described a nonlinear dimensionality reduction algorithm and its linear extension for image retrieval. The proposed scheme infers a reduced space that preserves locality. A corresponding relevance feedback scheme is proposed. It is conducted on the image manifold, rather than Euclidean space. As can be seen from the experiments, our proposed algorithm performs much better than principal component analysis, especially when drastic dimensionality reduction is performed.

A fundamental problem in image retrieval is the gap between high-level semantic concepts and low-level image features. Most existing relevance feedback techniques focus on improving the retrieval performance of current query session, and the knowledge obtained form the past user interactions with the system is forgotten. A possible extension of our work is to somehow adjust the local topology of images from user's relevance feedback, so the system will gradually improve its retrieval performance through accumulated user interactions. We are currently exploring the effect and impact of this extension.

References

[1] Belkin, Mikhail and Niyogi, Partha. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. *NIPS*, 2001.

[2] Blake, C.L. and Merz, C.J. (1998). UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[3] Cohen, I., Tian, Q., Zhou, X. S., and Huang, T. S. 2002. Feature selection using principal feature analysis, submitted to *ICIP*'02.

[4] Cox, I.J., Minka, T.P., Papathomas, T.V., and Yianilos, P.N. 2000. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiemtns, *IEEE Trans. On Image Processing – Special Issue on Digital Libraries*, pp. 20-37, vol. 9, No. 1.

[5] Ishikawa, Y., Subramanya, R. and Faloutsos, C. 1998. MindReader: query databases through multiple examples". In *proc. of the 24th Int. Conf. on Very Large DataBases*, pp. 218-227.

[6] Ma, W.Y. and Manjunath, B.S. 1997. Netra: A toolbox for navigating large image databases. In *Proc. IEEE International Conference on Image Processing*, Vol. 1, pp 568-571.

[7] Rocchio, J. J., 1971. Relevance Feedback in Information Retrieval. In the *Smart System – Experiments in Automatic Document Processing*, Englewood Cliffs, NJ, Prentice Hall, pp. 313-323.

[8] Roweis, S. T. and Sual, L. K. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, vol 290, 22.

[9] Rui, Y., Huang, T.S., Mehrotra, S., and Ortega, M. 1997. A relevance feedback architecture for content-based multimedia information retrieval systems. In *Proc. IEEE workshop on content-based access of image and video libraries*, pp. 82-89.

[10] Rui, Y., Huang, T. S., and Mehrotra, S. 1997. Content-Based Image Retrieval with Relevance Feedback in MARS, in *Proc. IEEE Int. Conf. on Image Processing*.

[11] Salton, G. and McGill, M. 1983. *Introduction to modern information retrieval*. McGraw-Hill, New York.

[12] Smith, J.R. and Chang, S.-F., 1996. VisualSEEk: A fully automated Content-based Image Query System. *ACM Multimedia International Conference*, Boston, MA.

[13] Stone, C. J. 1982. Optimal global rates of convergence for nonparametric regression, *Annals of statistics*, 10:1040-1053.

[14] Su, Zhong, Li, Stan, and Zhang, H. 2000. Extraction feature subspaces for content-based retrieval using relevance feedback, *ACM Multimedia*.

[15] Tenenbaum, J. B., Silva, V., and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction, *Science*, vol 290, 22.

[16] Tieu, K. and Viola, P. 2000. Boosting image retrieval", in *Proc. IEEE Conf. CVPR*, pp. 228-235.

[17] Tong, S. and Chang, E. 2001. Support vector machine active learning for image retrieval, in *Proc. ACM Multimedia*.

[18] Wu. P., Manjunath., B.S., and Shin, H.D. 2000. Dimensionality Reduction for Image Retrieval. In *Proc. ICIP'00*.

[19] Wu, Y., Tian, Q., and Huang, T.S., 2000. Discriminant-EM algorithm with application to image retrieval, in *Proc. IEEE Conf. CVPR*.

Color-1	Color histogram in HSV space with quantization 256
Color-2	First and second moments in Lab space
Color-3	Color coherence vector in LUV space with quantization 64
Texture-1	Tamura coarseness histogram
Texture-2	Tamura directionary
Texture-3	Pyramid wavelet texture feature

Table 1. Image features

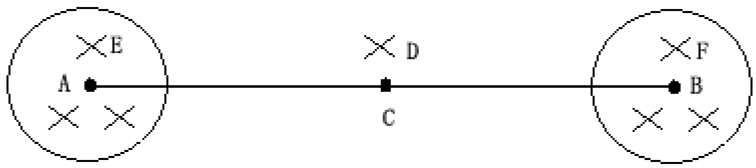
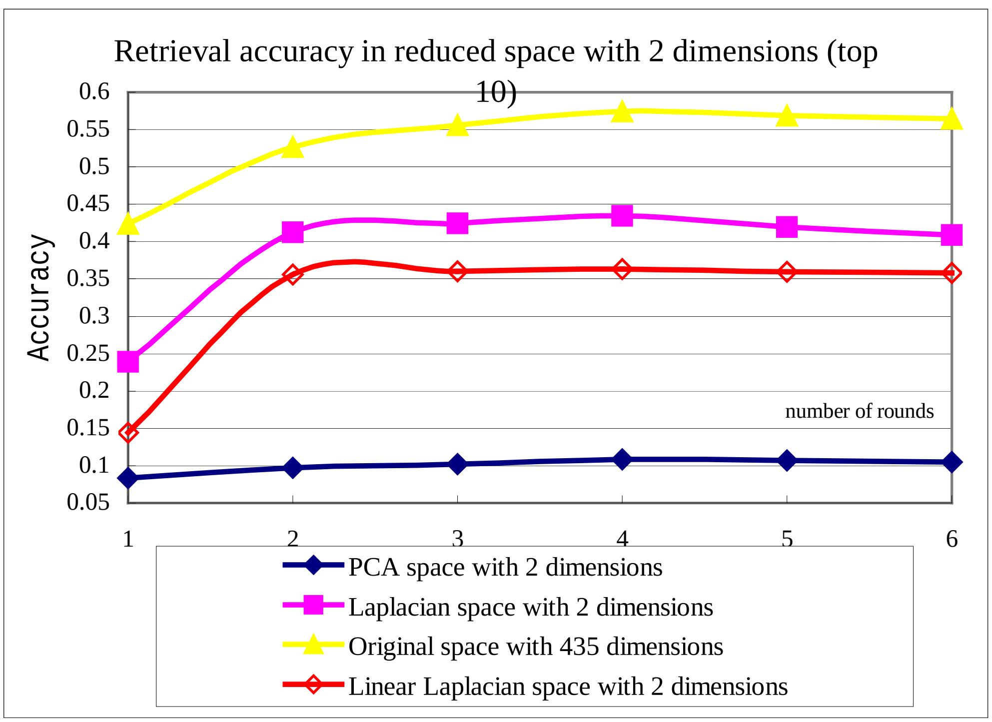


Figure 1. A case where Rocchio's relevance feedback gives a bad result. Image A is the initial query example. Image B is a relevant example. Image C is the refined query. By Rocchio's relevance feedback scheme, image D will be retrieved, rather than E or F.



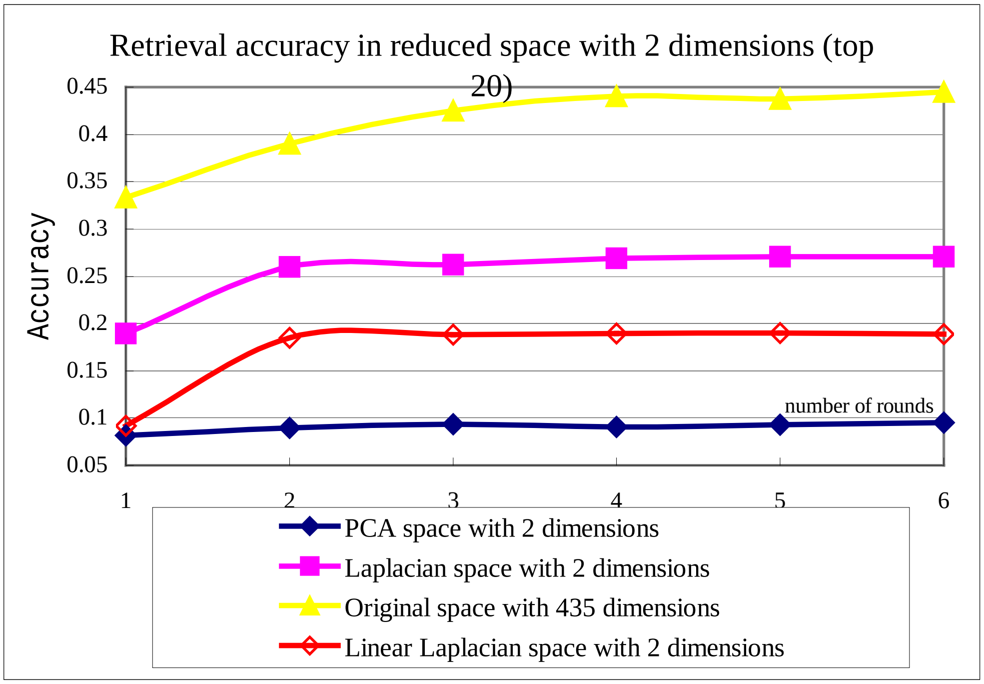
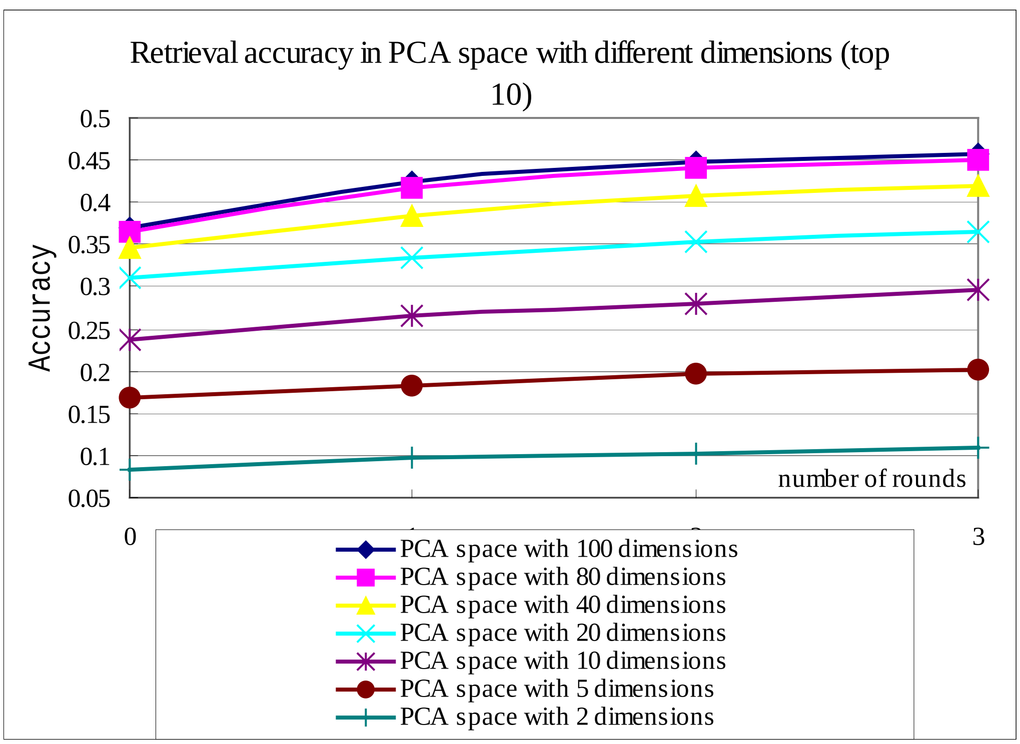
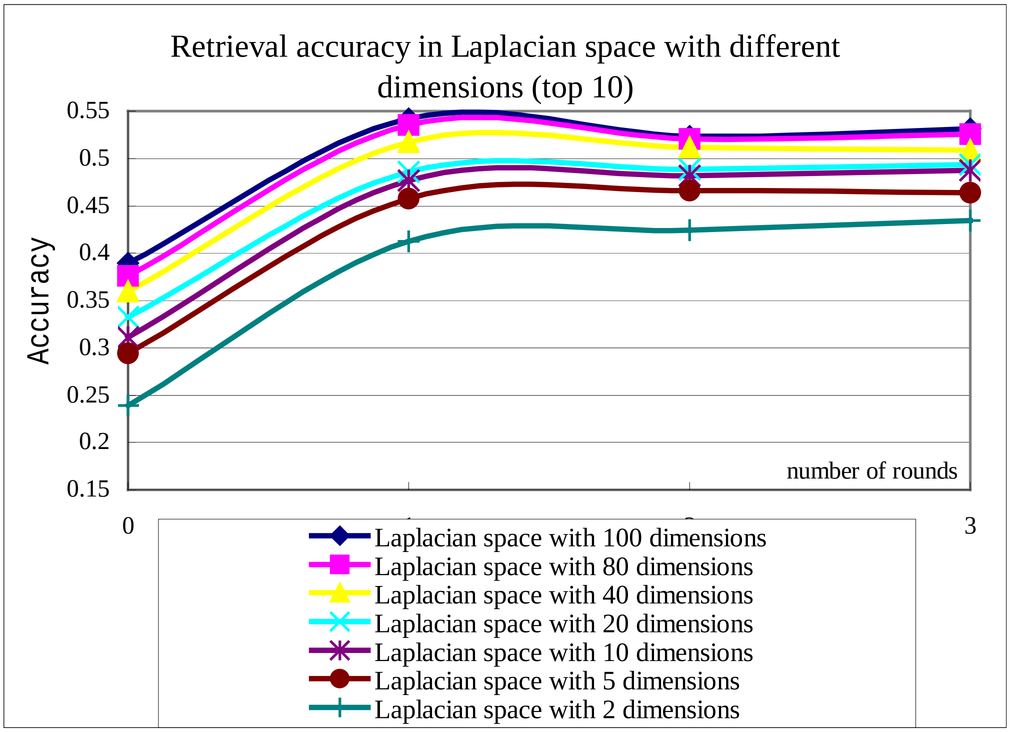


Figure 2. Image retrieval in 2-dimensional reduced space. Rocchio's relevance feedback scheme is used in PCA space, while our proposed algorithm is used in laplacian space, linear laplacian space, and original space.





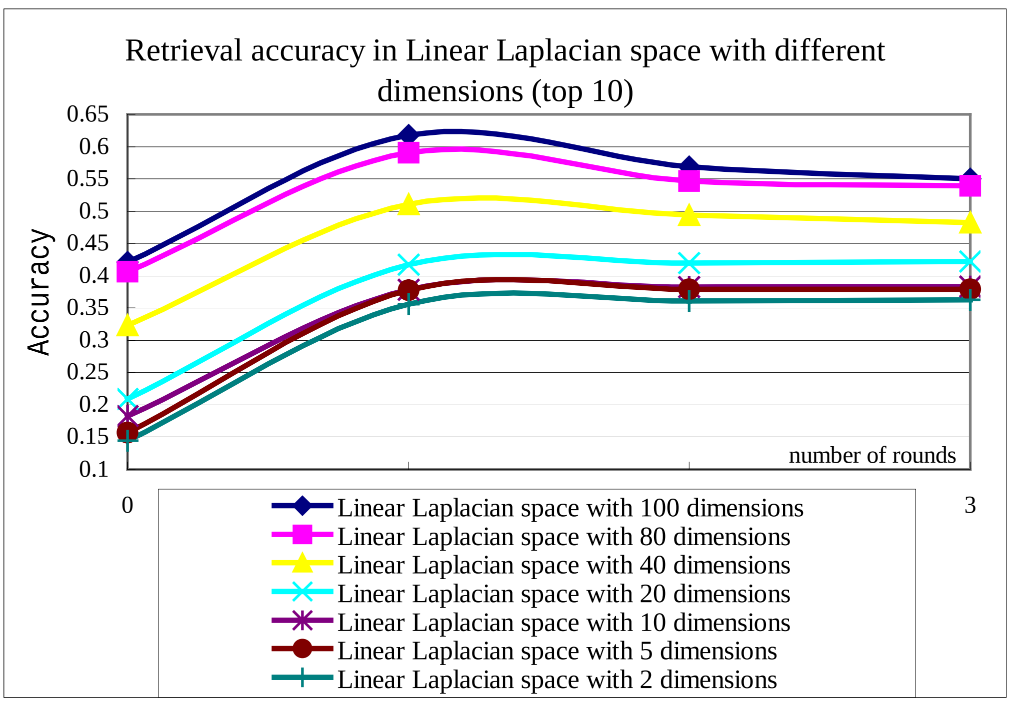
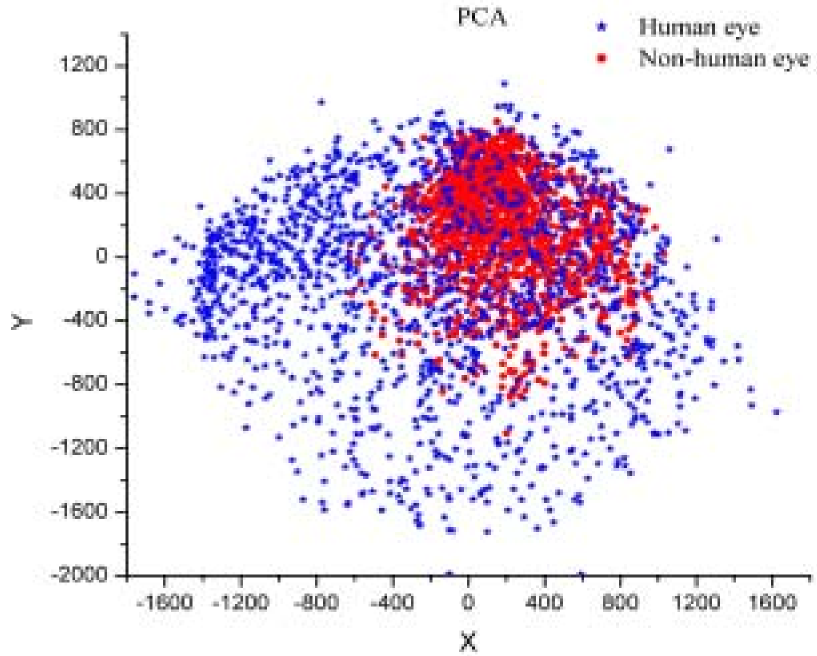
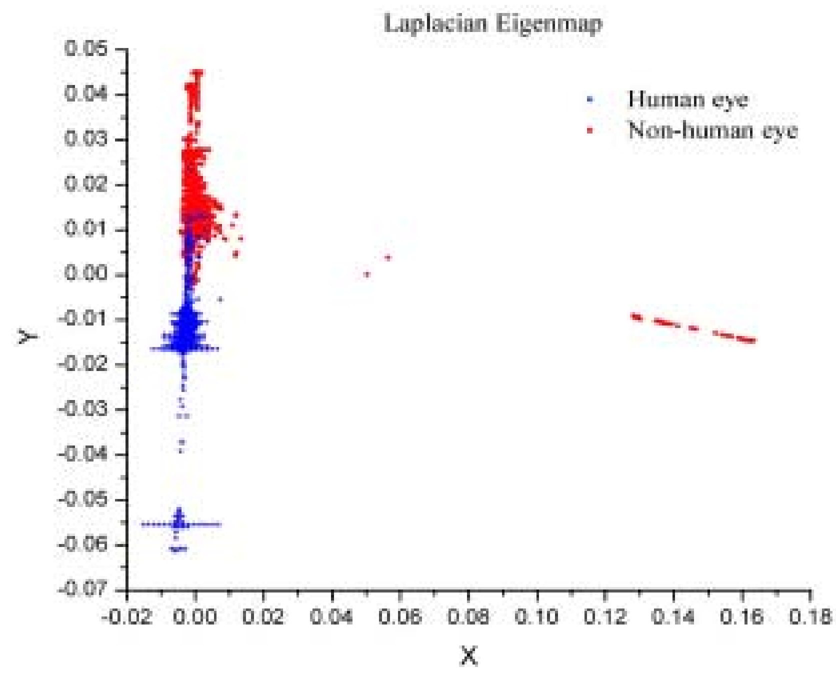


Figure 3. Image retrieval in reduced space with different dimensions. Rocchio's relevance feedback scheme is used in PCA space, while our proposed algorithm is used in laplacian space and linear laplacian space.


PCA
Human eye
Non-human eye
1200
800
400
0
-400
-800
-1200
-1600
-2000
Y
-1600
-1200
-800
-400
0
400
800
1200
1600
X


Laplacian Eigenmap
Human eye
Non-human eye
0.05
0.04
0.03
0.02
0.01
0.00
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
Y
-0.02
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
X

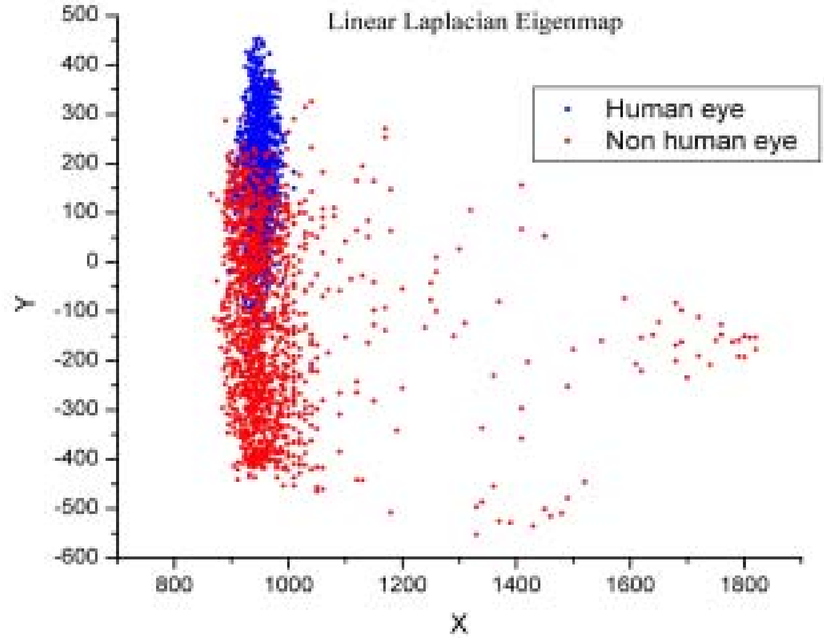
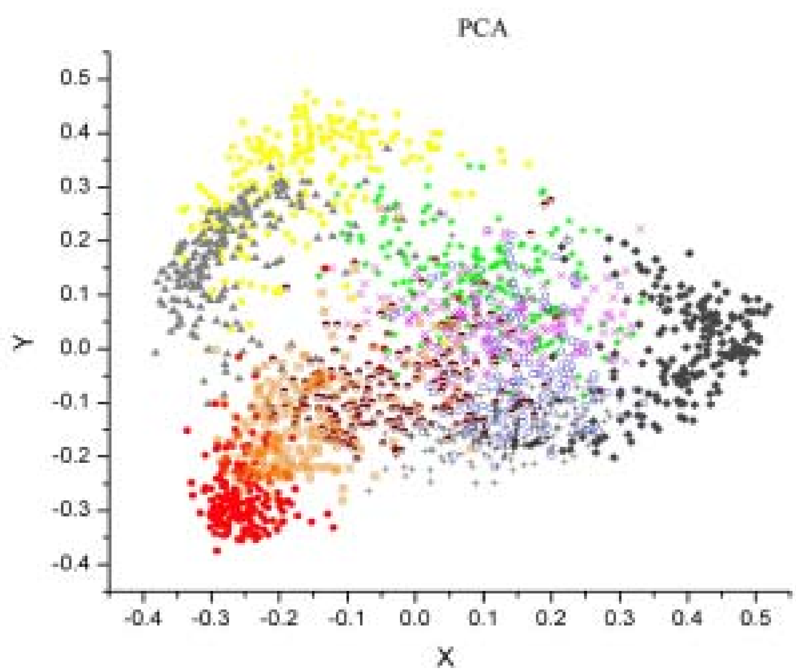


Figure 4. Laplacian eigenmap vs. PCA. (Two classes)



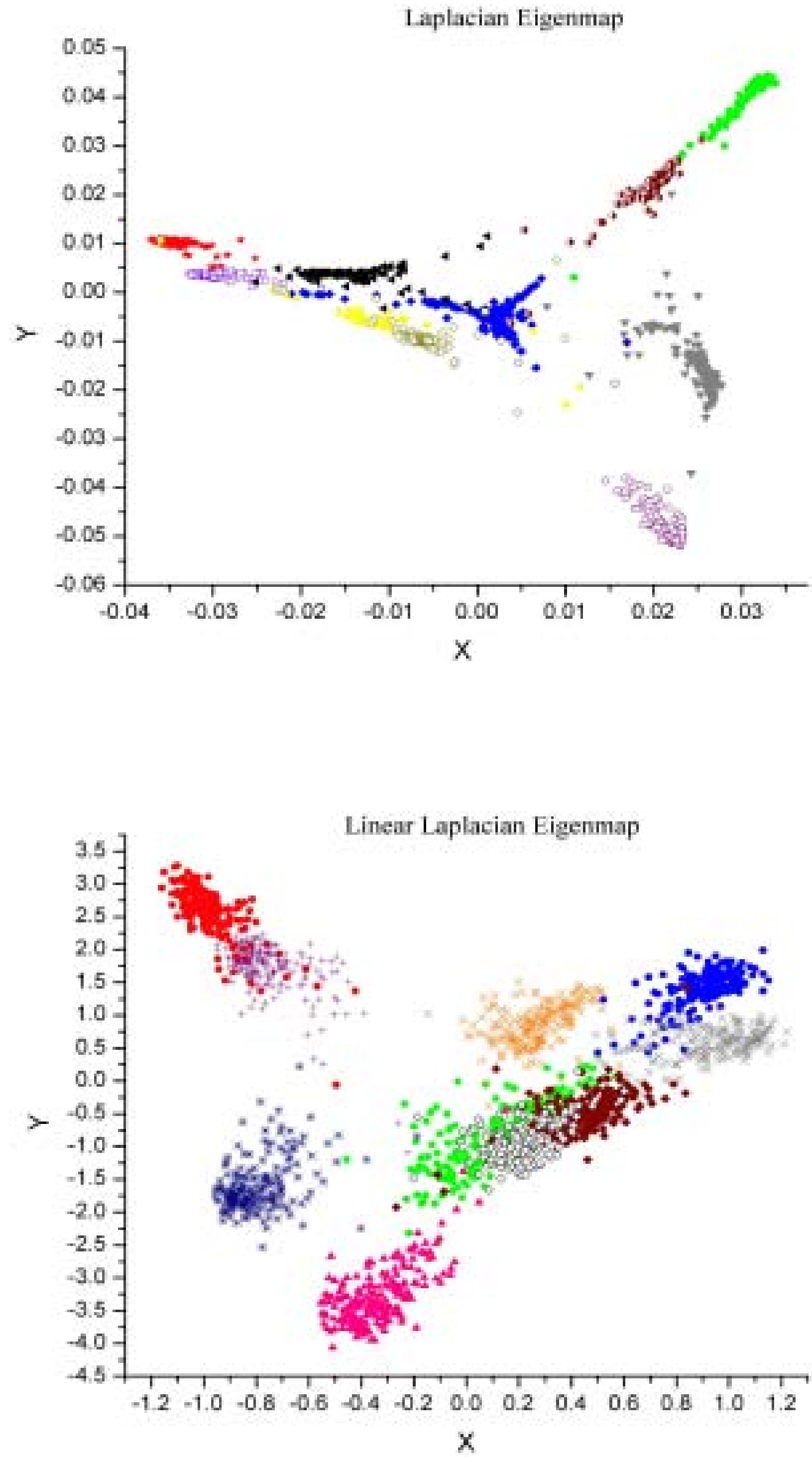


Figure 5. Laplacian eigenmap vs. PCA (10 classes, digits ‘0’-‘9’)