Machine Vision and Applications
DOI 10.1007/s00138-008-0169-4

SHORT PAPER

Neighborhood linear embedding for intrinsic structure discovery

Shuzhi Sam Ge · Feng Guan · Yaozhang Pan · Ai Poh Loh

Received: 30 August 2006 / Accepted: 10 September 2008

Abstract In this paper, an unsupervised learning algorithm, neighborhood linear embedding (NLE), is proposed to discover the intrinsic structures such as neighborhood relationships, global distributions and clustering property of a given set of input data. This algorithm eases the process of intrinsic structure discovery by avoiding the trial and error operations for neighbor selection, and at the same time, allows the discovery to adapt to the characteristics of the input data. In addition, it is able to explore different intrinsic structures of data simultaneously, and the discovered structures can be used to compute manipulative embeddings for potential data classification and recognition applications. Experiments for image object segmentation are carried out to demonstrate some potential applications of the NLE algorithm.

1 Introduction

Intrinsic structures, such as neighborhood relationship, global distribution and clustering, are the essence of exploratory data [1]. The discovery of intrinsic structures has a significant impact on data representation and manipulation, and has potential applications in many disciplines including information retrieval, image and video database analysis, data mining, climate pattern analysis, speech recognition and so forth [2–9].

The strategies and methodologies to discover intrinsic structure can be mainly categorized into linear methods such as Principal Component Analysis (PCA) which discovers the structural properties of these input using cross correlation [10] and Multidimensional Scaling (MDS) which seeks to preserve pairwise Euclidean distance and simple formations of data [11], and nonlinear methods such as Isomap by which the geodesic relationship among input data is kept unchanged in the embeddings computed [12], Locally Linearly Embeddings (LLE) by which the local neighborhood structures are remained in dimensionality reduction [1], and Laplacian Eigenmap (LE) which contributes to the weight computation of these links using a method deduced from the heat equations [13]. These geometry-based nonlinear methods were proposed due to the nonlinear properties of high-dimensional input data and seek to map a given set of high-dimensional data points into a low-dimensional space, starting with a preprocessing step that decides for each datum point which of the rest data points should be considered its neighbors [14]. Then they link each datum point to its neighbors and compute measures of the local geometry among input data points. These nonlinear methods have to choose appropriate neighbors for each datum point [15], which is a fundamental problem as neighbor selection affects the final outcome of the surrogate computed embeddings in a low-dimensional space. Normally, neighbor selection requires a priori information about the global geometry of the high-dimensional input data points, which is unavailable in most applications such that many methods may have to involve a trial and error process to select neighbors for each datum point, and the algorithms may not adapt to the data with different characteristics. Moreover, they focus on discovering either neighborhood relationship or global distribution. Some methods can overcome this problem. For example,Local tangent space alignment (LTSA) is a manifold learning algorithm efficient for many nonlinear dimension reduction problems and robust against the number of nearest neighbors. But large data sets

S. S. Ge (✉) · F. Guan · Y. Pan · A. P. Loh
Social Robotics Lab, Interactive Digital Media Institute,
Department of Electrical and Computer Engineering,
The National University of Singapore,
Singapore 117576, Singapore
e-mail: samge@nus.edu.sg

and new come data may cause performance decline of this method. In this paper, we propose an unsupervised learning algorithm to discover neighborhood relationship, global distribution, as well as clustering of input data points simultaneously and adaptively. The preliminary results of this research was presented in [7]. The main contributions of this paper are as follows:

i) An unsupervised learning algorithm, neighborhood linear embedding (NLE) is proposed, which eases the process of intrinsic structure discovery by avoiding the trial and error operations for neighbor selection, and at the same time, allows the discovery to adapt to the characteristics of the input data points. Furthermore, it is able to discover neighborhood relationship and global distribution of input data points simultaneously.
ii) The NLE algorithm can be extended to simultaneously discover another important structure intrinsic in the data, namely clustering, by using Euclidean distance histogram and a threshold approach.
iii) A closed-form solution is provided to compute the weight matrix constructed by the NLE algorithm.
iv) An image object segmentation approach is proposed whereby the operations are carried out in patches. The NLE algorithm combined with a dimensionality reduction approach is able to map objects in an image to clusters in the embedding space, which profits the subsequent classification and recognition processes.

For ease of presentation, the symbols used in this paper can be found in Table 1.

Table 1 Nomenclature

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Set of n-dimensional vectors
x_i	ith Input datum point
$\hat{x}_i$	Approximation of the ith input datum point
y_i	ith output embedding
X	Input matrix containing input data points
Y	Output matrix containing computed embeddings
n_D	Dimension of input data points
n_d	Dimension of computed embeddings
n_i	Neighbor number of x_i
N	Number of input data points
$\zeta_{i\sim j}$	Similarity measurement between x_i and x_j
$\varrho_{i,j}$	Euclidean distance between x_i and x_j
$\varrho_{i.\min}$	Euclidean distance of x_i to its nearest neighbor
Ω_i	Neighborhood set of x_i
$\Omega_i(m)$	mth element of Ω_i
$E^i_{k\|\phi}$	Evaluation of the additional information (provided by x_k) to x_i in the presence of an empty set, ϕ
$E^i_{k\|j}$	Evaluation of the additional information (provided by x_k) to x_i in the presence of x_j
$E^i_{k\|\Omega_i}$	Evaluation of the additional information (provided by x_k) to x_i in the presence of the set, Ω_i
w_{ij}	Weight of the link from x_j to x_i
W	The weigh matrix
γ	Constant demarking the decision boundary
η_i	Lagrange coefficient

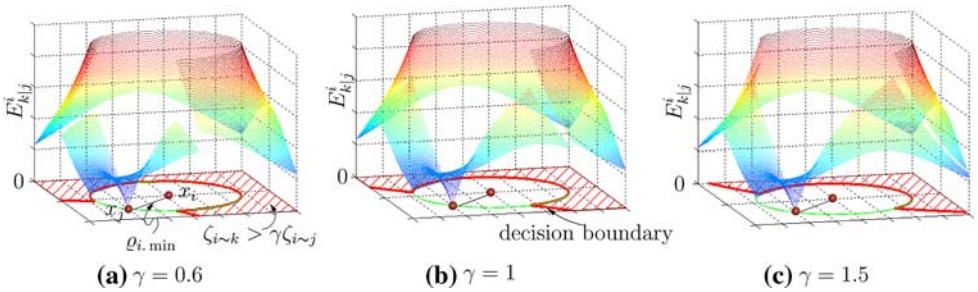
2 Intrinsic structure discovery

In general, methods of structure discovery are integrated into manifold learning or dimensionality reduction in the following manner: (i) a geometric structure is set up for each datum point, say x_i, by linking this point to its neighbors. These neighbors can be selected if they are the K-nearest neighbors (known as KNNs) or within a hyper-ball centered at x_i with radius, ϵ (known as ϵ-neighborhoods [13]), (ii) the weight of each link is assigned by methods deduced by the heat equations [13] or is computed such that all neighbors of a point can be used to approximate it with the minimum approximation error in the least square sense [1], and (iii) the computed weight matrix is used to compute embeddings corresponding to input data through dimensionality reduction methods. This process shows that intrinsic structure discovery forms the first and important pass of the whole process, and has significant impacts on the subsequent steps.

2.1 Neighborhood linear embedding (NLE)

To explore the intrinsic structure of the input data points, the KNN method has been widely used due to its simplicity and ease of implementation. However, this method is less geometrically intuitive as: (i) a small K leads to possible isolation of points, and (ii) a large K may result in the grouping of different clusters. In addition, the selection of K is a trial and error process that affects the tradeoff between cases (i) and (ii). The similarity is drawn to the selection of ϵ for the ϵ-neighborhoods method. Furthermore, due to the complexity, nonlinearity and variety of high-dimensional input data, it is difficult to apply a fixed K to all data points. An adaptive scheme to select K is more appropriate. Thus, we seek to select neighbors adaptively for the representation of each input datum point and avoid redundant information of its representation as much as possible. While it eases setting up a criterion for neighbor selection such that neighbor selection can be achieved in an unsupervised manner, this criterion may select neighbors in a more global view so that

Fig. 1 γ effect given $n_c = 1$



(a) $\gamma = 0.6$ **(b)** $\gamma = 1$ **(c)** $\gamma = 1.5$

more helpful information may be reserved. In this way, an unsupervised learning algorithm can be derived to select the neighbors for each input datum point, x_i, adaptively such that the data point can be approximated by

$$\hat{x}_i = \sum_{j=1}^{n_i} w_{ij} x_j \tag{1}$$

where w_{ij} is the weight of the link from a neighbor, x_j, to x_i, and n_i is the neighbor number of x_i. Before proceeding further to select these neighbors, we define a similarity measurement for two input data points, x_i and x_j ($x_i, x_j \in \mathbb{R}^{n_D}$), by

$$\zeta_{i\sim j} = \exp(-\varrho_{i,j}) \tag{2}$$

where $\varrho_{i,j} = \|x_i - x_j\|$ is the Euclidean distance between x_i and x_j. The properties of this similarity measurement are:

1. $\zeta_{i\sim j}$ is maximized, i.e., 1, if $\varrho_{i,j} = 0$. It indicates that x_i and x_j are identical, and x_i can be fully represented or approximated by x_j.
2. $\zeta_{i\sim j}$ is minimized, i.e., 0, if $\varrho_{i,j} \to +\infty$, which means that x_j is totally different from x_i. Thus, point x_j has no contribution to the presentation of x_i.
3. $\zeta_{i\sim j}$ decreases monotonically with respect to $\varrho_{i,j}$. It means that the further x_j is from x_i, the less contribution of x_j to the representation of x_i.

These properties provide an evaluation about how much x_j can be used to approximate x_i. If x_j is the nearest point to x_i, $\zeta_{i\sim j}$ is of the maximum value as compared to those of other points. Thus, we assume that the nearest point of x_i is one of its neighbors. If a new point, say x_k, is given in the presence of x_j, we need to evaluate the additional information provided by x_k (to x_i) to avoid any redundancy in the overall representation of x_i. This evaluation is obtained based on the following assumptions:

A1: If x_k coincides with x_j, x_k can be fully represented by x_j. It has no contribution to the approximation of x_i and should be discarded to avoid redundancy of x_i representation.

A2: Point x_k is a new neighbor of x_i if it is more similar to x_i than x_j. Mathematically, this condition can be expressed by $\zeta_{i\sim k} \geq \gamma\zeta_{j\sim k}$, where γ is a constant.

A3: Point x_k has the same similarity to both x_i and x_j if the distance from x_k to x_i (x_j) is large as compared to ϱ_{ij}.

To fulfil these assumptions, we define a convention by

$$E^i_{k|j} = \begin{cases} \zeta_{i\sim k}, & \text{if } \zeta_{i\sim k} \geq \gamma\zeta_{j\sim k} \\ (1-\zeta_{j\sim k})^{n_c}\zeta_{i\sim k}, & \text{otherwise} \end{cases} \tag{3}$$

to evaluate the additional representation information provided by x_k, where n_c is a constant. The value of γ demarkes a boundary where $\zeta_{i\sim k}$ is γ times of $\zeta_{j\sim k}$. To investigate the effect of γ on $E^i_{k|j}$, a simple illustration is shown in Fig. 1, where x_i and its nearest neighbor, x_j, are two-dimensional input data points. The distance from x_i to x_j is $\varrho_{i.\min}$. It determines a circle centered at x_i. The area outside of this circle is the possible location of x_k, which is divided into two sub-areas by part of the circle and a boundary defined by $\zeta_{i\sim k} = \gamma\zeta_{j\sim k}$. The sub-area with the hashed pattern shows the condition where $\zeta_{i\sim k} > \gamma\zeta_{j\sim k}$. If x_k tends to x_j, $E^i_{k|j}$ goes to zero, which fulfills the assumption A1. It is clear from (3) that $(1-\zeta_{j\sim k})^{n_c}\zeta_{i\sim k} \leq \zeta_{i\sim k}$, and $(1-\zeta_{j\sim k})^{n_c}\zeta_{i\sim k} \approx \zeta_{i\sim k}$ iff $\varrho_{jk}(\varrho_{ik}) \to +\infty$. This fulfills the assumption A3. For completion, the convention in (3) is extended to other two cases where an empty set, ϕ, and a neighborhood set, Ω_i, are present, and let $E^i_{k|\phi} = \zeta_{i\sim k}$ and $E^i_{k|\Omega_i} = \min\{E^i_{k|m}\}$, $m \in \Omega_i$.

Based on these evaluations, the neighbors of x_i are chosen in the following manner:

1. If x_j is the closest point to x_i, x_j is assumed to be a neighbor of x_i. Thus, $\Omega_i = \{j\}$ initially.
2. Suppose that x_k is the second nearest point to x_i. Point x_k is a neighbor of x_i if $E^i_{k|j}$ in (3) is maximized. To maximize $E^i_{k|j}$, we can simply apply the criterion $\zeta_{i\sim k} \geq \gamma\zeta_{j\sim k}$, which can be simplified by

$$\varrho_{j,k} \geq \varrho_{i,k} + \ln\gamma \tag{4}$$

where $\ln\gamma \geq 0$ ($\gamma \geq 1$) can be an additional condition to avoid eliminating the effect of $\varrho_{i,k}$. Thus, the

neighborhood set, Ω_i, can be updated by

$$\Omega_i = \begin{cases} \Omega_i \cup \{k\}, & \text{if } \varrho_{j,k} \geq \varrho_{i,k} + \ln \gamma \\ \Omega_i, & \text{otherwise} \end{cases} \tag{5}$$

In this way, points in Ω_i have greater effects on the presentation of x_i as compared to the reciprocal effects among themselves.

3. To examine whether x_m is a neighbor of x_i if Ω_i contains two or more elements, the condition in (4) is applied to all these elements and Ω_i is updated by

$$\Omega_i = \begin{cases} \Omega_i \cup \{m\}, \text{ if } \varrho_{j,m} \geq \varrho_{i,m} + \ln \gamma \quad \forall j \in \Omega_i \\ \Omega_i, \qquad \text{otherwise} \end{cases} \tag{6}$$

These three steps are applied to all data points (except for x_i) in an ascending order of distance to x_i and Ω_i is thus obtained. To store the discovered links among input data points, the neighborhood sets Ω_i $(i = 1, 2, \ldots, N$, N is the number of input data points) are used to construct a weight matrix $W = \{w_{ij}\}, (i, j = 1, \ldots, N)$ with the following constraint

$$\sum_{j=\Omega_i(1)}^{\Omega_i(n_i)} w_{ij} = 1, \quad \forall \quad i = 1, \ldots, N, \tag{7}$$

where $\Omega_i(j)$ is the jth element in Ω_i and w_{ij} is a constant weight representing the contribution level of x_j to the approximation of x_i. The way to compute the weight matrix will be discussed later. The construction of Ω_i and W is detailed in Algorithm 1.

Algorithm 1: $W = \mathbf{NLE}(X, \gamma)$

Data: γ, X (compute distance matrix D from X and sort it in ascending order to have the sorted distance matrix NE and corresponding index matrix L)
Result: W

```
for i = 1 to N do
    Ω_i = {L_2i};
    n_i = 1;
    for j = 3 to N do
        if (NE_ji + ln γ) ≤ ϱ_{k L_ji}   ∀ k ∈ Ω_i then
            Ω_i = Ω_i ⋃ {L_ji};
            n_i = n_i + 1 ;      /* n_i is the neighbor number
            of x_i */
for i = 1 to N do
    for j = Ω_i(1) to Ω_i(n_i) do
        compute w_ij ;          /* Construct weight matrix W */
```

The links among input data points can be discovered using the NLE algorithm. But the weight of these links, w_{ij}, are unknown. The next step is to compute the weight matrix W such that the intrinsic structure of the data can be completely known. The computation of weight matrix can be achieved by minimizing the approximation error for each datum point x_i [1]. Take the ith row of weight matrix, W_i, for example, the approximation error can be computed by

$$\begin{aligned} \varepsilon(W_i) &= \|x_i - \hat{x}_i\| = \|x_i \sum_{j=\Omega_i(1)}^{\Omega_i(n_i)} w_{ij} \\ &\quad - \sum_{j=\Omega_i(1)}^{\Omega_i(n_i)} w_{ij} x_j\| = \| \sum_{j=\Omega_i(1)}^{\Omega_i(n_i)} w_{ij}(x_i - x_j)\| \\ &= \sum_{j=\Omega_i(1)}^{\Omega_i(n_i)} w_{ij} \sum_{k=\Omega_i(1)}^{\Omega_i(n_i)} w_{ik}(x_i - x_j)^T (x_i - x_k). \end{aligned} \tag{8}$$

By defining $C_i(j,k) = (x_i - x_j)^T (x_i - x_k)$ and applying a Lagrange multiplier to Eq. (7), we have

$$\begin{aligned} \varepsilon(W_i) &= \sum_{j=\Omega_i(1)}^{\Omega_i(n_i)} w_{ij} \sum_{k=\Omega_i(1)}^{\Omega_i(n_i)} w_{ik} C_i(j,k) \\ &\quad + \eta_i \left(\sum_{j=\Omega_i(1)}^{\Omega_i(n_i)} w_{ij} - 1 \right) \end{aligned} \tag{9}$$

where η_i is the Lagrange coefficient. The partial differentiation of ε with respect to each weight $w_{i\Omega_i(j)}$ is

$$\frac{\partial \varepsilon(W_i)}{\partial w_{i\Omega_i(j)}} = 2 \sum_{k=\Omega_i(1)}^{\Omega_i(n_i)} w_{ik} C_i(\Omega_i(j), k) + \eta_i, \quad \forall\, j \in \Omega_i. \tag{10}$$

Let $\frac{\partial \varepsilon(W_i)}{\partial w_{i\Omega_i(j)}} = 0, \forall j$ and consider the weight constraint in Eq. (7), we have

$$\overline{C}\ \overline{W}_i^T = \overline{q} \tag{11}$$

where

$$\overline{C} = \begin{bmatrix} 0 & \mathbf{1} \\ \mathbf{1} & 2C \end{bmatrix}$$

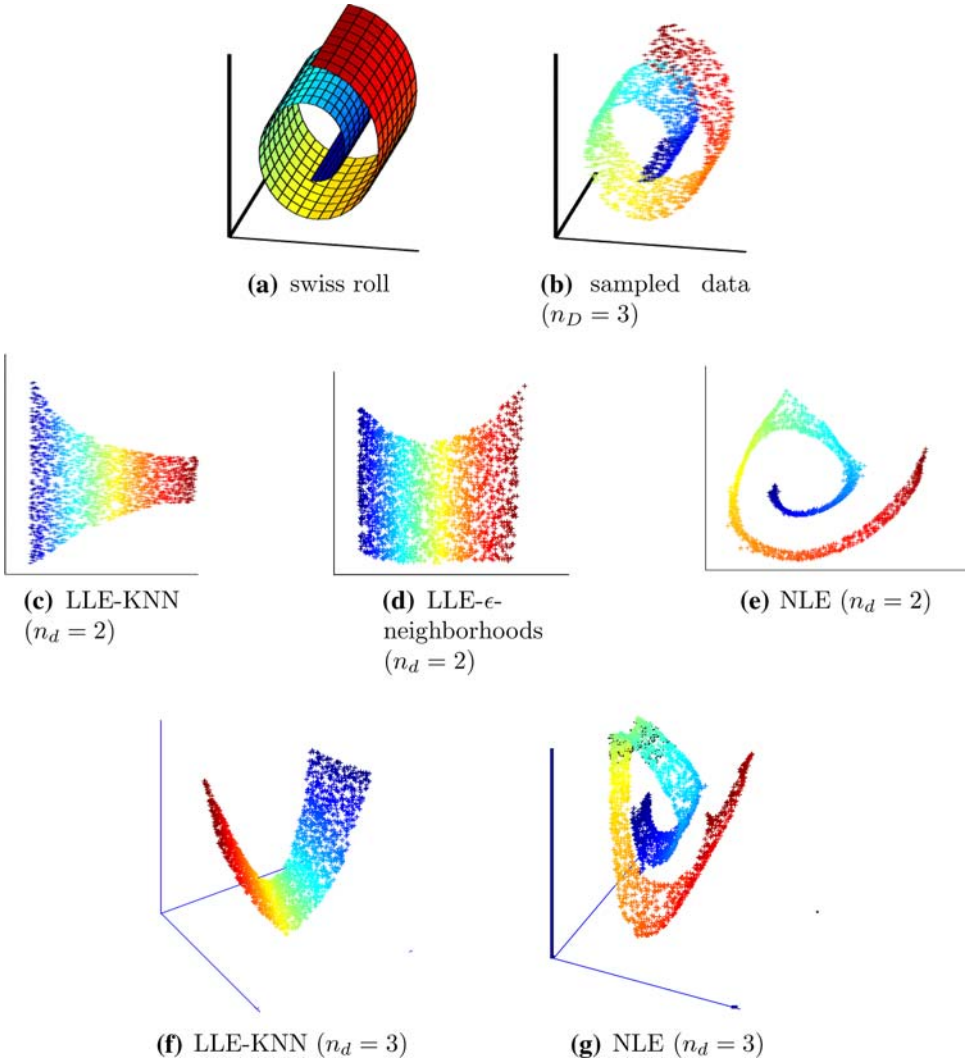
$$C = \begin{bmatrix} C_i(\Omega_i(1), \Omega_i(1)) & \cdots & C_i(\Omega_i(1), \Omega_i(n_i)) \\ \vdots & \ddots & \vdots \\ C_i(\Omega_i(1), \Omega_i(n_i)) & \cdots & C_i(\Omega_i(n_i), \Omega_i(n_i)) \end{bmatrix}$$

$$\overline{W}_i^T = [\,\eta_i \quad w_{i\Omega_i(1)} \quad w_{i\Omega_i(2)} \quad \cdots \quad w_{i\Omega_i(n_i)}\,]^T = [\eta_i\ W_i]^T$$

$$\bar{q} = [1 \quad 0 \quad \cdots \quad 0]^T$$

where $C = [C_{jk}](j, k = 1, \ldots, n_i)$ is a symmetric matrix with dimension $n_i \times n_i$, $C_{jk} = C_i(\Omega_i(j), \Omega_i(k))$, $W_i = [w_{i\Omega_i(1)}\, w_{i\Omega_i(2)} \cdots w_{i\Omega_i(n_i)}]$. Since the inverse of C depends strongly on the input data points, it may exist under very restrictive conditions. If the number of neighbors is larger than the dimension of input data points, C may be singular, which subsequently leads to the singularity of $\overline{C}$. To calculate the weight matrix W_i uniquely, matrix C is regulated by $C = C + \eta_r \mathrm{I}$, where I is an $n_i \times n_i$ identity matrix and η_r is

Fig. 2 Example of Swiss roll



a constant with a small value. The regularization allows $\overline{C}$ to be full rank. However, to reduce the effect of η_r on $\overline{C}$, η_r is chosen to be small as compared to the trace of C. Thus, with this necessary regularization of C in $\overline{C}$, we have

$$\overline{W}_i^T = \overline{C}^{-1}\overline{q}. \tag{12}$$

In this way, W is obtained row by row. The computed weight matrix contains information about the neighborhood relationship represented spatially by the position of the non-zero components, and the contribution of one point to another is represented numerically by the component values.

To manipulate the high-dimensional data points and visualize their intrinsic structures, NLE is combined with a dimensionality reduction technique to observe these structures in a low-dimensional description space. The low-dimensional embeddings Y can be obtained by choosing the eigenvectors associated with the n_d lowest eigenvalues of matrix $M = (\mathrm{I} - W)^T(\mathrm{I} - W)$, with the Rayleitz–Ritz Theorem playing its role [16]. For comparison purpose, simulations based on the Swiss roll data are given in Fig. 2. Figure 2b shows the 3D data points randomly sampled from the 3D manifold illustrated in Fig. 2a while Fig. 2c–e illustrate the computed embeddings based on KNN, ϵ-neighborhoods and NLE algorithm, respectively, for the condition where $n_d = 2$. According to the color coding, the neighborhood relationships are preserved by selecting K or ϵ using the trial and error method. Unlike these "unfolded" embeddings, NLE based approach computes the embeddings by putting neighborhood embeddings closer and preserving the shape of Swiss roll. To have a better comparison between KNN and NLE, we compute the embeddings of the same data points by setting $n_d = n_D = 3$. It is expected that the computed embeddings should be similar to those in Fig. 2b. The computed embeddings using KNN and NLE based approaches are shown in Fig. 2f and e, respectively. It shows that KNN based approach loses a main feature of input data points, i.e., roll shape, while NLE preserves not only the neighborhood

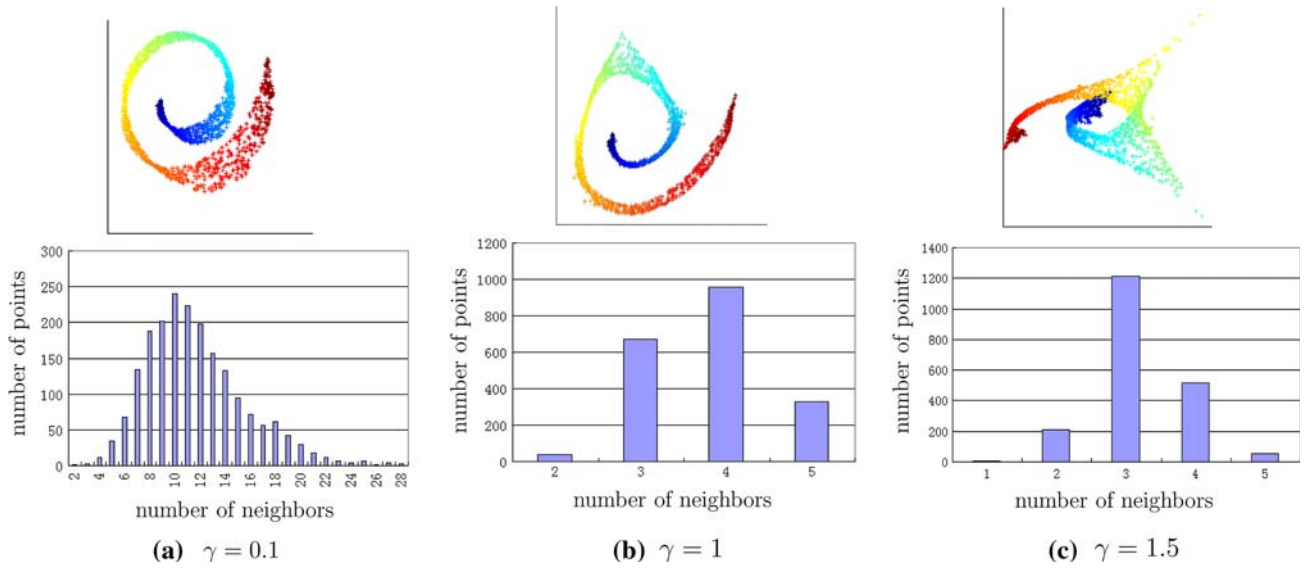


Fig. 3 Structure discovery of Swiss roll using NLE

relationship but also this global distribution feature. In this way, NLE based approach is able to preserve the intrinsic structure of the input data as much as possible in the process of dimensionality reduction. The fundamental characteristics of these results are summarized as follows:

i) It is clear from the color coding that neighborhood relationships in high-dimensional input data are preserved in the low-dimensional embeddings.
ii) NLE preserves not only the local neighborhood relationship but also the global distribution of the original data set. As shown in Fig. 2, the roll-like shape of the three-dimensional manifold is kept unchanged, which cannot be achieved by KNN or ϵ-neighborhoods method.

In order to investigate the effects of γ, the sampled data points as in Fig. 2b are processed by NLE based approach with different values of γ. The computed embeddings are shown in Fig. 3. The three graphs in the first row are the computed embeddings and the second row shows the distribution of the neighbor number. The results show that the value of γ affects the level of neighbor number. The smaller value of γ is, the more neighbors an input point may have. However, $\gamma = 1$ is able to provide satisfactory results such that γ is set to 1 in the rest of this paper.

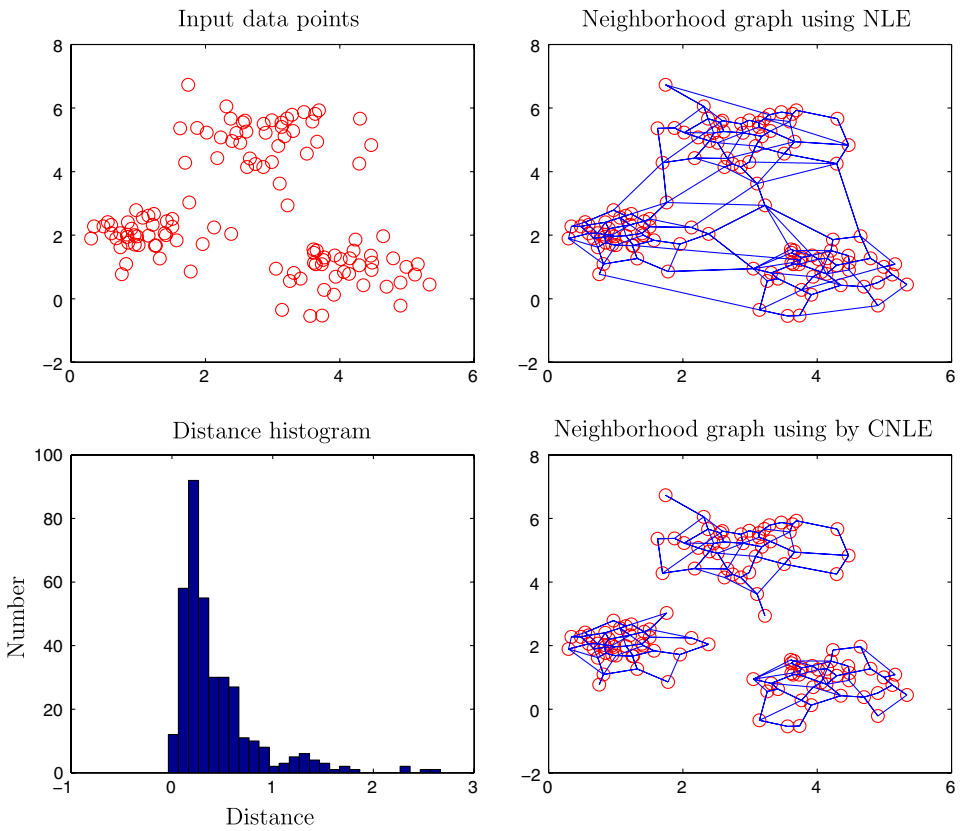
Although NLE is able to discover the neighborhood relationship and global distribution of input data points simultaneously, it increases the computation complexity of discovering the links to $O(N^2 \times n_D + N^3)$ while that of KNN and ϵ-neighborhoods are $O(N^2 \times n_D)$ for a fixed value of K or ϵ. Since KNN and ϵ-neighborhoods involve trial and error operations to look for an appropriate value of K and ϵ, NLE may be computationally low as compared to KNN and ϵ-neighborhoods methods from the overall point of view. Due to the fact that NLE is able to avoid the trial and error operations used in KNN or ϵ-neighborhoods methods, it eases the discovery process. Furthermore, it allows the discovery process to adapt to the characteristics of each input datum point as in Fig. 3.

2.2 Clustering

Besides neighborhood relationship and global distribution, clustering is another important intrinsic structure of input data. Generally, data with similar features or structures are close geometrically to each other. This subsequently increases the density at certain areas. For ease of density computation, we make use of the Euclidean distance histogram to simply extend NLE and design the clustering neighborhood linear embedding algorithm (CNLE). Figure 4 shows the procedure of clustering based on this idea. The top right graph shows the discovered structures of a given set of 2D data that are shown in the top left graph by using NLE. The corresponding distance histogram for these links in the graph is shown in the bottom left graph. Based on the histogram, a threshold approach can be applied to remove links that have low probability of occurrence. These links correspond to the low density areas in the input space. Let c_h be the threshold such that a link with length larger than c_h will be removed. The clustered structure is shown in the bottom right graph wherein three clusters are identified. The algorithm to implement CNLE is detailed in Algorithm 2 with input X and threshold c_h.

For the comparison of the clustering property, graphic illustrations for LLE–KNN, NLE and CNLE are shown by using the data in the right graph of Fig. 5. These data are

Fig. 4 Clustering procedure



Algorithm 2: $W = \mathbf{CNLE}(X, \gamma, c_h)$

Data: γ, c_h, X (compute distance matrix D from X and sort it in ascending order to have the sorted distance matrix NE and corresponding index matrix L)
Result: W

for $i = 1$ **to** N **do**
  $\Omega_i = \{L_{2i}\}$;
  $n_i = 1$;
  for $j = 2$ **to** N **do**
    if $(NE_{ji} + \ln\gamma) \le \varrho_{kL_{ji}}$ and $NE_{ji} \le c_h \quad \forall\ k \in \Omega_i$ **then**
      $\Omega_i = \Omega_i \bigcup \{L_{ji}\}$;
      $n_i = n_i + 1$; /* n_i is the neighbor number of x_i */
for $i = 1$ **to** N **do**
  for $j = \Omega_i(1)$ **to** $\Omega_i(n_i)$ **do**
    $w_{ij} = a_{ij}$ $(a_{ij} \neq 0)$; /* construct weight matrix W */

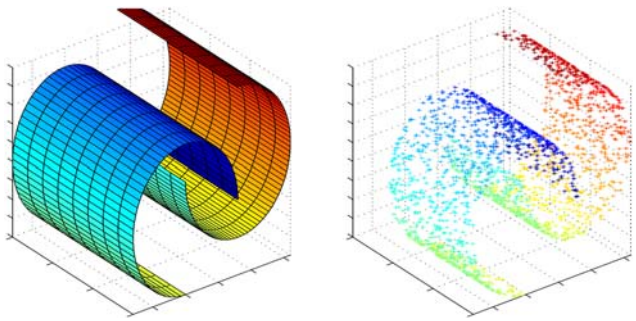


Fig. 5 Manifold of two rolls and corresponding samples

randomly sampled from two disjoint rolls that are close together with certain overlapping as shown in the left graph of the figure. Computed embeddings using LLE–KNN with different K values are shown in Fig. 6. If $K = 7$, the embeddings form a line, which cannot provide any neighborhood or cluster characteristics of these two rolls. Similarly, no information about neighborhood relationship and cluster information can be obtained if $K = 34$ as in Fig. 6c. If the number of neighbors is properly chosen, e.g., $K = 16$, cluster structure is discovered as in Fig. 6b. However, it is not able to preserve the global distribution of input data points. These results indicate that LLE–KNN can only discover a limited type of intrinsic structures even if K is carefully chosen through trial and error process. In comparison to LLE–KNN, NLE is able to preserve the neighborhood relationship and the global distribution of the data simultaneously as shown in Fig. 7a. However, from the clustering point of view, the computed embeddings are overlapped to certain degree. The graph in Fig. 7b, however, shows that CNLE not only preserves the neighborhood relationship and global distribution but also the clustering of the input data points. It is worth noting that the threshold in CNLE should be chosen carefully as it affects the clustering property greatly. Moreover, CNLE cannot adapt to the clustering statistics of input data points. For further comparison, the same input data

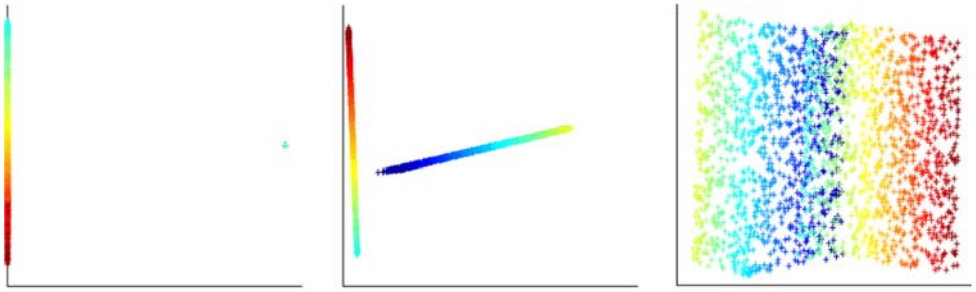


Fig. 6 Clustering based on LLE–KNN

(a) K=7 **(b)** K=16 **(c)** K=34

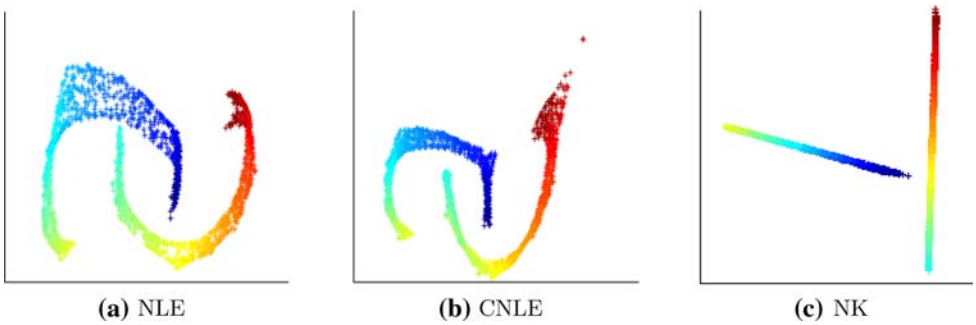


Fig. 7 Clustering based on NLE, CNLE and NK algorithm

(a) NLE **(b)** CNLE **(c)** NK

points are fed into neighborhood knowledge (NK) algorithm to compute embeddings. NK algorithm aims to improve the clustering property of LLE and ISOMAP algorithm [17,18]. Two main parameters for this algorithm are: (i) neighborhood number $K = 16$, and (ii) a threshold $P_{false} = 2$, which are carefully chosen through trial and error process. Figure 7c shows the computed embeddings. Although this approach is able to cluster the computed embeddings and constructs the neighborhood graph within each cluster, the global distribution property of input data points cannot be preserved.

From the numerical analysis presented in Figs. 6 and 7, we have the following observations:

(i) LLE–KNN and NK can give the most intuitive and straightforward clustering as shown by Figs. 6b and 7c.
(ii) The LLE–KNN method may give a good clustering result to make the consequent classification procedure easier, but this can only be obtained with a careful selected parameter K.
(iii) NK gives good result in this simulation study, but the algorithm add a new parameter P_{false}, which make the situation more complicated and the computational complexity is quite high.
(iv) NLE and CNLE have advantages in that they preserve not only the clustering of the input data points but also the neighborhood relationship and global distribution and avoid trial and error procedure for parameter selection. But from the clustering point of view, the disadvantages of NLE and CNLE is that they don't give a very strict and direct clustering result as LLE–KNN and NK.
(v) CNLE may make less overlapping in clustering result, but the threshold in CNLE affects the clustering property greatly.
(vi) As such, a trade off to be made in all the methods presented in computational complexity, user interferences in the tuning, which method is better is a situation dependent case.
(vii) Many other common spectral methods such as spectral clustering [19,20] and normalized cuts [21] are efficient for object segmentation in images.

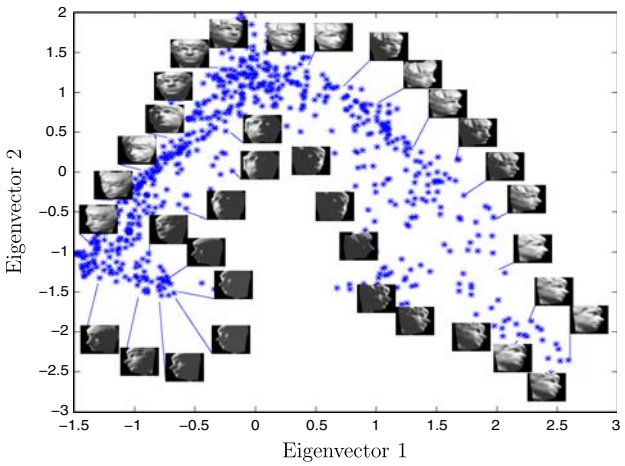
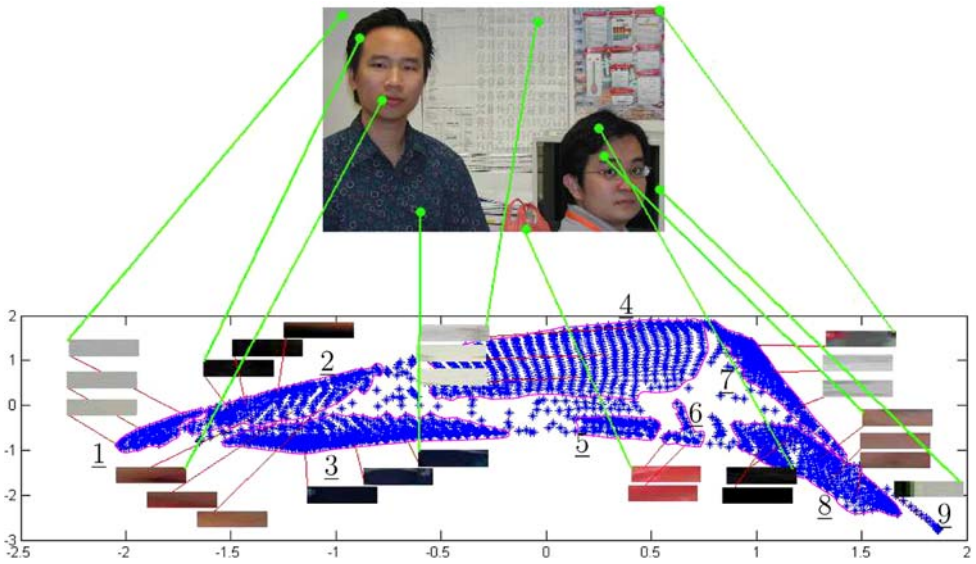


Fig. 8 Calculated embeddings of face pose

Fig. 9 Feature clustering



3 Simulation studies

In this section, several simulation studies are presented to demonstrate the potential applications of NLE and CNLE. The first simulation study is to find the coherent relationship among a set of face images using NLE approach. This data set contains $N = 698$ gray images at a resolution of 64×64, i.e., $n_D = 4096$ [12]. Each image can be regarded as a collection of numbers, each specifying light intensity at an image pixel. This collection of numbers also specifies the Cartesian coordinates of a point with respect to a set of axes. Therefore, each image can be identified with a point in an abstract high-dimensional image space. The input datum point x_i is constructed by formatting the image pixel column by column from left to right and concatenating them to form a column vector. The computed two-dimensional embeddings are shown by "$*$" in Fig. 8, some of which have corresponding image shown next to them. These embeddings form an arch-bridge shape, in which the individual axes correspond roughly to the small number of degrees of freedom present in the data. Although n_D is large, the motion of the subject head can be parameterized by only two variables, namely, azimuth, α, and elevation, β, which may be represented by the horizontal and vertical axis respectively. Suppose each of these images, x_i, is associated with a known vector $[\alpha_i, \beta_i]$. If a new face image is given, we can compute its corresponding embedding and find its position in Fig. 8. By using the relationships of the new computed embedding to its neighbors, the vector $[\alpha, \beta]$ associated with the new image can be estimated. Thus, we are able to identify the direction the subject is looking at.

The second simulation is to study image object segmentation using the CNLE approach. In this simulation study,

Table 2 Relationship between computed clusters and image objects

Cluster	Image object
1	The wall in the top left area
2	Hair (denser area) and face of the left colleague
3	T-shirt
4	Light color paper in the middle
5	Paper in bottom middle area
6	Color plastic bag
7	Color calendar in the top right area
8	Hair (the top left sub-cluster in this cluster) and face of the right colleague
9	Edge of the PC screen

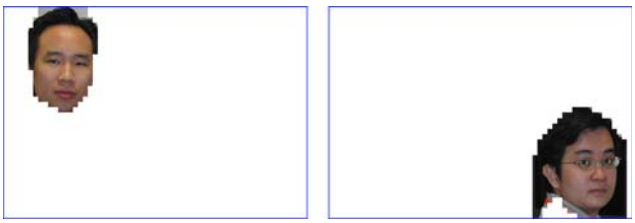


Fig. 10 Image feature

we segment the image objects from another perspective: an image can be divided into many image patches. Each image patch is treated as an input datum point. For simplicity, these image patches are identical in size. A sample image is given at the top of Fig. 9. This picture is divided into square patches at a resolution of 17×17. Due to the fact that the image objects are distributed in the image spatially, the input data points are constructed using vector concatenation by joining

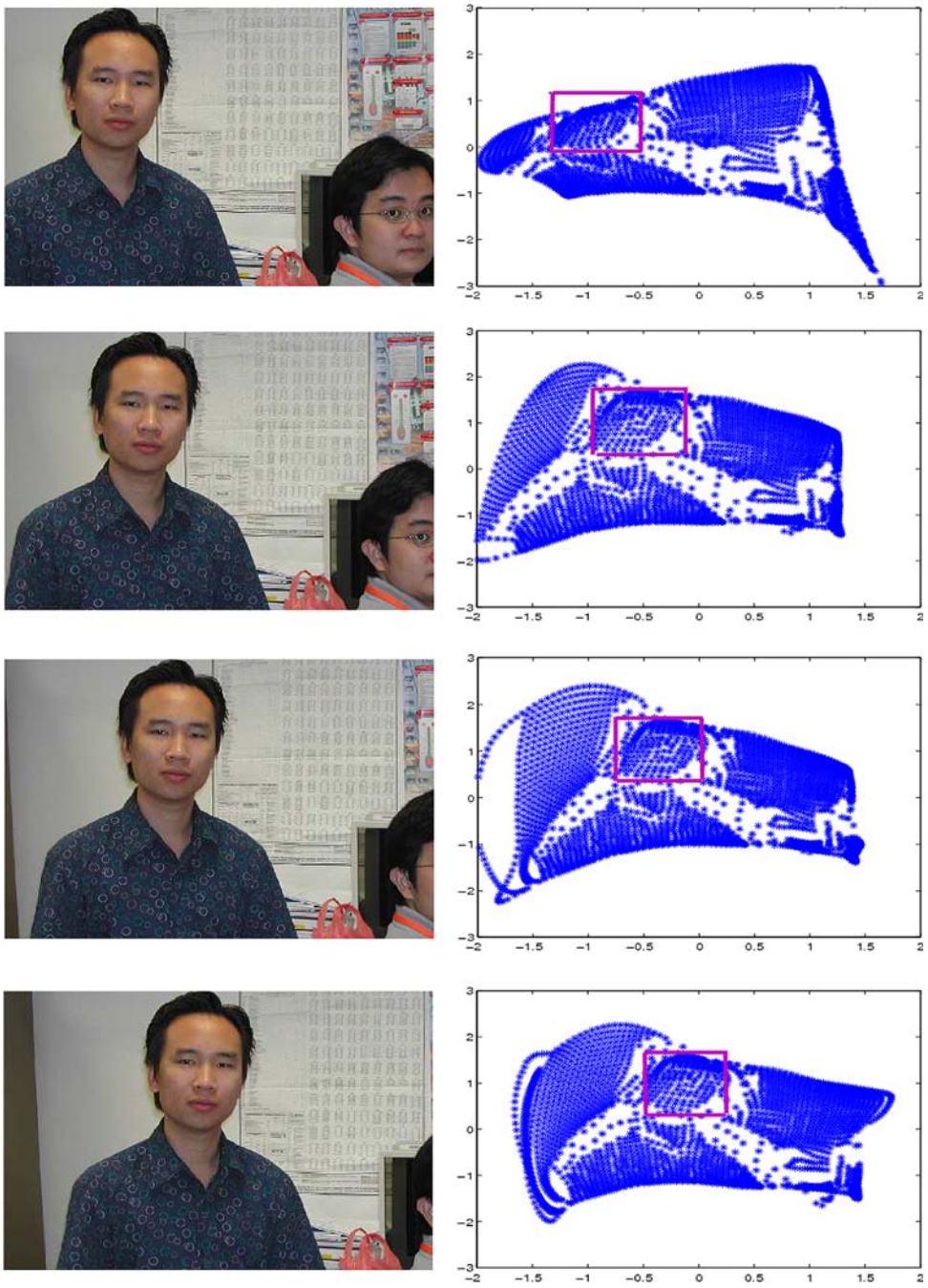


Fig. 11 Motion sequence and corresponding embeddings

the spatial information to image color information, namely, $\bar{x}_i = [\eta_c x_i^T, \eta_s \tilde{x}_i^T]^T$, where $\tilde{x}_i = [y_i, z_i]^T$ is the image coordinate of the center of an image patch corresponding to x_i, η_c and η_s are constants that summarizes the weight of color and spatial information in the input data. The computed embeddings are shown in the bottom graph of Fig. 9. It can be observed that these embeddings form several clusters and each cluster corresponds to one object in the image as summarized in Table 2. Clearly, the clustering of data are realized in clusters and the neighborhood relationships are preserved in the form of clusters. Since the embedding y_i corresponds to x_i and x_i is associated with the spatial information provided by $\tilde{x}_i$,

$$\begin{array}{l} y_i \in \text{Cluster}_m \ \ (\text{the } m\text{th cluster in the embedding space}) \\ \downarrow \\ x_i \in \text{Object}_m \ \ (\text{the } m\text{th object in image}) \end{array} \tag{13}$$

an image object can be extracted by piecing up the image patches corresponding to the embeddings in a cluster. In this

way as shown in Fig. 10, two image objects can be extracted from the image straightaway.

This result indicates that image objects can be segmented in a straightforward manner without complicated image processing techniques such as edge detection, model generation of object and so forth. However, it is also observed that the head of the right colleague corresponding to cluster $\underline{8}$ as shown in Fig. 9 is not segmented properly, because some image patches of this object is similar to those of the PC screen and they are spatially connected. Moreover, it should be brought into attention that the construction of $\bar{x}_i$ lends weight to argue the effect of η_s on the overall result, i.e., spatial information on the global clustering. Our investigation shows that the computed embeddings spread if η_s increases. The investigation is omitted in this paper due to the space limitation.

The third simulation is to study image object tracking. It is based on the observations in (13). Accordingly, tracking of an image object in an image sequence can be achieved by tracking the cluster of interest. A sampled image sequence is given in the left column of Fig. 11 where the second column shows the corresponding computed embeddings. The rectangle covers the embeddings that are associated with the head of the left colleague. By observing the motion sequence, it can be concluded that the embeddings move in the same manner as that of the objects. Therefore, tracking of an embedding cluster is equivalent to tracking of an image object.

4 Conclusion

In this paper, we have presented an unsupervised learning algorithm to discover the intrinsic structures of data, such as neighborhood relationships, global distributions and clustering. The proposed algorithm eases the process of intrinsic structure discovery by avoiding trial and error operations for neighbor selection, and at the same time, allows the discovery to adapt to the characteristics of input data. Furthermore, it is able to discover intrinsic structures of data simultaneously, and the discovered structures can be used to compute manipulative embeddings for potential classification and recognition purposes. Experiments for image object segmentation have been carried out to demonstrate some potential applications of the NLE algorithm.

Acknowledgments The authors would like to thank J. B. Tenenbaum, Massachusetts Institute of Technology, for providing the data for experiments.

References

1. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
2. Besada, J.A., Molina, J.M., Garcfa, J., Berlanga, A., Portillo, J.: Aircraft identification integrated into an airport surface surveillance video system. Mach. Vis. Appl. **15**, 164–171 (2004)
3. Yang, M.-H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 34–58 (2001)
4. Zhang, C.S., Wang, J., Zhao, N.Y., Zhang, D.: Reconstruction and analysis of multi-pose face images based on nonlinear dimensionality reduction. Pattern Recognit. **37**, 325–336 (2004)
5. Gu, H.S., Ji, Q.: Information extraction from image sequences of real-world facial expressions. Mach. Vis. Appl. **16**, 105–115 (2005)
6. Charif, H.N., McKenna, S.J.: Tracking the activity of participants in a meeting. Mach. Vis. Appl. **17**, 83–93 (2006)
7. Ge, S.S., Guan, F., Loh, A.P., Fua, C.H.: Feature representation based on intrinsic structure discovery in high dimensional space. In: The 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida, pp. 3399–3404, 15–19 May (2006)
8. Ge, S.S., Fua, C.H.: Queues and artificial potential trenches for multi-robot formations. IEEE Trans. Robot. **21**, 646–656 (2005)
9. Ge, S.S., Lai, X.C., Mamun, A.A.: Boundary following and globally convergent path planning using instant goals. IEEE Trans. Systems Man Cybern. B Cybern. **35**, 240–254 (2005)
10. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, Heidelberg (2002)
11. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. Monographs on Statistics and Applied Probability, vol. 88, 2nd edn. CRC Press (2000)
12. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A gloabal geometric framework for nonlinear dimensionality reduction, Science **290**,2319–2323, December 2000 http://isomap.stanford.edu/datasets.html
13. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**(6), 1373–1396 (2003)
14. Seung, H.S., Lee, D.D.: The manifold ways of perception. Science **290**, 2268–2269 (2000)
15. Balasubramanian, M., Schwartz, E.L.: The isomap algorithm and topological stability. Science **295**, 7a (2002)
16. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1990)
17. Zhang, Y.G., Zhang, C.S., Wang, S.J.: Clustering in knowledge embedded space. In: Title: The 14th European Conference on Machine Learning (ECML-2003), Cavtat, Croatia, Proceedings, September 2003. Lecture Notes in Computer Science, vol. 2837, pp. 480-491 (2003)
18. Zhang, Y.G., Zhang, C.S., Zhang, D.: Distance metric learning by knowledge embedding. Pattern Recognit **37**, 161–163 (2004)
19. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems (NIPS), vol. 14 (2002)
20. Bach, F.R., Jordan, M.I.: Learning spectral clustering. In: Advances in Neural Information Processing Systems (NIPS) 16, (2004)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)