PAPER

View-Based Object Recognition Using ND Tensor Supervised Neighborhood Embedding

Xian-Hua HAN^{†a)}, Nonmember, Yen-Wei CHEN[†], Member, and Xiang RUAN^{††}, Nonmember

SUMMARY In this paper, we propose N-Dimensional (ND) Tensor Supervised Neighborhood Embedding (ND TSNE) for discriminant feature representation, which is used for view-based object recognition. ND TSNE uses a general N^{th} order tensor discriminant and neighborhood-embedding analysis approach for object representation. The benefits of ND TSNE include: (1) a natural way of representing data without losing structure information, i.e., the information about the relative positions of pixels or regions; (2) a reduction in the small sample size problem, which occurs in conventional supervised learning because the number of training samples is much less than the dimensionality of the feature space; (3) preserving a neighborhood structure in tensor feature space for object recognition and a good convergence property in training procedure. With Tensor-subspace features, the random forests is used as a multi-way classifier for object recognition, which is much easier for training and testing compared with multi-way SVM. We demonstrate the performance advantages of our proposed approach over existing techniques using experiments on the COIL-100 and the ETH-80 datasets.

key words: tensor analysis, supervised neighborhood embedding, subspace learning, random forests, view-based object recognition

1. Introduction

View-based 3D object recognition remains a hard problem for computer vision. Several types of approaches have been proposed in the literature [1]–[6]. Poggio and Edelman [1] proposed an exemplar-based approach with a network of generalized radio basis functions for recognizing a 3D object from its 2D images. Murase and Nayar [2] proposed a parametric appearance eigenspace approach, which has been further extended effectively to modelling face images by Gong et al. [3]. The machine learning algorithm based on Statistical Learning Theory has also been used for view-based object recognition. Particularly, support vector machines (SVMs) have been extensively evaluated for object recognition. Both linear and non-linear kernels have been used achieving good results on benchmark data sets [4], [5]. Another machine learning technique, called Sparse Network of Winnows (SNoW), has also been shown to be effective for view based object recognition [6]. SNoW is able to learn explicitly a representation of an object, unlike SVMs which only define discriminating boundaries. However, all of the above have one common characteristic: representing 2D images by 1D vectors. This vectorization is rather ad hoc and

not optimal because it does not preserve any nonlinear structure and shape information of the data. It can also result in a very large image representational space with poor numerical properties and computational tractability. Wang [7] proposed to use Tensor Discriminant Analysis for view-based object recognition, which obtained much better results on the benchmark databases (COIL-100 and ETH-80) compared to those of the state of the art approaches. However, Tensor discriminant Analysis deals with samples in the same category equally and does not consider the relationships among the same class samples. Then, if some outliers or un-conventional samples are used for training data, In order to minimize the distance between the same class samples, the learning procedure maybe lead to biased basis functions. Therefore, our work propose to use neighbor similarity in the same category as weight of minimizing cost function for N^{th} order tensor analysis, which is able to estimate geometrical and topological properties of the sub-manifold tensor from random points (“scattered data”) lying on this unknown sub-manifold.

The proposed N^{th} order SNE tensor represents color images based on the considerations of multilinear algebra and differential geometry. For a color image of size $N_1 \times N_2 \times 3$, it is represented as the third order tensor in the tensor space $R^{N_1 \times N_2 \times 3}$. On the other hand, the color image space is generally a sub-manifold embedded in $R^{N_1 \times N_2 \times 3}$. Given some color images sampled from the object-view manifold, we can build a supervised adjacency graph to model the local geometrical structure of the manifold for the same category. ND SNE finds a projection that respects this graph structure. The obtained N^{th} tensor subspace provides an optimal linear approximation to the object manifold in the sense of local isometry. In this work, we use the proposed ND SNE analysis for View-based object recognition. This method represents a color image as a 3rd order tensor, resulting in a much smaller dimension. A similar idea of 2D tensor neighborhood embedding analysis has also been proposed by He et al. [8], which is usually for gray face recognition. For representing the color object-view images, we extend the 2D to ND tensor, and use supervised neighborhood embedding analysis for sub-manifold learning. The proposed ND SNE is a generalization from 2D images to M^{th} order tensors. A significant advantage of this method over 2D tensor analysis is its superior convergence property while the latter is difficult to converge. For classification, we map the original tensor features into a low-dimensional feature space and use simple KNN algorithm and random forest

Manuscript received March 28, 2011.

Manuscript revised August 18, 2011.

[†]The authors are with College of Information Science and Engineering, Ritsumeikan University, Kusatsu-shi, 525–8577 Japan.

^{††}The author is with Omron Corporation, Kusatsu-shi, 525–0035 Japan.

a) E-mail: hanxhua@fc.ritsumeik.ac.jp

DOI: 10.1587/transinf.E95.D.835

for classification. Random forest classifiers were first introduced in [9] and developed further in [10]. Their recent popularity is largely due to the tracking application [11]. They have been applied to object recognition in [12], [13]. The advantage of randomized trees, as has been noted by previous authors [14], is that they are much faster in training and testing than traditional classifiers (such as SVM). They also enable different cues (such as appearance and shape) to be “effortlessly combined” [15].

The remaining parts of this paper are organized as follows. We introduce the algebra of tensor analysis and related subspace learning approaches in Sect. 2. In Sect. 3, we give the N-dimensional tensor supervised neighborhood embedding (ND TSNE), and recognition algorithm in Sect. 4. We introduce the used dataset and experimental setup, and then, report the experimental results in Sect. 5. Conclusions are given in Sect. 6.

2. Related Work

In this section, we firstly briefly introduce the tensor algebra and then review subspace-based feature extraction approaches such as PCA, LPP.

Tensors are arrays of numbers which transform in certain ways under coordinate transformations. The order of a tensor $\mathcal{X} \in R^{N_1 \times N_2 \times \dots \times N_M}$, represented by a multi-dimensional array of real numbers, is M . An element of $\mathcal{X}$ is denoted as $\mathcal{X}_{i_1, i_2, \dots, i_M}$, where $1 \leq i_j \leq N_j$ and $1 \leq j \leq M$. In the tensor terminology, the mode- j vectors of the n th-order tensor $\mathcal{X}$ are the vectors in R^{N_j} obtained from $\mathcal{X}$ by varying the index i_j while keeping the other indices fixed. For example, the column vectors in a matrix are the mode-1 vectors and the row vectors in a matrix are the mode-2 vectors.

Definition. (Modeproduct). The tensor product $\mathcal{X}_{\times d} \mathbf{U}$ of tensor $\mathcal{X} \in R^{N_1 \times N_2 \times \dots \times N_M}$ and a matrix $\mathbf{U} \in R^{N_d \times N'}$ is the $N_1 \times N_2 \times \dots \times N_{d-1} \times N' \times N_{d+1} \times \dots \times N_M$ tensor:

$$(\mathcal{X}_{\times d} \mathbf{U})_{i_1, i_2, \dots, i_{d-1}, j, i_{d+1}, \dots, i_M} = \sum_{i_d} (\mathcal{X}_{i_1, i_2, \dots, i_{d-1}, i_d, i_{d+1}, \dots, i_M} \mathbf{U}_{i_d, j}) \quad (1)$$

for all index values. $\mathcal{X}_{\times d} \mathbf{U}$ means the mode d 's product of the tensor $\mathcal{X}$ with the matrix $\mathbf{U}$. The mode product is a special case of a contraction, which is defined for any two tensors not just for a tensor and a matrix. In this paper, we follow the definitions in Ref. [16] and avoid the use of the term “contraction”.

In tensor analysis, Principal Component Analysis (PCA) is used to extract the basis for each mode. The proposed ND TSNE approach is based on the basis idea of Locality Preserving Projection (LPP). Therefore, we simply introduce PCA, LPP and a 2D extension of LPP as the following.

(1) Principal component analysis extracts the principal eigen-space associated with a set (matrix) $\mathbf{X} = [\mathbf{x}_i]_{i=1}^N$ of training samples ($\mathbf{x}_i \in R^n$ with $1 \leq i \leq N$; N : sample number; n : dimension of the samples). Let $\mathbf{m}$ be the mean of the

N training samples, and $\mathbf{C} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$ be the covariance matrix of the $\mathbf{x}_i$. One solves the eigenvalue equation $\lambda \mathbf{u}_i = \mathbf{C} \mathbf{u}_i$ for eigenvalues $\lambda_i \geq 0$. The principal eigenspace $\mathbf{U}$ is spanned by the first K eigenvectors with the largest eigenvalues, $\mathbf{U} = [\mathbf{u}_i]_{i=1}^K$. If $\mathbf{x}_t$ is a new feature vector, then it is projected to eigenspace $\mathbf{U}$: $\mathbf{y}_t = \mathbf{U}^T (\mathbf{x}_t - \mathbf{m})$. The vector $\mathbf{y}_t$ is used in place of $\mathbf{x}_t$ for representation and classification.

(2) Locality Preserving Projection: LPP seeks a linear transformation $\mathbf{P}$ to project high-dimensional data into a low-dimensional sub-manifold that preserves the local Structure of the data. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ denotes the set representing features of N training image samples, and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N] = [\mathbf{P}^T \mathbf{x}_1, \mathbf{P}^T \mathbf{x}_2, \dots, \mathbf{P}^T \mathbf{x}_N]$ denotes the samples feature in transformed subspace. Then, the linear transformation $\mathbf{P}$ can be obtained by solving the following minimization problem with some constraints, which will be given later:

$$\min_{\mathbf{P}} \sum_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} = \min_{\mathbf{P}} \sum_{ij} \|\mathbf{P}^T \mathbf{x}_i - \mathbf{P}^T \mathbf{x}_j\|^2 W_{ij} \quad (2)$$

where W_{ij} evaluate the local structure of the image space. It can be simply defined as follows:

$$W_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is among the } k \text{ nearest neighbors of } \mathbf{x}_j \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

By simple algebra formulation, the objective function can be reduced to:

$$\begin{aligned} & \frac{1}{2} \sum_{ij} (\mathbf{P}^T \mathbf{x}_i - \mathbf{P}^T \mathbf{x}_j)^2 W_{ij} \\ &= \sum_i \mathbf{P}^T \mathbf{x}_i D_{ii} \mathbf{x}_i^T \mathbf{P} - \sum_{ij} \mathbf{P}^T \mathbf{x}_i W_{ij} \mathbf{x}_j^T \mathbf{P} \\ &= \mathbf{P}^T \mathbf{X} (\mathbf{D} - \mathbf{W}) \mathbf{X}^T \mathbf{P} = \mathbf{P}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{P} \end{aligned} \quad (4)$$

where each column $\mathbf{P}_i$ of the LPP linear transformation matrix $\mathbf{P}$ can not be zero vector, and a constraint is imposed as follows:

$$\mathbf{Y}^T \mathbf{D} \mathbf{Y} = \mathbf{I} \Rightarrow \mathbf{P}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{P} = \mathbf{I} \quad (5)$$

where $\mathbf{I}$ in constraint term $\mathbf{P}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{P} = \mathbf{I}$ or $\mathbf{Y}^T \mathbf{D} \mathbf{Y} = \mathbf{I}$ is an identity matrix. $\mathbf{D}$ is a diagonal matrix; its entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$, $D_{ii} = \sum_j W_{ij}$; $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix [5]. Matrix $\mathbf{D}$ provides a natural measure on the data samples. The bigger the value D_{ii} (corresponding to $\mathbf{y}_i$) is, the more importance is $\mathbf{y}_i$. The constraint for the sample $\mathbf{y}_i$ in $\mathbf{Y}^T \mathbf{D} \mathbf{Y} = \mathbf{I}$ is $D_{ii} * \mathbf{y}_i^T \mathbf{y}_i = 1$, which means that the more importance (D_{ii} is larger) the sample $\mathbf{y}_i$ is, the smaller the value of $\mathbf{y}_i^T \mathbf{y}_i$ is. Therefore, the constraint $\mathbf{Y}^T \mathbf{D} \mathbf{Y} = \mathbf{I}$ will try to make the important point (has density distribution around the important point) near the origin of the projected subspace. Then,

the density region near the origin of the projected subspace includes most of the samples, which can make the objective function in Eq. (2) as small as possible, and at same time, can avoid the trivial solution $\|\mathbf{P}_i\|^2 = 0$ for the transformation matrix $\mathbf{P}$.

Then, The linear transformation $\mathbf{P}$ can be obtained by minimizing the objective function under constraint $\mathbf{P}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{P} = \mathbf{I}$:

$$\underset{\mathbf{P}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{P} = \mathbf{I}}{\operatorname{argmin}} \mathbf{P}^T \mathbf{X} (\mathbf{D} - \mathbf{W}) \mathbf{X}^T \mathbf{P} \quad (6)$$

Finally, the minimization problem can be converted to solve a generalized eigenvalue problem as follows:

$$\mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{P} = \lambda \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{P} \quad (7)$$

In Face recognition application, He et al [8] extended LPP method into 2D dimension analysis, named as Tensor Subspace Analysis (TSA). TSA can directly deal with 2D gray images, and achieved better recognition results than the conventional 1D subspace learning methods such as PCA, LDA and LPP. However, for object recognition, color information also plays an important role for distinguishing different objects. Then, in this paper, we extend LPP to ND tensor analysis, which can directly deal with not only 3D Data but also ND data structure. At the same time, in order to obtain stable transformation tensor basis, we regularize a term in the proposed TSNE objective function for object recognition, which is introduced in Sect. 3 in detail.

3. N-Dimensional Tensor Supervised Neighborhood Embedding

In order to model N -Dimensional data without rasterization, tensor representation is proposed and analyzed for feature extraction or modeling. In this section, we propose a ND tensor supervised neighborhood embedding to not only extract discriminant feature but also preserve the local geometrical and topological properties in same category for recognition. The proposed approach decompose each mode of tensor with objective function, which consider neighborhood relation and class label of training samples.

Suppose we have ND tensor objects $\mathcal{X}$ from C classes. The c^{th} class has n^c tensor objects and the total number of tensor objects is n . Let $\mathcal{X}_{i_c} \in R^{N_1 \times N_2 \times \dots \times N_L}$ ($i_c = 1, 2, \dots, n^c$) be the i^{th} object in the c^{th} class. For color object image tensor, L is 3, N_1 is the row number, N_2 is the column number, and N_3 is the color space components ($N_3=3$). We can build a nearest neighbor graph $\mathcal{G}$ to model the local geometrical structure and label information of $\mathcal{X}$. Let $\mathbf{W}$ be the weight matrix of $\mathcal{G}$. A possible definition of $\mathbf{W}$ is as follows:

$$W_{ij} = \begin{cases} \exp^{-\frac{\|\mathcal{X}_i - \mathcal{X}_j\|^2}{\tau}} & \text{if sample } i \text{ and } j \text{ is in same class} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where $\|\mathcal{X}_i - \mathcal{X}_j\|^2$ means Euclidean distance of two tensor,

which is the summation square root of all corresponding elements between $\mathcal{X}_i$ and $\mathcal{X}_j$, and $\|\bullet\|$ means l_2 norm in our paper.

Let $\mathbf{U}_d$ be the d -mode transformation matrices (Dimension: $N_d \times N'_d$). A reasonable transformation respecting the graph structure can be obtained by solving the following objective functions:

$$\min_{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_L} \frac{1}{2} \sum_{ij} \|\mathcal{X}_{i \times 1} \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_L \mathbf{U}_L - \mathcal{X}_{j \times 1} \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_L \mathbf{U}_L\|_2 W_{ij} \quad (9)$$

where $\mathcal{X}_i$ is the tensor representation of the i^{th} sample; $\mathcal{X}_{i \times 1} \mathbf{U}_1$ means the mode 1's product of the tensor $\mathcal{X}_i$ with the matrix $\mathbf{U}_1$, and $\mathcal{X}_{i \times 1} \mathbf{U}_1 \times_2 \mathbf{U}_2$ means the mode 2's product of the tensor $\mathcal{X}_{i \times 1} \mathbf{U}_1$ with the matrix $\mathbf{U}_2$, and so on. The above objective function incurs a heavy penalty if neighboring points of same class $\mathcal{X}_i$ and $\mathcal{X}_j$ are mapped far apart. Therefore, minimizing it is an attempt to ensure that if $\mathcal{X}_i$ and $\mathcal{X}_j$ are "close", then $\mathcal{X}_{i \times 1} \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_L \mathbf{U}_L$ and $\mathcal{X}_{j \times 1} \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_L \mathbf{U}_L$ are "close" as well. Let $\mathcal{Y}_i = \mathcal{X}_{i \times 1} \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_L \mathbf{U}_L$ with dimension $N_1 \times N_2 \times \dots \times N_L$, and $(\mathcal{Y}_i)^d = (\mathcal{X}_{i \times 1} \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d$ with dimension: $N_d \times (N_1 \times N_2 \times \dots \times N_{d-1} \times N_{d+1} \times \dots \times N_L)$ is the d -mode extension of tensor $\mathcal{Y}_i$, which is a 2D matrix. Let $\mathbf{D}$ be a diagonal matrix, $D_{ii} = \sum_j W_{ij}$. Since $\|\mathbf{A}\|^2 = \operatorname{tr}(\mathbf{A} \mathbf{A}^T)$, we see that

$$\begin{aligned} & \frac{1}{2} \sum_{ij} \|\mathcal{X}_{i \times 1} \mathbf{U}_1 \cdots \times_L \mathbf{U}_L - \mathcal{X}_{j \times 1} \mathbf{U}_1 \cdots \times_L \mathbf{U}_L\|^2 W_{ij} \\ &= \frac{1}{2} \sum_{ij} \operatorname{tr}((\mathcal{Y}_i)^d - (\mathcal{Y}_j)^d)((\mathcal{Y}_i)^d - (\mathcal{Y}_j)^d)^T W_{ij} \\ &= \operatorname{tr} \left(\sum_i D_{ii} (\mathcal{Y}_i)^d ((\mathcal{Y}_i)^d)^T - \sum_{ij} W_{ij} (\mathcal{Y}_i)^d ((\mathcal{Y}_j)^d)^T \right) \\ &= \operatorname{tr} \left(\sum_i D_{ii} (\mathbf{U}_d^T (\mathcal{X}_{i \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d \right. \\ & \quad \left. ((\mathcal{X}_{i \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d)^T \mathbf{U}_d \right. \\ & \quad \left. - \sum_{ij} W_{ij} (\mathbf{U}_d^T (\mathcal{X}_{i \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d \right. \\ & \quad \left. ((\mathcal{X}_{j \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d)^T \mathbf{U}_d \right) \\ &= \operatorname{tr} \left(\mathbf{U}_d^T \left(\sum_i D_{ii} ((\mathcal{X}_{i \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d \right. \right. \\ & \quad \left. \left. ((\mathcal{X}_{i \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d)^T \right. \right. \\ & \quad \left. \left. - \sum_{ij} W_{ij} ((\mathcal{X}_{i \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d \right. \right. \\ & \quad \left. \left. ((\mathcal{X}_{j \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d)^T \right) \mathbf{U}_d \right) \\ &= \operatorname{tr}(\mathbf{U}_d^T (\mathbf{D}_d - \mathbf{S}_d) \mathbf{U}_d) \end{aligned} \quad (10)$$

where $\mathbf{D}_d = \sum_i D_{ii} (\mathcal{X}_{i \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d ((\mathcal{X}_{i \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d)^T$ and $\mathbf{S}_d = \sum_{ij} W_{ij}$

Table 1 The flowchart of N -dimension tensor supervised neighborhood embedding (ND TSNE).

Algorithm 1: ND tensor supervised neighborhood embedding
Input: Tensor objects X_i^c from C classes, X_i^c denotes the i^{th} tensor object in the c_{th} class Graph-based weights: Building nearest neighbor graph in same class and calculate the graph weight $\mathbf{W}$ according to Eq. 8 and $\mathbf{D}$ from $\mathbf{W}$ Initialize: Randomly initialize $\mathbf{U}_r^d \in \mathbb{R}^{N_d}$ for $d = 1, 2, \dots, L$ for $t=1:T$ (Iteration steps) or until converge do for $d=1:L$ (Iteration steps) do • Calculate $\mathbf{D}_d$ and $\mathbf{S}_d$ assuming $\mathbf{U}_i (i = 1, 2, \dots, d-1, d+1, \dots, L)$ fixed. • Solve the minimizing problem: $\min_{\mathbf{U}_d} \text{tr}(\mathbf{U}_d^T (\mathbf{D}_d - \mathbf{S}_d) \mathbf{U}_d)$ with eigenspace analysis end for end for output: the ND TSNE tensor $\mathcal{T}_j = \mathbf{U}_1 \times \mathbf{U}_2 \times \dots \times \mathbf{U}_L, j = 1, 2, \dots, (N'_1 \times N'_2 \times \dots \times N'_L)$.

$(X_{i \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d ((X_{j \times 1} \mathbf{U}_1 \cdots \times_{d-1} \mathbf{U}_{d-1} \times_{d+1} \mathbf{U}_{d+1} \cdots \times_L \mathbf{U}_L)^d)^T$. In optimization procedure of each mode, we also impose a constraint to achieve the transformation matrix (such as $\mathbf{U}_d$ in mode d) as the following:

$$\mathbf{U}_d^T \mathbf{Y}^d \mathbf{D} (\mathbf{Y}^d)^T \mathbf{U}_d = \mathbf{I} \Rightarrow \mathbf{U}^T \mathbf{D}_d \mathbf{U} = \mathbf{I} \quad (11)$$

For the optimization problem of all modes, we adopt an alternative least square (ALS) approach. In ALS, we can obtain the optimal base vectors on one mode by fixing the base vectors on the other modes and cycle for the remaining variables. The d -mode transformation matrix $\mathbf{U}_d$ can be achieved by minimizing the following cost function:

$$\underset{\mathbf{U}_d}{\text{argmin}} \mathbf{U}_d^T (\mathbf{D}_d - \mathbf{S}_d) \mathbf{U}_d \quad (12)$$

In order to achieve the stable solution, we firstly regularize the symmetric matrix $\mathbf{D}_d$ as $\mathbf{D}_d = \mathbf{D}_d + \alpha \mathbf{I}$ (α is a small value, $\mathbf{I}$ is an identity matrix of same size with the matrix $\mathbf{D}_d$). Then, the minimization problem for obtaining d -mode matrix can be converted to solve a generalized eigenvalue problem as follows:

$$(\mathbf{D}_d - \mathbf{S}_d) \mathbf{U}_d = \lambda \mathbf{D}_d \mathbf{U}_d \quad (13)$$

We can select the corresponding generalized eigenvectors with the first N'_d smaller eigenvalues in Eq. (13), which can minimize the objective function in Eq. (12). However, the eigenvectors with the smallest eigenvalues are usually unstable. Therefore, we convert Eq. (13) into:

$$\mathbf{S}_d \mathbf{U}_d = (1 - \lambda) \mathbf{D}_d \mathbf{U}_d \Rightarrow \mathbf{S}_d \mathbf{U}_d = \beta \mathbf{D}_d \mathbf{U}_d \quad (14)$$

The corresponding generalized eigenvectors with the first N'_d smaller eigenvalues λ in Eq. (13) means those with the first N'_d larger eigenvalues $\beta(1 - \lambda)$ in Eq. (14). Therefore, the corresponding generalized eigenvectors with the first N'_d larger eigenvalues can be selected for minimizing the objective function in Eq. (12). The details algorithm of ND TSNE are listed in Table 1. In ND TSNE algorithm, we need to

decide the retained number of the generalized eigenvectors (mode dimension) for each mode. Usually, the dimension numbers in most discriminant tensor analysis methods are decided empirically or according to applications [7], [17]. In our experiments, we retain different dimension numbers for different modes, and do recognition for objects and object categories. The recognition accuracy with varied dimensions in different modes are also given in the experiment part. The dimension numbers is decided empirically in the compared results with the state-of-art algorithms.

After obtaining the ND TSNE basis of each mode, we can project each tensor object into these TSNE tensors. For classification, the projection coefficients can represent the extracted feature vectors and can be inputted into any other classification algorithm. In our work, excepting Euclidean distance as KNN ($k=1$) classifier, we also use random forest for classification with the TSNE tensor coefficients, which is described in the following section.

4. Recognition Algorithms

In this paper, In order to prove the discriminant properties of the features in the proposed ND Tensor Supervised Neighborhood Embedding space, we also use the simple k -nearest neighbors algorithm (K-NN) [18] to recognize unknown samples. Then, random forest classifiers are used for recognition with ND TSNE-domain feature, which usually is stable for classification.

(1) K-NN algorithm: In pattern recognition, the KNN is a method for classifying objects based on closest training examples in the feature space. KNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k -nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor. In our experiments, the training sample number is very small, therefore, we select K as one ($k=1$). For simplicity, we use Euclidean distance as the distance metric in KNN algorithm. Therefore, in our experiments, we just use the simple Euclidean distance and the nearest neighbor in known database (Training sets) to recognize unknown samples.

(2) Random Forest classifier: Random forest is a multi-way classifier, which consists of a number of trees with each tree grown using some form of randomization. The leaf nodes of each tree are labeled by estimates of the posterior distribution over the image classes. Each internal node contains a test that best splits the space of data to be classified. An image is classified by sending it down every tree and aggregating the reached leaf distributions. Randomness can be injected at two points during training: in sub-sampling the training data so that each tree is grown using a different subset; and in selecting the node tests.

Growing the trees. The trees here are binary and are constructed in a top-down manner. The binary test at each node can be chosen in one of two ways: (i) randomly, i.e. data independent; or (ii) by a greedy algorithm which picks the test that best separates the given training examples. “Best” here is measured by the information gain:

$$\Delta E = \sum_i \frac{|Q_i|}{Q} E(Q_i) \quad (15)$$

caused by partitioning the set Q of examples into two subsets Q_i according to the given test. Here $E(q)$ is the entropy $-\sum_{j=1}^N p_j \log 2(p_j)$ with p_j the proportion of examples in q belonging to class j , and $|\cdot|$ the size of the set. The process of selecting a test is repeated for each nonterminal node, using only the training examples falling in that node. The recursion is stopped when the node receives too few examples, or when it reaches a given depth.

Learning posteriors. Suppose that T is the set of all trees, C is the set of all classes and L_c is the set of all leaves for a given tree. During the training stage the posterior probabilities ($P_{t,l}(Y(I) = c)$) for each class $c \in C$ at each leaf node $l \in L_c$, are found for each tree $t \in T$. These probabilities are calculated as the ratio of the number of images I of class c that reach l to the total number of images that reach l . $Y(I)$ is the class-label c for image I .

Classification. The test image is passed down each random tree until it reaches a leaf node. All the posterior probabilities are then averaged and the argmax is taken as the classification of the input image.

5. Experiments

5.1 Database

We use two datasets to validate the proposed NF TSNE for view-based object categorization and recognition. The first is the Columbia COIL-100 image library [5]. It consists of color images of 72 different views of 100 objects. The images were obtained by placing the objects on a turntable and taking a view every 5° . The objects have a wide variety of complex geometric and reflectance characteristics. Figure 1 (a) shows some sample images from COIL-100. The second dataset is the ETH Zurich CogVis ETH-80 dataset [3]. This dataset was setup by Leibe and Schiele [3] to explore the capabilities of different features for object class recognition. In this dataset, eight object categories including apple, pear, tomato, cow, dog, horse, cup and car have been collected. There are 10 different objects spanned large intra-class variance in each category. Each object has 41 images from viewpoints spaced equally over the upper viewing hemisphere. On the whole we have 3280 images, 41 images for each object and 10 objects for each category. Figure 1 (b) shows some sample images from ETH-80.

5.2 Experimental Setup

We take different experimental setup in these two datasets.

Fig. 1 Some sample images. (a) Sample images from two different objects with 4 view points in COIL-100. (b) Sample images from 2 objects with 2 views in three categories (apple, car, cup) in ETH-80.

For COIL-100, the objective is to be discriminated between the 100 individual objects. In most previous experiments on object recognition using COIL-100, the number of views used as training set for each object varied from 36 to 4. When 36 views are used for training, the recognition rate using SVM was reported approaching 100% [5]. In practice, however, only very few views of an object are available. In our experiment, in order to compare experimental results with those in [7], we follow the experiment setup, which used only 4 viewpoint images for training and the rest 68 views for testing. In total it is equivalent to 400 images for training and 6800 images for testing. The error rate is the overall error rate over 100 objects. The 4 training viewpoints are sampled evenly from the 72 viewpoints, which can capture enough variance on the change of viewpoint for tensor learning, and then there are 18 available groups of 4 evenly views for training. In order to obtain the evaluation recognition rate under change of training sample groups, we did 18 runs with different groups of 4 views for training, and evaluate recognition performances with average rate, maximum and minimum deviation from average.

For ETH-80, it aims to discriminate between the 8 object categories. Most previous experiments using ETH-80 dataset all adopted leave-one-object-out cross-validation. The training set consists of all views from 9 objects from each category. The testing set consists of all views from the

Fig. 2 Compared average recognition rates with different classifiers corresponding to different mode components on COIL-100 dataset.

remaining object from each category. In this setting, objects in the testing set have not appeared in the training set, but those belonging to the same category have. Classification of a test image is a process of labeling the image by one of the categories. Reported results are based on average error rate over all 80 possible test objects [19]. Similar to the above, instead of taking all possible views of each object in a training set, we take only 5 views of each object as training data. By doing so we have decreased the number of the training data to 1/8 of that used in [19], [20]. The testing set consists of all the views of an object. In all our experiments, each training image is 128×128 and represented as a 3rd tensor object using raw RGB data. The dimension of each tensor object is $128 \times 128 \times 3$.

5.3 Results

On each dataset, we carried out ND Tensor Supervised Neighborhood Embedding (ND TSNE) analysis with training samples for obtaining the basis tensor $\mathcal{T}_j = \mathbf{U}_1 \times \mathbf{U}_2 \times \cdots \times \mathbf{U}_L$, $j = 1, 2, \dots, (N'_1 \times N'_2 \times \cdots \times N'_L)$, and then project color image tensor onto the basis tensor for obtaining the tensor coefficients. Finally, the tensor coefficients are used for recognition with simple KNN classifier (Because the training sample number for each object is small, we select $k=1$ in the KNN classifier) only denoted as ND TSNE, and random forest classifier denoted as ND TSNE-RF. Experimental setup is same as given in Sect. 5.2. The average recognition or error rates are the means of all 18 runs for varied dimensions on different modes on COIL-100 dataset.

The compared experimental results with different classifiers are shown in Fig. 2 corresponding to different mode components number (We keep all 3 components for color mode in all of our experiments). From Fig. 2, it is obvious that the average recognition rates of 18 runs with random forest classifier (denoted as ND TSNE-RF) are much better than those with Euclidean distance (simple KNN with $k=1$) for most of different components. In order to evaluate performance variation with different training groups, Fig. 3 also gives the compared maximum, average and minimum recognition rates of the 18 done runs on different mode components using our proposed ND TSNE analysis and the previous mentioned two classifiers, where the same color represents the same recognition rate. From the experimental results of different runs, the recognition rate variations from the average ones are about $\pm 2-3\%$, and then no large variation is appeared when selecting different training groups.

Because, our proposed ND TSNE algorithm is an extension of supervised Locality Preserving Projections (LPP), we also did recognition experiments on COIL-100 database with the same experiment setup given in Sect. 5.2 using supervised LPP, which firstly vectorizing the color image into a vector. The average recognition rates of the 18 experimental runs with different retained components (dimensions) are shown in Fig. 4 using KNN ($k=1$) and random forest classifiers. From Fig. 4, the maximum recognition can be achieved when the retained component number is about 20 for both KNN (82.25% with dimension 20) and random forest (84.12% with dimension 20) classifiers, which are denoted as LPP+KNN and LPP+RF, re-

Fig. 3 Recognition rate variation from the average ones. In our experiments, we did 18 runs with different training images for COIL-100 database, and then 18 recognition rates can be achieved for each of selected component numbers. Therefore, the average, maximum, minimum recognition rates can be calculated on varied retained component numbers as shown in this figure with (a) Euclidean distance classifier; (b) Random forest classifier.

spectively. Next, in order to validate the discriminant properties of the proposed ND TSNE tensor analysis, we give the compared recognition rates using the two used classifier with the conventional algorithm [2], [4], [5], [7], [13] in Table 2 on COIL-100 dataset. The best result with the same experiment setup (400 training samples and 6800 test samples) on COIL-100 is reported in [7], in which the average recognition rate using tensor LDA and AdBoost classifier (DTROD+AdBoost) is 84.5%, and the recognition rate of the tensor LDA and simple Euclidean distance (DTROD) in

Fig. 4 Recognition rates on COIL-100 dataset with supervised LPP method.

Table 2 Overall recognition error rates on COIL-100. The results of our proposed approach are compared with that of PCA+LDA which vectorizes the 3rd tensor objects and using a nearest neighbor classifier, SVM (linear&non-linear) [4], [5], Columbia 3D object recognition system [2], SNoW [13] and DTROD, DTROD+AdaBoost [7].

Method	Error rate (%)
PCA+LDA	32
SVM (linear)	21.5
SVM (Non-linear)	17.7
Columbia	22.5
SNoW	18.5
DTROD	20.3
DTROD+AdBoost	15.5
LPP+KNN	17.75
LPP+RF	15.86
2D TSNE using gray image	33.66
ND TSNE (Ours)	14.57 ± 1.45
ND TSNE-RF (Ours)	8.766 ± 3.14

[7] (like our KNN method with $k=1$) is 79.7. The retained component dimensions in the conventional subspace learning methods (PCA+LDA and DTROD) is decided empirically, and the best recognition rates among those achieved by varying component number are given in Table 2. The conventional supervised LPP method can achieve 84.12% recognition rate with random forest classifier. 2D TSNE approach with gray image tensor, which firstly transform 3D RGB image in 2D gray image, can obtain 66.34% recognition rate by retaining 20 row and column mode dimensions, respectively, as shown in Table 2. However, our proposed ND TSNE approach can achieve about 85.43% with simple distance classifier (KNN), and 91% average recognition rate with random forest classifier. In order to validate variance of recognition rates for different object categories, Fig. 5 gives the confusion table by the proposed ND TSNE and random forest classifier (ND TSNR-RF) on Coil-100 dataset. From Fig. 5, it can be seen that most object categories can achieve acceptable recognition rates.

For the ETH-80 dataset, we also do the experiment similar to COIL-100. The experimental setup is explained in

Fig. 5 Confusion table using ND TSNR-RF on COIL-100 dataset.

Fig. 6 Average recognition rates with random forest classifier corresponding to different mode components on ETH-80 dataset.

Sect. 5.2. With different mode-dimension features (retained row components: 3–15; Column components: 3–15; Color components: 3), the overall recognition rates using random forest classifier is given in Fig. 5. From Fig. 6, it can be seen that when the retained row and column mode component number is small, the recognition rates can be improved with increasing row or column components. However, the recognition rates can only improved a little or keep stable even

Table 3 Overall recognition error rates on ETH-80. RSW denotes random subwindow method [20] and LS denotes the results from Leibe and Schiele [19]. ND TSNE means the proposed ND TSNE using simple Euclidean distance; ND TSNE-RF means the proposed ND TSNE using random forest classifier, where mode dimension is $7*7*3$.

Methods	#training	#testing	Error rate(%)
PCA+LDA	360	328	37.0
DTROD	360	328	30.0
DTROD+AdBoost	360	328	24.0
RSW	2925	328	26.0
LS	2925	328	35.0
ND TSNE (Ours)	360	328	19.52
ND TSNE-RF (Ours)	360	328	16.67

with increasing components when a amount row and column components are used. According to our experiments, the recognition rates are 80.48% and 83.33% with KNN and random forest classifiers using the retained components $7*7*3$, where the digital number means the retained component numbers of row, column and color modes, respectively. The recognition rates change a little with component increasing from the retained $7*7*3$ ones. Therefore, in the compared results between our proposed ND TSNE and the state of the art approach shown in Table 3, the retained dimension for row, column and color modes are $7*7*3$. From Table 3, It can be seen that our proposed approach can greatly improve the overall recognition rate compared with the state of the art method (from 60–76% to about 84%).

6. Conclusions

In this paper, we proposed an N-dimensional Tensor supervised Neighborhood Embedding (ND TSNE) approach for

view-based object categorisation and recognition. The ND RSNE tensor approach avoids the need to vectorize 3D color images or 2D grayscale images into high-dimensional feature vectors and has a good property in convergence. Our experiment shows that the proposed ND RSNE analysis in multi-order tensor space outperforms conventional supervised subspace learning methods such as LDA in a vectorised/flattened feature space and also the ND tensor extension of LDA for view-based object categorisation and recognition. Using ND TSNE tensor analysis project coefficients as features for a random forest based classification, we demonstrated comparable recognition results to the state of the art on the COIL-100 dataset and much better categorization results on the ETH-80 dataset using only a much smaller number of training samples. Although the ETH-80 dataset contains big intra-class variance, the ND RSNE tensor analysis approach still achieved much better results than those of the state of the art approaches.

References

- [1] T. Poggio and S. Edelman, "A network that learns to recognize three-dimensional objects," *Nature*, vol.343, pp.263–266, 1990.
- [2] H. Murase and S.K. Nayar, "Visual learning and recognition of 3-D objects from appearance," *Int. J. Comput. Vis.*, vol.14, pp.5–24, 1995.
- [3] S. Gong, S. McKenna, and J. Collins, "An investigation into face pose distributions," *AFGR*, pp.265–270, 1996.
- [4] D. Roobaert and M.V. Hulle, "View-based 3D object recognition with support vector machines," *IEEE International Workshop on Neural Networks for Signal Processing*, pp.77–84, 1999.
- [5] M. Pontil and A. Verri, "Support vector machines for 3D object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp.637–646, 1998.
- [6] M.-H. Yang, D. Roth, and N. Ahuja, "Learning to recognize 3D objects with SNoW," *Proc. Sixth European Conference on Computer Vision*, pp.439–454, 2000.
- [7] Y. Wang and S. Gong, "Tensor discriminant analysis for view-based object recognition," *Proc. ICPR*, pp.33–36, 2006.
- [8] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," *Proc. NIP*, 2005.
- [9] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol.9, pp.1545–1588, 1997.
- [10] L. Breiman, "Random forests," *Mach. Learn.*, vol.45, pp.5–32, 2001.
- [11] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.28, no.9, pp.1465–1479, 2006.
- [12] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," *NIPS*, pp.985–992, 2007.
- [13] M. Ozuysal, P. Fua, and V. Lepetit, "Fast keypoint recognition in ten lines of code," *CVPR*, pp.1–8, 2007.
- [14] J. Winn and J. Shotton, "The layout consistent random field for recognizing and segmenting partially occluded objects," *CVPR*, pp.37–44, 2006.
- [15] J. Winn and A. Criminisi, "Object class recognition at a glance," *CVPR*, video track, 2006.
- [16] L.D. Lathauwer, *Signal processing based on multilinear algebra*, Ph.D. Thesis, Katholieke Universiteit Leuven, 1997.
- [17] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Trans. Image Process.*, vol.16, no.1, pp.212–220, Jan. 2007.
- [18] F. Nigsch, A. Bender, B. van Buuren, J. Tissen, E. Nigsch, and J.B.O. Mitchell, "Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization," *J. Chemical Information and Modeling*, vol.46, no.6, pp.2412–2422, 2006.
- [19] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," *CVPR*, pp.409–415, June 2003.
- [20] R. Marée, P. Geurts, J. Piater, and L. Wehenkel, "Random subwindows for robust image classification," *CVPR*, vol.1, pp.34–40, June 2005.

Xian-Hua Han received a B.E. degree from ChongQing University, ChongQing, China, a M.E. degree from ShanDONG University, Ji-Nan, China, a D.E. degree in 2005, from the University of Ryukyus, Okinawa, Japan. She is now a post-doctoral researcher at Ritsumeikan University, Kusatsu, Japan. Her current research interests include image processing and analysis, feature extraction and pattern recognition.

Yen-Wei Chen received a B.E. degree in 1985 from Kobe University, Kobe, Japan, a M.E. degree in 1987, and a D.E. degree in 1990, both from Osaka University, Osaka, Japan. From 1991 to 1994, he was a Research Fellow with the Institute of Laser Technology, Osaka. From Oct. 1994 to Mar. 2004, he was an associate professor and a professor with the Department of Electrical and Electronics Engineering, University of the Ryukyus, Okinawa, Japan. He is currently a professor with the College of Information Science and Engineering, Ritsumeikan University, Japan and a professor with the Institute for Computational Science and Engineering, Ocean University of China, China. He is an Overseas Assessor of the Chinese Academy of Science and Technology, an associate Editor of the *International Journal of Image and Graphics (IJIG)*, an Editorial Board member of the *International Journal of Knowledge-Based Intelligent Engineering Systems* and an Editorial Board member of the *International Journal of Information*. His research interests include intelligent signal and image processing, radiological imaging and soft computing. He has published more than 100 research papers in these fields. Dr. Chen is a member of the IEEE and IEE (Japan).

Xiang Ruan received B.E. degree from Shanghai Jiaotong University in 1997, and the M.E. and Ph.D. degrees from Osaka City University in 2001 and 2004. Now, he is a research engineer of OMRON Corporation. His current research interests include computer vision, machine learning and image processing. He is a member of IEEE.