ELSEVIER

Neurocomputing 31 (2000) 15–28

NEUROCOMPUTING

www.elsevier.com/locate/neucom

A modular network scheme for unsupervised 3D object recognition

Satoshi Suzuki[a,b,*], Hiroshi Ando[a]

[a] *ATR Human Information Processing Research Laboratories, Soraku-gun, Kyoto 619-0288 Japan*
[b] *NTT Communication Science Laboratories, Soraku-gun, Kyoto 619-0237 Japan*

Received 23 July 1997; accepted 29 April 1999

Abstract

This paper presents an unsupervised learning scheme for recognizing 3D objects from their 2D projected images. The scheme consists of a mixture of nonlinear autoencoders which can compress various views of 3D objects into representations that indicate the view direction. We evaluate the performance of the proposed modular network scheme through simulations using 3D wire-frame objects and discuss its related issues on object representations in the primate visual cortex.

Keywords: Unsupervised learning; 3D object recognition; Modular networks; Autoencoders

1. Introduction

The human visual system can recognize 3D (three-dimensional) objects from their 2D (two-dimensional) retinal images even though the images will significantly vary as the viewpoint changes. An important problem in 3D object recognition is therefore how to accomplish view invariant recognition. The traditional approach is to recover the 3D shape and to construct structural descriptions of 3D volumetric primitives in the object-centered coordinate frames ([14], for example). Nevertheless, due to the complexity of 3D shape processing, the reliable construction of volumetric primitives is often difficult and time-consuming. Alternatively, recent computational studies have explored view-based approaches, where 3D objects are recognized more directly from their 2D projected views ([16,18], for example). In these approaches, a limited

* Corresponding author.
E-mail address: satoshi@cslab.kecl.ntt.co.jp (S. Suzuki)

0925-2312/00/$ - see front matter
PII: S0925-2312(99)00148-4

number of view examples are learned to achieve view invariant recognition through a generalization property of the trained systems. View-based models are attractive not only for their computational efficiency and simplicity, but also for their biological relevance. Psychophysical studies have indicated that representations of 3D objects in the human visual system are viewpoint-specific [2,5]. In addition, recent electrophysiological experiments on behaving monkeys suggest that the primate inferotemporal cortex (IT) employs viewer-centered object representations [12,13].

While most existing view-based models for 3D object recognition employ supervised learning, this paper focuses on unsupervised aspects of object recognition. In the real world, a given view is not always labeled explicitly as belonging to an object. Even when an identity for different views of an object is provided through motion, only a fraction of all of these views is usually given at each temporal sequence. The human visual system seems to have a mechanism that can automatically cluster fragmental pieces of object views without using explicit object labels. Recent cognitive science studies have focused on the implicit learning ability of humans, where knowledge is automatically or unconsciously acquired without any explicit intention or instruction [25]. We therefore investigate a system that discovers object identities by itself solely from the input views of various 3D objects.

To achieve unsupervised object learning, we propose a modular network scheme which consists of a mixture of nonlinear autoencoders. Each autoencoder compresses a set of images of an object to form a manifold in a low-dimensional subspace. The compressed representations describe the degree-of-freedom of the input variations, which specifically represents the view direction or the pose of the 3D object. For the automatic clustering of various input views, we formulate an unsupervised modular learning algorithm, which is an extension of the adaptive mixture model [9–11]. The adaptive mixture model was devised for the competitive decomposition of a function approximation task based on maximum likelihood estimation. Unlike the original supervised formulation, the proposed unsupervised formulation does not incorporate a gating network which directly splits the input space.

There is a computational motivation for using a modular structure for unsupervised object classification. When the input data for recognition is high-dimensional, it is essential to reduce the dimensionality and extract intrinsic information from the input variations. A number of techniques have been proposed for dimensionality reduction such as principal component analysis (PCA) [4]. In general, however, how the dimensionlity is reduced depends on the input data set to be used. For example, PCA representations are greatly altered if another set of data is included in the input data. In most data compression applications, all of the available data is simply used or a set of data is manually defined. The proposed scheme, on the other hand, performs clustering and compression at the same time, i.e., while the input space is divided into subspaces, the scheme performs dimensionality reduction within each subspace. Specifically, in the case of 3D object recognition, the scheme clusters various views of 3D objects into individual object classes and each module compresses the views in each object class to form a compact subspace.

This paper is organized as follows. Section 2 describes details of the proposed unsupervised learning scheme. Section 3 examines the performance of the scheme

through computer simulations. We used synthetic 3D wire-frame objects for the simulations, which have commonly been used in computational studies [18], psychophysics [2,5], and electrophysiology [12,13]. Section 4 discusses some related computational and biological issues and concludes this paper. The earlier versions of this research were presented in [1,20,21]. This paper expounds upon these preliminary results with additional simulations on the effect of varying the number of objects.

2. The network model

2.1. The nonlinear autoencoders

Storing all available raw data in a high-dimensional input space is impractical in general, so it is essential to reduce the dimensionality and to extract the intrinsic information causing the input variations. Intrinsic information can be found if a statistical structure exists behind the data distribution. In the case of projected views of a 3D object, despite the significant image variations with changing viewpoint, the number of parameters constraining the input distribution is intrinsically limited. In fact, any rigid object transformation can be described by six parameters, three for rotation and three for translation. Such intrinsic parameters define the degree of freedom of the data distribution.

A number of techniques have been proposed for reducing the high dimensionality of the input space. The linear subspace methods among them, such as principal component analysis (PCA), also known as Karhunen–Loéve transform, are widely used for data compression, mainly due to their simplicity and practicality. Nevertheless, these methods yield only approximate dimensions when the underlying statistical structure is nonlinear in nature. In the case of projected images of 3D objects, a mapping from input views to viewpoint representations should be intrinsically nonlinear. To obtain such intrinsic dimensionality of the view distribution, we need to use more general nonlinear dimensionality reduction methods.

The proposed scheme thus exploits a nonlinear autoencoder for identifying an object. The autoencoder, or an auto-associative network, finds an identity mapping through a bottleneck in the hidden layer, that is, where the number of units in the hidden layer is smaller than the number of input and output units. Hence, the network approximates functions F and F^{-1} such that $R^n \xrightarrow{F} R^m \xrightarrow{F^{-1}} R^n$ where $m < n$. The auto-associative network compresses the input into a low-dimensional representation by eliminating redundancy in the input data distribution. If we use a five-layer perceptron network with a bottleneck in the third layer, the network can achieve a nonlinear dimensionality reduction, which is a nonlinear analogue to the principal component analysis [3,17]. A three-layer autoencoder, on the other hand, performs no better than a linear data compression specified by the principal components of the input distribution [17]. Five-layer autoencoders are simple yet powerful for nonlinear data compression. They have therefore been applied to various tasks, such as time

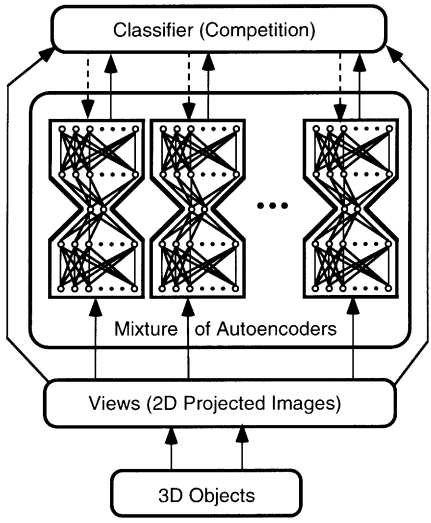


Fig. 1. Modular network architecture for unsupervised 3D object recognition. Network scheme consists of a set of five-layer auto-associative networks and competition among modular networks leads to unsupervised classification of input views.

series data compression and face recognition [3], color information encoding [27], and visual and motor information integration for grasping [26].

2.2. The modular network architecture

The network architecture that we propose for 3D object recognition consists of a set of modules, as illustrated in Fig. 1. Each module is a five-layer auto-associative network that encodes and decodes various views of an object using nonlinear mappings. If each module can learn to encode and decode various views of a different object, then each module can recover all the views of only a single object. Thus, we can identify an input view as an object by selecting the module whose output view best fits the input view. Specifically, we assume a classifier that contains n units where n is the number of modules. The output value of the ith unit of the classifier is given by the softmax function of the negative squared difference between the input and the output of the module, i.e.,

$$f_i = \exp[-\|x - y_i(x)\|^2] \Big/ \sum_j \exp[-\|x - y_j(x)\|^2], \tag{1}$$

where x and $y_i(x)$ denote the input and output vector of the ith module, respectively. Therefore, if only one of the modules has an output that best matches the input, then the output value of the corresponding unit in the classifier nearly becomes one and the output values of the other units nearly become zero.

As described in the previous section, the number of dimensions constraining the input variation of a rigid 3D object is the degree of freedom of the view distribution. Therefore, we can set the number of units in the third layer of each network to the degree of freedom. Nonetheless, there is an exception, where the encoded dimension does not correspond to the degree of freedom of the view distribution. This occurs when the intrinsic parameters are all periodic in nature. For example, to encode the pose variation of a rigid object rotating around a single axis, a two-dimensional Euclidian space is needed to describe a closed one-dimensional manifold or torus in the encoded space. The rotation of a lighting position is another example that requires an additional dimension to fully represent the input variation. To set the encoded dimension in such periodic cases, we may simply add one more dimension to the degree of freedom of the object transformation.

In the proposed network scheme, an input view can be represented by any type of image information, such as feature positions, orientations, shapes, textures or gray-level images, as long as there exists a continuous nonlinear mapping between the input views and the viewpoint representations in the hidden layer. This generic property exhibits the flexibility of the proposed scheme, because the reliable extraction of particular features depends on individual object images. Although the simulations in this paper only use the positions or the angles of image features as the inputs to the networks to show the scheme's basic properties, we are currently investigating different types of input information. Preliminary results show that the scheme can also be applied to the gray-level images of 3D objects [1,7].

2.3. The unsupervised modular learning method

The modular scheme illustrated in Fig. 1 can identify an input view as an object by selecting the network whose output view best matches the input view. To achieve this classification, each network needs to be trained with the views of a single object so that only one module can recover different views of an object. When we know which object each input view belongs to in the supervised learning, we can simply select the module that corresponds to the object and train the network by minimizing the difference between the input and the output views. This paper, on the other hand, proposes an unsupervised learning method that can automatically classify the given views without knowledge of the object identities.

To achieve unsupervised clustering, some form of competition should be introduced among the modules during the learning process so that only those modules whose output views resemble the input view can change their connection weights. Since it is difficult for each module to represent the views of many objects due to the constraint given by the bottleneck in the hidden layer, we expect that as the training proceeds each module gradually learns to recover the views of one object. Some similar modular architectures using autoencoders have been independently proposed for character or digit recognition [8,19], but unlike the unsupervised algorithm presented in this paper, these methods are all designed for supervised learning, i.e., the identity of each character or digit must be provided during the training.

We can formally derive an unsupervised learning algorithm based on the adaptive mixture model [9–11]. The adaptive mixture model is designed to partition the input space into multiple independent regions and to allocate different expert networks to learn the different input regions. An appropriate decomposition is found by forcing the expert networks to compete in learning the training data and by simultaneously training an extra gating network to find the responsibility of each expert network for each training data. Therefore, the final mixed output of the entire networks $y(x)$ is given by

$$y(x) = \sum_i g_i(x) y_i(x), \tag{2}$$

where x, $g_i(x)$, $y_i(x)$, denote the input vector, the activation of the ith output unit of the gating network, and the output vector of the ith expert network, respectively.

The adaptive mixture model can be statistically interpreted. Namely, the training patterns are assumed to be generated by multiple stochastic processes, and each process is selected with a prior probability at each pattern generation. We can assume that the prior probability is given by the gating network $g_{i.}$ if a softmax activation function is used for the output of the gating network, and that the conditional probability $p(x|i)$ is modeled by a Gaussian distribution function in the form

$$p(x|i) = \exp[-\|y^* - y_i(x)\|^2/(2\sigma^2)], \tag{3}$$

where y^* denotes the target vector and σ^2 denotes the variance of the distribution. As a result, the expert and gating networks are simultaneously trained by adjusting their connection weights so as to maximize the log likelihood function (L):

$$\ln L = \ln \sum_i g_i(x) p(x|i) = \ln \sum_i g_i(x) \exp[-\|y^* - y_i(x)\|^2/(2\sigma^2)]. \tag{4}$$

When the adaptive mixture model is applied to supervised learning tasks, we need a gating network in order to determine the final mixed output in Eq. (2). In unsupervised data clustering, on the other hand, there is no final mixed output to be computed. Therefore, we use an autoencoder for each expert network and compute the responsibility for each autoencoder based on its recovery error instead of an extra gating network. The log likelihood function for training a mixture of autoencoders is obtained by replacing g_i in Eq. (4) with the softmax function f_i described in Eq. (1). The resulting function is written in the form

$$\ln L = \ln \frac{\sum_i \exp[-\alpha\|x - y_i(x)\|^2]}{\sum_j \exp[-\|x - y_j(x)\|^2]}, \tag{5}$$

where x and $y_i(x)$ denote the input and output vector of the ith autoencoder, respectively, and $\alpha = 1 + 1/2\sigma^2$. To maximize the log likelihood function, we apply the steepest ascent method; i.e., we derive update equations for estimating the connection weights of each autoencoder by computing the derivatives of the log likelihood function with respect to the output vector $y_i(x)$ and then applying the chain rule for partial derivatives as in the case of a standard back-propagation algorithm. Since this optimization process introduces competition among the autoencoders, only

one of the autoencoders learns to recover each input data as the training of the networks proceeds. In particular, when the views of 3D objects are the input data to be learned, we limit the number of dimensions in the third layer of each autoencoder to the degree of freedom of the data distribution, as described in Sections 2.1 and 2.2. The bottleneck constraint imposed in the hidden layer attempts to prevent each network from encoding views of more than one object class. As a result, the networks gradually approach to a state where the views of different 3D object are clustered into different modules.

3. Simulations

We implemented the network scheme described in the previous section to evaluate its performance in a 3D object recognition task. This section shows unsupervised classification results and analyses of the encoded representations acquired in the hidden layer.

3.1. 3D objects and training procedures

The 3D objects that we used for our simulations were novel five-segment wire-frame objects whose six vertices were randomly selected in a unit cube, as shown in Fig. 2a. Various views of the objects were obtained by orthographically projecting the objects onto an image plane whose position covered a sphere around the objects as shown in Fig. 2b. The view position was defined by two parameters: θ and ϕ ($0 \leq \theta \leq \pi$, $0 \leq \phi < 2\pi$). In principle, the scheme can use various types of view information for the inputs to the networks as described in Section 2.2. In the following simulations, we used two types of features for the inputs: one was the x and y image coordinates of the six vertices which formed a 12-dimensional vector, and the other was the angles between the projected images of the adjoining segments which formed a four-dimensional vector.

The network scheme contains a set of modules whose number is equal to the number of objects used for the simulations. The number of units in the third layer of each module is set equal to the number of view parameters, which is two in our simulations.We varied the number of units in the second and fourth layers and found that five units were enough to achieve a reasonable clustering performance in most of the simulations. To obtain a more accurate clustering performance, the number of units was increased up to 20 for these layers. To train the network more efficiently, we can initially limit the ranges of θ and ϕ to $\pi/4$ and $\pi/2$, respectively, and gradually increase the ranges until they cover the whole sphere. During the training, objects are randomly selected among the object set and their views are randomly selected within the view ranges. The networks are trained in an *on-line* manner, i.e., the networks receive the inputs one after another during the training. To train the networks, we maximize the objective function $\ln L$ described in Eq. (5) while setting $\alpha = 100$. The steepest ascent method is used to maximize the objective function in the simulations,

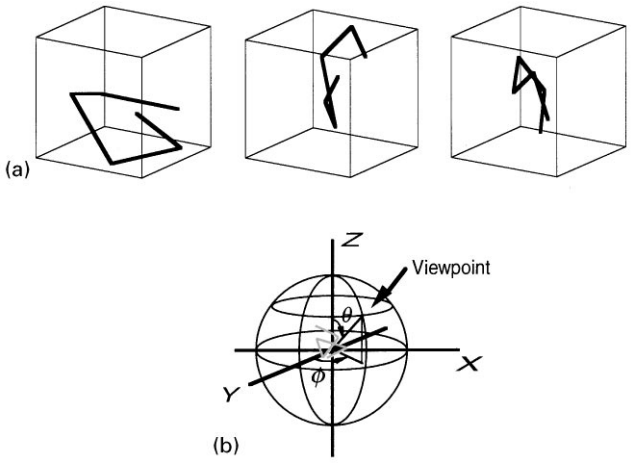


Fig. 2. (a) Examples of 3D wire-frame objects used for simulations. The 3D objects were produced by connecting six vertices randomly generated in a unit cube. (b) A viewpoint which covers a sphere around a 3D object. The position of the viewpoint is specified by two parameters (θ, ϕ). Various views of the 3D objects are obtained by orthographically projecting the objects onto an image plane.

but more efficient methods, such as the conjugate gradient method, can also be applied.

3.2. Results and analyses

We trained the networks with two different types of input features: the x and y image coordinates of the vertices and the angles formed by the adjoining segments of the wire-frame objects. Although the latter required a longer learning period, both types of features yielded a satisfactory classification performance. Thus, in the following analyses, we only show simulation results for the former type of input features.

We first examined recognition properties and internal representations of the proposed network scheme using three objects. Fig. 3 illustrates some examples of the relationship between the input views of a 3D object (Object 3) and views recovered by a trained module (Module 2). The figures show that this module can recover various views of Object 3 although the views are significantly different from one another. To confirm the recovery ability of the networks, we tested the networks using 2,500 views for each object, covering the entire view range ($0 \leq \theta \leq \pi$, $0 \leq \phi < 2\pi$). Fig. 4 illustrates the squared difference between the input views and the generated views of a module (Module 2) over the entire view range for all three objects. As shown in the figure, the recovery error is nearly zero for the views of one object (Object 3) but the error is significantly larger for the views of the two other objects. The results show that each trained module can compress and recover all the views of a single object but cannot recover the views of other objects.

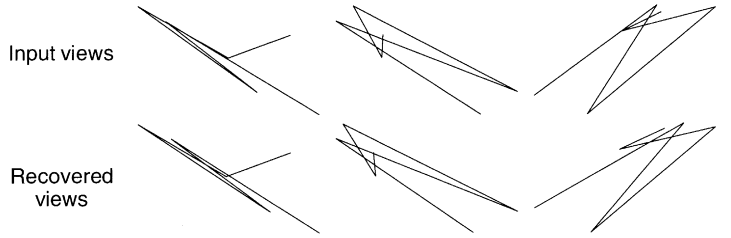


Fig. 3. Examples of input views of a 3D object (Object 3) and views recovered by a module (Module 2). The module can recover different views of the object after compressing the input views into low-dimensional representations.

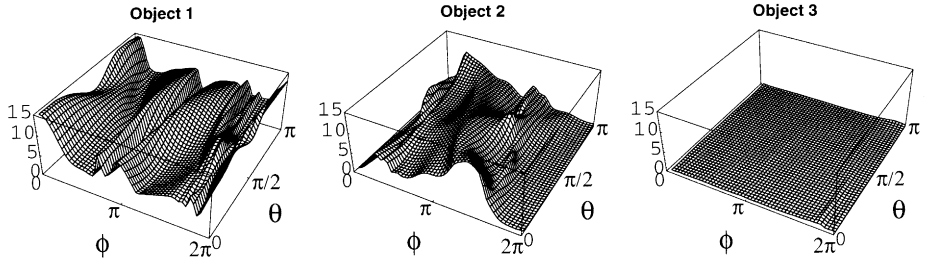


Fig. 4. Squared difference between input views and views generated from a module (Module 2) over entire view range ($0 \leq \theta \leq \pi, 0 \leq \phi < 2\pi$). Recovery error is nearly zero for views of only one object (Object 3).

The ability of each network to recover the views of a single object can be used to classify various 3D object views. Fig. 5 shows the output of the softmax classifier, f_i ($i = 1, 2, 3$), which is defined by Eq. (1) plotted over the view directions (θ, ϕ), when the inputs are all views of an object (Object 3). The output value of one unit (Unit 2) in the classifier is almost equal to one over the entire range of the view directions, while the outputs of the two other units are close to zero. Similarly, when the views of other objects (Object 1 or 2) are provided to the networks, only the output of one unit (Unit 1 or 3) is activated. Therefore, the results show that the proposed network scheme is able to cluster various views into their correct 3D object classes without any object identity provided to the networks.

We then analyzed the relationship between the compressed representations in the hidden layer of the networks and the view directions (θ, ϕ) of the input patterns. Fig. 6 illustrates a mapping from the images of a 3D object to the compressed representations of the corresponding network. Specifically, this figure plots the outputs of the two units in the third layer of the network with labels of the view directions (θ, ϕ) of the input views. For example, when the object view at $(\theta, \phi) = (0, 0)$ is provided to the network, the output values of the two hidden units are indicated by the rightmost point of a distorted 'disk' shown in the figure. Thus, if we rotate the object in the θ direction, the output values of the hidden units move towards the center of the disk;

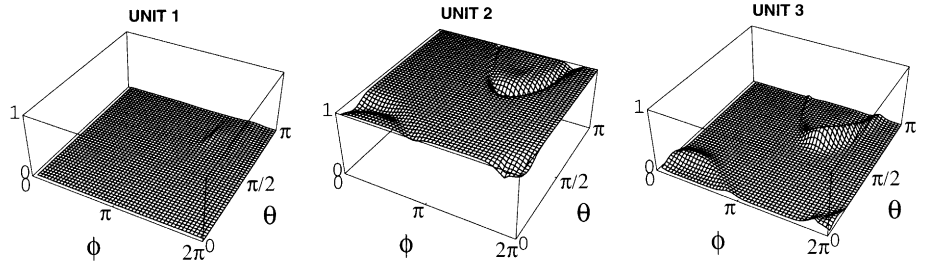


Fig. 5. Output of classifier plotted over entire view range when views of an object (Object 3) are used for the inputs. The output value of one unit (Unit 2) in the classifier is nearly one over the entire view range, while the outputs of two other units are nearly zero.

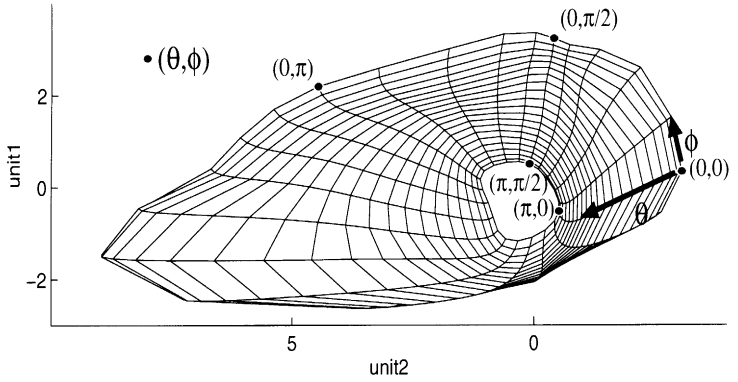


Fig. 6. Compressed representations of object views in the third layer of a network. The figure shows the relationship between the outputs of two hidden units and the view positions (θ, ϕ) of the input patterns.

if we rotate the object in the ϕ direction, the values of the hidden units move around the center of the disk. Therefore, the view direction of the 3D object is continuously and uniquely represented in the hidden layer. The result demonstrates that each autoencoder is able to extract the view direction information from the high-dimensional images and to compactly describe this intrinsic information in the low-dimensional hidden space.

We further studied the classification ability of the scheme by varying the number of objects used for the training. Specifically, we used sets of 3, 5, and 10 wire-frame objects randomly generated in a unit cube. By repeating the simulations 12 times using different sets of objects, we obtained average recognition rates of 90.2%, 90.6%, and 78.7% for 3, 5, and 10 objects, respectively. The results were compared with the performance of the conventional K-means clustering algorithm. Using the same set of objects, the K-means algorithm yielded average recognition rates of 81.4%, 73.3%, and 56.6% for 3, 5, and 10 objects, respectively. The results show that the proposed scheme achieves a significantly better performance than the K-means algorithm.

4. Discussion and conclusions

One advantage of using a mixture of nonlinear autoencoders for recognition tasks is that the mixture scheme has the potential ability to deal with complex geometrical distributions of the data set in a high-dimensional input space. Specifically, in the case of 3D object recognition, we presume that a distribution of various views of a 3D object should be highly nonlinear in the input space so that the input views of different 3D objects form manifolds that cannot be easily partitioned by hyperplanes. We investigated the linear separability of the view distributions of 3D objects to confirm this presumption. Using the same data set employed in the simulations described in the previous section, we performed supervised training of a simple perceptron that contains no hidden layer. During the training, the binary teacher signal was provided to the output units of the network, which indicates an object identity of the input view. According to the convergence theorem of a simple perceptron, the network should converge to the correct data classification within a finite-learning period if the data sets are linearly separable [15]. Nonetheless, the simulation results showed that the classification error did not approach zero, which suggests that the views of different objects cannot be simply separated by hyperplanes. This observation indicates the effectiveness of the proposed modular scheme on complex clustering problems that are not linearly separable. We also note that any view of a 3D object can be described as a linear combination of a small number of other views [24]. This property, however, is only valid for x and y image coordinates under orthographic projection, while the proposed scheme can use different types of input information (Section 2.2); in addition, there seems to be no straightforward way of using the linear combination property for unsupervised view clustering.

The proposed modular architecture may have some implications for the nature of object representations in the cortical areas of primates. Data from unit recording experiments have shown that cells in the anterior inferotemporal (IT) cortex selectively respond to moderately complex object features, and cells that respond to similar features cluster in a columnar region [6,22,23]. A recent physiological study using an optical imaging technique has revealed that in the primate inferotemporal cortex the activation spot gradually shifts as the viewing angle of a face changes; thus, representations of different views of a face may cluster in a modular region [28]. Single-unit recording experiments have also suggested that 3D objects are represented in a viewer-centered coordinate frame in the IT area [12,13]. These observations indicate that the extraction of intrinsic information on the object patterns and the clustering of extracted object representations to form multiple modular sub-regions occur in the anterior IT cortex.

However, it is still not clear how such intrinsic representations are extracted and self-organized into a modular structure in the IT area. The proposed network scheme may provide a useful framework for investigating neural mechanisms of modular organization in the cortex. In particular, effective clustering results of the proposed scheme suggest that competition among the modules as well as dimensionality reduction within each module may be possible underlying mechanisms of modular organization of 3D object representations in the cortex. In order to construct a more

biologically feasible model, the proposed scheme should incorporate more elaborate functions in the future. Such functions include an automatic determination of the number of modules and the dimension of encoded information [3], a representation of encoded information using population codes [29], and a hierarchical construction of modules that encode local features in the lower level and object categories in the higher level.

To conclude, we have presented an unsupervised learning scheme for clustering 2D projected views of 3D objects without using any explicit identification of the object classes. The scheme consists of a mixture of nonlinear autoencoders which extract view direction information from the object images. The results of simulations using 3D wire-frame objects demonstrated that the scheme is more effective in clustering 2D object views compared with the traditional K-means algorithm. The proposed modular scheme may thus provide a useful architecture in further investigations on complex pattern clustering and in studies on how object representations are acquired in the primate visual cortex.

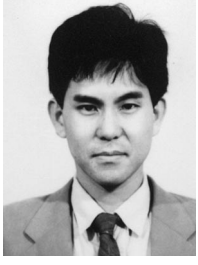
Acknowledgements

We would like to thank M. Kawato and T. Poggio for their helpful and insightful discussions. We are also grateful to the anonymous reviewers for useful comments on the paper.

References

[1] H. Ando, 3D Object recognition using bidirectional modular networks, in: S.Z. Li (Ed.), Recent Developments in Computer Vision (ACCV'95), Springer, New York, 1996, pp. 467–475.
[2] H.H. Bülthoff, S. Edelman, Psychophysical support for a two-dimensional view interpolation theory of object recognition, Proceedings of the National Academy of Science USA, Vol. 89, 1992, pp. 60–64.
[3] D. DeMers, G. Cottrell, Non-linear dimensionality reduction, in: S.J. Hanson, J.D. Cowan, C.L. Giles (Eds.), Advances in Neural Information Processing Systems, Vol 5, Morgan Kaufmann, San Mateo, CA, 1993, pp. 580–587.
[4] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[5] S. Edelman, H.H. Bülthoff, Orientation dependence in the recognition of familiar and novel views of 3D objects, Vision Research 32 (1992) 2385–2400.
[6] I. Fujita, K. Tanaka, M. Ito, K. Cheng, Columns for visual features of objects in monkey inferotemporal cortex, Nature 360 (1992) 343–346.
[7] T. Fujita, S. Suzuki, H. Ando. 3D object recognition by coupling mixtures of autoencoders and dynamic matching, Proc. of the International Conference on Neural Info. Proc. Hong Kong, 1996, pp. 377–382.
[8] G.E. Hinton, M. Revow, P. Dayan, Recognizing handwritten digits using mixtures of linear models, in: G. Tesauro, D. Touretzky, T. Leen (Eds.), Advances in Neural Information Processing Systems, Vol.7, The MIT Press, Cambridge, MA, 1995, pp. 1015–1022.
[9] R.A. Jacobs, M.I. Jordan, Learning piecewise control strategies in a modular neural network architecture, IEEE Trans. Systems, Man, and Cybernet. 23 (2) (1993) 337–345.
[10] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, Neural Computation 3 (1) (1991) 79–87.

[11] M.I. Jordan, R.A. Jacobs, Hierarchical mixtures of experts and the EM algorithm, Neural Computation 6 (2) (1994) 181–214.
[12] N.K. Logothetis, J. Pauls, Psychophysical and physiological evidence for viewer-centered object representations in the primate, Cerebral Cortex 3 (1995) 270–288.
[13] N.K. Logothetis, J. Pauls, T. Poggio, Spatial reference frames for object recognition tuning for rotations in depth, A.I. Memo No. 1533, C.B.C.L. Paper No. 120, MIT Press, Cambridge, MA, 1995.
[14] D. Marr, H.K. Nishihara, Representation and recognition of the spatial organization of three-dimensional shapes, Proc. Roy. Soc. London B 200 (1978) 269–294.
[15] M.L. Minsky, S.A. Papert, Perceptrons: an introduction to computational geometry (expanded edition), The MIT Press, Cambridge, MA, 1988.
[16] H. Murase, S.K. Nayar, Visual learning and recognition of 3D objects from appearance, Int. J. Comput. Vision 14 (1) (1995) 5–24.
[17] E. Oja, Data compression, feature extraction, and autoassociation in feedforward neural networks, in: T. Kohonen, K. Mäkisara, O. Simula, J. Kangas (Eds.), Artificial Neural Networks, Elsevier, North-Holland, Amsterdam, 1991, pp. 737–745.
[18] T. Poggio, S. Edelman, A network that learns to recognize three-dimensional objects, Nature 343 (1990) 263–266.
[19] H. Schwenk, M. Milgram, Transformation invariant autoassociation with application to handwritten character recognition, in: G. Tesauro, D. Touretzky, T. Leen (Eds.), Advances in Neural Information Processing Systems, Vol. 7, The MIT Press, Cambridge, MA, 1995, pp. 991–998.
[20] S. Suzuki, H. Ando, Recognition and classification of 3D objects using a modular learning network, Technical Report of IEICE, NC93-62, 1993, pp. 59–66.
[21] S. Suzuki, H. Ando, Unsupervised classification of 3D objects from 2D views, in: G. Tesauro, D. Touretzky, T. Leen (Eds.), Advances in Neural Information Processing Systems, Vol. 7, The MIT Press, Cambridge, MA, 1995, pp. 949–956.
[22] K. Tanaka, Neuronal mechanisms of object recognition, Science 262 (1993) 685–688.
[23] K. Tanaka, H. Saito, Y. Fukada, M. Morita, Coding visual images of objects in the inferotemporal cortex of the macaque monkey, J. Neurophysiol. 6 (1) (1991) 170–189.
[24] S. Ullman, R. Basri, Recognition by linear combinations of models, IEEE Trans. Pattern Anal. Mach. Intell. 13 (10) (1991) 992–1006.
[25] G. Underwood (Ed.), Implicit Cognition, Oxford University Press, Oxford, UK, 1996.
[26] Y. Uno, N. Fukumura, R. Suzuki, M. Kawato, A computational model for recognizing objects and planning hand shapes in grasping movements, Neural Networks 8 (6) (1995) 839–851.
[27] S. Usui, S. Nakauchi, M. Nakano, Reconstruction of Munsell color space by a five-layer neural network, J. Opt. Soc. Am. A 9 (4) (1992) 516–520.
[28] G. Wang, K. Tanaka, M. Tanifuji, Optical imaging of functional organization in the monkey inferotemporal cortex, Science 272 (1996) 1665–1668.
[29] R. Zemel, G.E. Hinton, Learning population codes by minimizing description length, Neural Computat. 7 (1995) 549–564.



Satoshi Suzuki received the B.S. degree from the College of Arts and Sciences at the University of Tokyo, Japan in 1990. He joined NTT (Nippon Telegraph and Telephone Corporation) in 1990, and from 1992 to 1997, he was a researcher of ATR Human Information Processing Research Laboratories. He is currently working at NTT Communication Science Laboratories, Kyoto, Japan. His research interest are in object recognition and computational approaches to brain functions.

Hiroshi Ando received the B.S. degree in Physics, the B.A. and M.A. degrees in Experimental Psychology from Kyoto University, Japan in 1983, 1985 and 1987, respectively, and the Ph.D. degree in Computational Neuroscience from Massachusetts Institute of Technology, USA in 1993. In December 1992 he joined advanced telecommunication research institute (ATR) Human Information Processing Research Laboratories, Kyoto, Japan, where he is currently a Senior Researcher. His research interests include neural computation, visual motion and shape processing, object recognition, learning models, as well as visual psychophysics.