ELSEVIER

Image and Vision Computing 19 (2001) 585–592

image AND vision COMPUTING

www.elsevier.com/locate/imavis

Neural networks for the recognition and pose estimation of 3D objects from a single 2D perspective view

C. Yuan*, H. Niemann

University of Erlangen-Nuremberg, Martensstr. 3, 91058 Erlangen, Germany

Received 5 April 2000; accepted 18 December 2000

Abstract

In this paper we present a neural network (NN) based system for recognition and pose estimation of 3D objects from a single 2D perspective view. We develop an appearance based neural approach for this task. First the object is represented in a feature vector derived by a principal component network. Then a NN classifier trained with *R*esilient back*prop*agation (Rprop) algorithm is applied to identify it. Next pose parameters are obtained by four NN estimators trained on the same feature vector. Performance on recognition and pose estimation for real images under occlusions are shown. Comparative studies with two other approaches are carried out.

Keywords: 3D object recognition; Pose estimation; Neural nets; Appearance based

1. Introduction

Recognition and pose estimation of 3D objects from arbitrary viewpoint is a fundamental issue in computer vision and has applications in many areas such as automatic target recognition (ATR), navigation, manufacturing and inspection. Despite the considerable endeavor and success achieved so far, a practical system which has a good compromise involving performance, implementation and computational complexity, still remains a desired goal. Beyond being accurate, such system must be robust, low in computational requirement during run-time, and simple to implement. In this work, we aim to develop a system that satisfies all these criteria. Instead of using additional information such as depth or color which demands additional hardware and software costs but contributes few to the recognition improvement [19], the input data is the minimum possible information that one can extract from a 3D pattern, namely a single 2D gray-level image. With this less demanding input, both the complexity of the system as well as its execution time can be reduced.

In this paper, a fast and robust system is presented that recognizes a 3D object from a single 2D image of the object viewed from an arbitrary angle in the 3D space. In addition, two translation and two rotation parameters are computed. The two translation parameters are the x- and y-coordinate of the object (in pixel) in the image plane. And the two rotation parameters are the rotation angles of the object with respect to the camera: the aspect (viewing) angle α and the elevation angle ϕ accordingly. In concrete, α is the rotation of the object within the image plane (internal rotation), and ϕ is the rotation out of the image plane (external rotation). Schematic views of these angles are shown in Fig. 1.

There has been extensive research on object detection and classification [7,27]. The approaches vary mainly in the representation of 3D objects and in the search techniques for matching data to models [21,22]. The shape-based representations usually store an explicit 3D model for each known object, where the models are obtained either manually or by a computer-aided design (CAD) system [12]. The recognition is performed by matching object data structure derived from the observed 2D images to the 3D model data structure. Thus, 3D to 2D or 2D to 3D transformations must be performed before a matching can take place [2].

Another frequently used approach is feature-based representation. Some early works use regular moment (RM) or Fourier descriptors (FD) to characterize the boundary of segmented object [6,28]. Unfortunately RM and FD are shown to be rather sensitive to noise and to perturbations in the object boundary [23]. Structural features based on corners and line segments are used in Refs. [3,8]. Other

* Corresponding author.

E-mail addresses: yuan@informatik.uni-erlangen.de (C. Yuan), niemann@informatik.uni-erlangen.de (H. Niemann).

0262-8856/01/$ - see front matter
PII: S0262-8856(01)00037-3

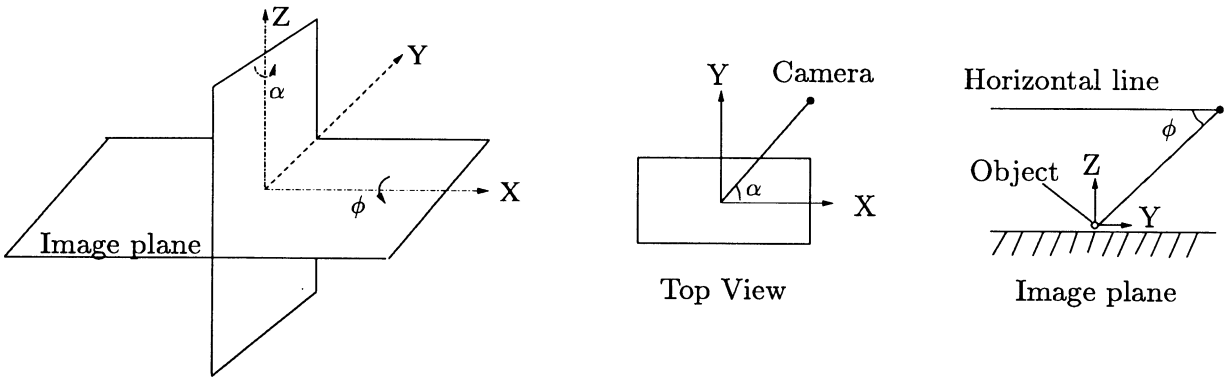

Fig. 1. Schematic of the two rotation parameters.

approaches extract complex features based on geometric/projective/perspective invariants [13,15,24]. Yet, extraction of such features is not only time consuming but often unreliable.

In contrast to the above representation paradigm, a number of appearance-based systems have emerged. Begin with the eigenface approach proposed by Turk and Pentland [26] for face recognition, and later extended by Nayar [18] and Murase [17] to general object recognition, appearance-based approach uses a set of images obtained from different views as its implicit description. [25] applies a probabilistic approach for appearance modeling where receptive field histogram was used to represent objects. Yet, it is difficult to use this histogram-based strategy for object pose estimation. A recent work [20] in this category applies Gabor wavelet filters for statistical object localization. A statistical object model is built whose parameters are estimated using maximum likelihood estimation. A multi-matching strategy is necessary for object pose estimation, which consists of a global pose search and succeeding local search. Though it achieves considerable accuracy in parameter estimation, no direct report on recognition result can be found. A comparison of this work to our approach on object pose estimation can be found in Section 5.

Generally, vision systems must be able to reason about both identity and pose of objects. In this paper we utilize several neural nets (NNs) for appearance-based recognition and pose estimation. In this scheme, expensive storage of a multiview database is not needed, since during training the NNs are expected to extract all the relevant information and to form a compact representation of the objects. Also, due to generalization capability of the NNs, good results can be obtained with a relative small number of views. The adaptive feature of NNs causes the system to be robust and to perform well when distortions are present. Finally since a trained NN can execute quickly, the need for slow matching strategies is eliminated, and the overall speed of the system becomes quite reasonable.

The organization of the rest of this paper is as follows. Section 2 deals with feature extraction using the principal component network (PCN) approach. Section 3 describes the NN classifier. Pose estimation by NN is discussed in Section 4. Experimental results based on both 2D and 3D objects are presented in Section 5. Finally, Section 6 summarizes the whole paper.

2. Feature extraction using principal component network

A judicious selection of features reducing the dimensionality of the input vector while preserving most of the intrinsic information content is an approach that will improve the generalization of a classifier. Subspace methods such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are proved to be very effective ways for extracting low-dimensional manifolds from the original data space [4,10]. While ICA seeks statistically independent and non-Gaussian components and is suitable for blind source separation, PCA is an eigenvector-based technique which is commonly used for dimensionality reduction. Through PCA, a n-dimensional pattern vector $\boldsymbol{x}$ can be mapped to a feature vector c in a m-dimensional space, where $m < n$. In other words, PCA finds the invertible transform $\boldsymbol{T}$ such that truncation of $\boldsymbol{x}$ to c ($c = \boldsymbol{T}\cdot\boldsymbol{x}$) is optimum in the mean square error sense. The linearity and invertibility of transform $\boldsymbol{T}$ makes the components of such PCA orthogonal, ordered and linear.

Standard linear PCA has found wide applications in pattern recognition and image processing. Yet the linearity can cause problems in some cases [5]. The alternative non-linear PCA builds a non-linear (curved) lower-dimensional surface called principal surface that "passes through the middle of the data" and thus yield a relatively accurate representation of the data. Suppose functions $\boldsymbol{g}$ and $\boldsymbol{h}$ are two non-linear mapping each from $\Re^n$ to $\Re^m$ and from $\Re^m$ to $\Re^n$, respectively, the target of the non-linear PCA is the minimization of the non-linear reconstruction mean squared error

$$J = E\|\boldsymbol{x} - \boldsymbol{h}(\boldsymbol{g}(\boldsymbol{x}))\|^2 \tag{1}$$

by an optimal choice of $\boldsymbol{g}$ and $\boldsymbol{h}$.

One of the simplest methods for computing non-linear principal manifolds is the PCNs which has a non-linear hidden layer incorporating the sigmoid activation function [29]. As shown in Fig. 2(a), the output of the network is set to be equal to the input pattern vector $\boldsymbol{x}$, because it is to be trained to reproduce the input $\boldsymbol{x}$. And the hidden unit activations correspond to a feature vector $\boldsymbol{c}$ in $\Re^m$. The mapping from the input layer to hidden layer and from the hidden layer to the output layer can be regarded accordingly as the non-linear function $\boldsymbol{g}$ and $\boldsymbol{h}$. The main advantage of PCN is that it can be done automatically and no prior knowledge regarding the joint distribution of the components is necessary. It has been shown in Refs. [1,9] that, a PCN with linear functions can be made to converge to the principal components. Intuitively, one expects a greater discriminative power to result from the non-linear neurons.

We are interested not in the exact principal components, but in components of variables, or features, which are non-linearly related to the input variables. Thus in our implementation, we first train a 4-1-4 PCN and then apply the result function $\boldsymbol{g}$ hierarchically to the input image $\boldsymbol{f}_0$ (256×256 pixels). During the training process the network receives the 2×2 subimages which are cut sequentially out of $\boldsymbol{f}_0$ without overlap. The result function $\boldsymbol{g}$ is similar to that of a lowpass filter:

$$\boldsymbol{g}(\boldsymbol{x}) = \mathrm{Act}\left(\sum_{k=1}^{4} (w_k x_k + \theta)\right) \tag{2}$$

where w_k, θ are the weight vector and thresholds of the hidden neuron and Act(s) the sigmoid activation functions $1/(1 + \exp(-s))$. Apply the function $\boldsymbol{g}$ one time, we get an image $\boldsymbol{f}_1$ which is one-fourth of the original image $\boldsymbol{f}_0$ (see Fig. 2(b)). By repeating such operation

$$\boldsymbol{f}_n(i,j) = \boldsymbol{g} * \boldsymbol{f}_{n-1}(i,j) \tag{3}$$

four times, we get a feature vector $\boldsymbol{c} = \boldsymbol{f}_4$, which is of 256 ($16 \times 16$) dimensions and will be used in the subsequent recognition and localization process.

3. Neural classifier

To recognize an object, we need to classify it as belonging to one class Ω_k out of k object classes Ω_λ, $\lambda = 1, \ldots, k$ based on the feature vector $\boldsymbol{c}$. A three layer feed-forward network whose input neuron numbers equal to the dimension of $\boldsymbol{c}$ and whose output neuron numbers equal to k can be applied to form a model for classification. The output of the net o_λ, $\lambda = 1, \ldots, k$ can be interpreted as measuring the posterior possibility function $p(\Omega_\lambda|\boldsymbol{c})$ for each class. According to Bayes rule, vector $\boldsymbol{c}$ should be classified as coming from Ω_κ with $\kappa = \mathrm{argmax}\{o_\lambda\}$, $\lambda = 1, \ldots, k$. In order to reject objects that do not belong to any of the classes, additional criteria have been incorporated into the system. Let $\kappa_2 = \mathrm{argmax}\{o_\lambda\}$, $\lambda = 1, \ldots, k$, $\lambda \neq \kappa$, the image with feature vector $\boldsymbol{c}$ can be classified as containing an object of class Ω_κ only if:

$$o_\kappa \geq \Theta_0 \tag{4}$$

and

$$\frac{o_\kappa}{o_{\kappa_2}} \geq \Theta_1 \tag{5}$$

Θ_0 and Θ_1 are fixed before the experiment. Since the neuron output can vary between 0.0 and 1.0, we set $\Theta_0 = 0.6$ for all the NNs we use. As to Θ_1, it is set based on experience to $\Theta_1 = 1.3$.

The number of hidden nodes of the neural classifier is set by trail and error, varying from 20 to 80. By alternating the number of hidden nodes, a best net can be found, which can make the comparison of different approaches more reasonable. Training is done with the Rprop algorithm, which we describe briefly in the following. The basic principle of Rprop is to eliminate the harmful influence of the size of the partial derivative on the weight step. As a consequence, only the sign of the derivate is considered to indicate the direction of the weight update. Concretely the weight change $\Delta w_{ij}^{(t)}$ is determined as follows:

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & \text{if } \dfrac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)} & \text{if } \dfrac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ 0 & \text{else} \end{cases} \tag{6}$$

Then the new update-value $\Delta_{ij}^{(t+1)}$ is determined, basing on a sign-dependent adaptation process

$$\Delta_{ij}^{(t+1)} = \begin{cases} \eta^+ * \Delta_{ij}^{(t)} & \text{if } \dfrac{\partial E}{\partial w_{ij}}^{(t-1)} * \dfrac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \eta^- * \Delta_{ij}^{(t)} & \text{if } \dfrac{\partial E}{\partial w_{ij}}^{(t-1)} * \dfrac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t)} & \text{else} \end{cases} \tag{7}$$

where $0 < \eta^- < 1 < \eta^+$.

In our experiment we set $\eta^+ = 1.2$ and $\eta^- = 0.5$. Because Rprop modifies the size of the weight-step directly by introducing the concept of resilient update-value Δ_{ij}, it converges very fast. Another advantage is that no special choice of parameters is needed at all to obtain optimal or at least nearly optimal convergence time.

4. Neural pose estimator

After the object is recognized, the module that provides estimation of its pose parameters, namely the translation and rotation parameters is activated. The module consists of two stages with the first one estimating the x- and y-coordinates of the recognized object in the image plane and the second one providing the estimation of the internal and external

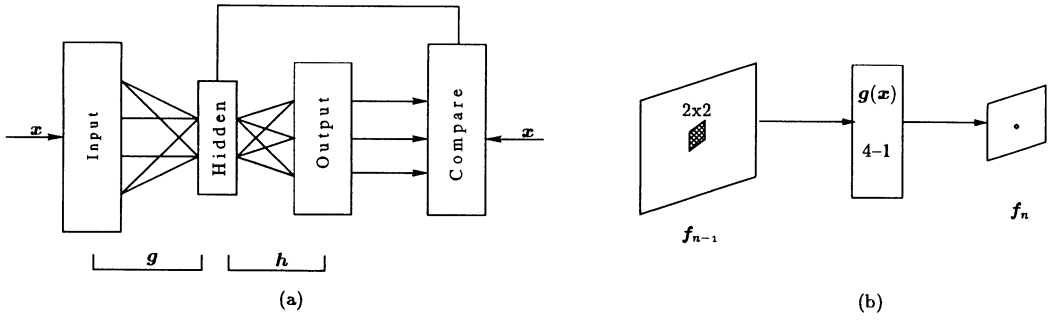

Fig. 2. (a) Topology of PCN. (b) Hierarchic feature generation.

rotation parameters. At each stage, the task is partitioned into simpler sub-problems, and multiple NNs are utilized to solve it. Each of these NNs receives the same feature vector as the NN classifier.

For computing of the translation parameters in x and y direction, two three-layer feed-forward NNs are implemented for each of the object class. Different from the NN for classification which is a full connected one there is only horizontal or vertical connection between the input layer and the hidden layer as illustrated in Fig. 3. Also they are NN estimators rather than classifiers. As shown in Fig. 3, each estimator has 16 neurons in the hidden layer and one output neuron. The output of each estimator is a real value between 0 and 1. By multiplying the output with the width or height of the image which is in our case both 256 pixel, we obtain the object position in x or y direction, respectively.

For each of the object classes, two NN estimators with identical architecture are configured to estimate the two rotation angles α and ϕ. In contrast to the translation parameter estimators which approximate the geometric center of the objects, we try to distinguish the different views as different rotation categories of an object. For this reason, a topology different from that of the translation parameter estimators is chosen. As Fig. 4 illustrates, each estimator receives the same feature vector and has only one output neuron. By initialization, the number of neurons in the hidden layer is equal to the discrete number of possible different category of viewing angles. For example, both α and ϕ span from $-45°$ to $45°$ in our 3D experiment. And the interval of viewing angles is $\Delta = 3°$ (both in the internal and external rotation). In this situation, each rotation estimator should have $N = 30$ hidden neurons arranged in ascending order at the initialization stage (Fig. 4). Moreover views with rotation angles belonging to $[n\Delta,\ (n+1)\Delta)$, $0 \le n < N$, are in the same rotation category. Though there is only N different rotation categories, within each rotation category, the views can have a difference within Δ.

Because of their particular topology, training of the rotation parameter estimators is done with the dynamic learning vector quantization (DLVQ) algorithm [30]. With DLVQ a natural grouping in a set of data can be found very quickly. Since vectors belonging to the same class (in our case with the same rotation category) should have smaller difference than those belonging to different classes (different rotation categories), we estimate the rotation angles by trying to find a natural grouping in the set of image data. The mathematic formulation of this training algorithm is as follows: Suppose that the vector $\boldsymbol{c}_i$ belonging to the same class (in our case within the same category of perspective views) are distributed normally with a mean vector $\boldsymbol{\mu}_i$. A feature vector $\boldsymbol{c}$ is assigned to the class Ω_i with the smallest Euclidean distance $\|\boldsymbol{\mu}_i - \boldsymbol{c}\|^2$. During the learning phase, the algorithm computes a new mean vector $\boldsymbol{\mu}_i$ for each class once every cycle and generates the hidden layer dynamically. Training is finished when correct $\boldsymbol{\mu}_i$ is found stable for every class. After training the network outputs a natural number n $(0 \le n < N)$. The rotation angle is computed as to be equal to $n * \Delta$.

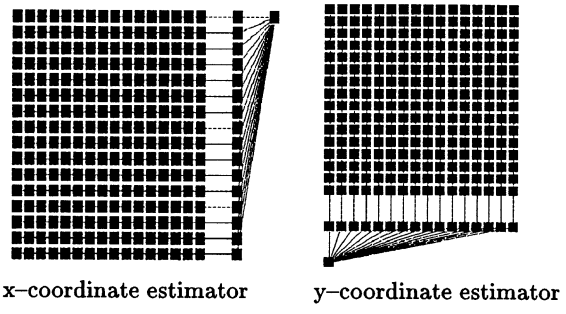

Fig. 3. Translation parameter estimators.

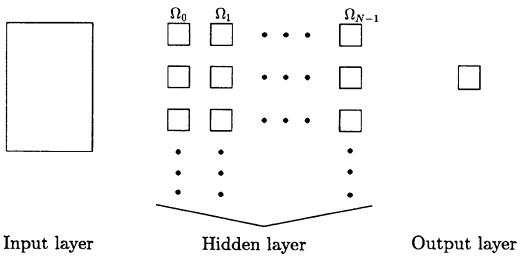

Fig. 4. Topology of the rotation parameter estimators.

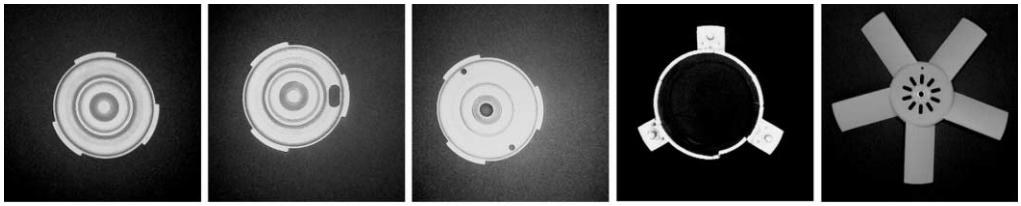

Fig. 5. Objects used in the first experiment.

5. Experimental study

Though there exist some image databases such as the famous Colombia image database, most such databases are designed for recognition, but not for pose estimation. And the unavailability of the original objects make it unfavorable in such situation where real-time application must be developed. Our goal is to develop an object recognition system to be used later by a service robot which can act in an office or a home environment. For the robot to be able to locate and grasp objects, we develop in the first step a neural system for object recognition and pose estimation. The system is built on a SGI O2 (R10000) workstation, which is connected to a CCD-camera (focal length = 16 mm) mounted on a robot (Scorbot ER-VII). By now, two experiments have been carried out on the system. The first experiment is based on five 2D objects shown in Fig. 5. By these objects there exists only the internal rotation α. In the second experiment we use three 3D objects (a), (b) and (c) shown in Fig. 6. All the images have a dimension of 256×256 pixels with the objects appearing in uniform scale. It takes about 0.5 s for feature extraction, recognition and pose estimation altogether. The achieved precision of the computed object center using neural estimators is in average 1.2 pixel. And the average errors of the rotation parameter are 1.8° for the internal rotation α and 2.6° for the external rotation ϕ.

In the 2D experiment, 400 images are taken with 10 different positions and 40 different rotation categories for each object class. The image data is divided into training and test samples randomly, so that some views of the test images have never appeared in the training process at all. Training set consists of 160 images/class, Test set consists of 240 image/class. The achieved recognition results on the test set using PCN as well as different wavelet transform are shown in Table 1. As we know, a 2D discrete wavelet transform is computed by applying a separable filterbank to the image repeatedly [16]. Through lowpass filter $\boldsymbol{H}$ followed by subsampling along image rows and columns, a low resolution image can be retrieved, which can be denoted mathematically as follows:

$$\boldsymbol{f}_n(x,y) = [\boldsymbol{H}_x * [\boldsymbol{H}_y * \boldsymbol{f}_{n-1}]_{|2,1}]_{|1,2}(x,y) \tag{8}$$

By applying the lowpass filter $\boldsymbol{H}$ four times, we can get the wavelet feature vector $\boldsymbol{f}_4$, which has the same dimension as the PCN networks. As for the description of the coefficients of the different kinds of wavelet transforms, please refer to Ref. [11]. Though the similarity of the three kinds of lid makes the recognition task a very difficult one, PCN based feature extraction achieves an average recognition rate of 99.6%. The best wavelet-based features resulted from Daubechies 4-tap wavelet, which is more than 1% lower than that of the PCN feature. Another superiority of the PCN feature detector is its fast convergence during training, which can be obviously seen from Fig. 7.

In the 3D experiment, Each object is available in four image sequences with 6144 image each. Each sequence is taken under a different illumination. Under each illumination, objects are taken having translational variation within the whole image plane and rotational variation (both α and ϕ) of 90°. This means that for each object, there are 24,576 images taken in different positions, at different perspective viewpoints, and with different lightning conditions. Here we divide the four sequences into two disjoint parts: training data and test data. The test data is further subdivided into a *validation set* and a *test set*. The training data is used for training the network. The *validation set* is used to estimate network performance during training as we use the early stopping criteria to avoid overfitting. Yet, the validation set is never used for weight adjustment. Therefore it is used with the *test set* together to evaluate the network performance after training is finished. Training set consists of 2048 images/class, which are

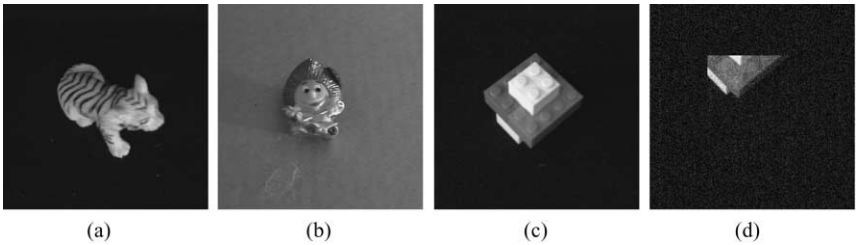

Fig. 6. Objects used in the second experiment.

Table 1
Recognition results-2D experiment

	Recognition rate (%)					
	Lid0	Lid1	Lid2	Case	Fan	Aver.
PCN	100	98.3	100	100	99.6	99.6
Haar	100	91.3	99.6	100	100	98.2
Daub4	99.6	93.75	98.3	100	99.6	98.25
Daub6	100	91.7	98.3	100	99.2	98.0
Daub7	100	91.3	99.6	99.6	98.75	97.8
Daub9	99.6	82.5	100	100	97.5	95.9
QMF8	100	87.5	100	100	99.2	97.3
QMF12	100	92.1	98.75	100	99.6	98.1
Villa6	100	83.3	99.6	100	100	96.6
Zhu4	93.75	79.2	98.3	100	97.9	93.75
QZ10	100	75.9	100	99.6	99.6	95.0

Table 2
Results of 3D object recognition

Approach	Object	Recognition rate (%)				
		Without occlusion when hidden neurons =				With occlusion with hidden neurons =
		30	42	56	64	56
PCN	(a)	97.6	98.4	98.4	98.3	67.1
	(b)	92.3	93.2	93.8	93.4	87.5
	(c)	97.2	97.8	99.2	98.2	88.4
SOM	(a)	96.5	96.3	97.1	96.5	57.4
	(b)	90.9	91.8	92.0	92.0	78.1
	(c)	93.1	93.8	93.9	93.2	76.9

randomly selected only from the first and second sequence. Another 2048 images/class randomly selected from the third and fourth sequence form the validation set. And the rest are used as *test set*. As such, the ratio of the amount of training data, *validation set*and *test set* is 1:1:10.

To evaluate the system performance on the 3D objects, first we have compared the PCN with Kohonen's self-organization feature map (SOM). As there is not enough room here, we refer to Ref. [14] for a detailed description of the SOM-based feature extraction method. The total processing time for feature extraction, recognition and pose estimation using SOM is 1.6 s on the same SGI workstation. Since the dimension of the feature vector resulting from this method is also 256, which is the same with our PCN features, we can compare the influence of SOM-feature and our PCN-feature on the NN classifier as well as NN estimators with same network architecture. In both cases, we vary the number of neurons in the hidden layer from 20 to 80 so as to find the best classifier. The best results are achieved with the number of hidden neurons between 30 and 64. Using the best NN configurations, we further test the robustness of both approach (PCN and SOM) for new views and in case of noise and occlusion. For this reason, Gaussian noise with zero mean and a standard deviation up to 75 is added to the images after objects are occluded up to 60% as shown in Fig. 6(d). Since occlusion is done by translating or rotating the objects out of their image plane, the resulted images have not only noise and occlusion, but also views different from the original data. Through such operations, a new data set with 4096 images/class is generated.

Furthermore we compared our approach with the statistical

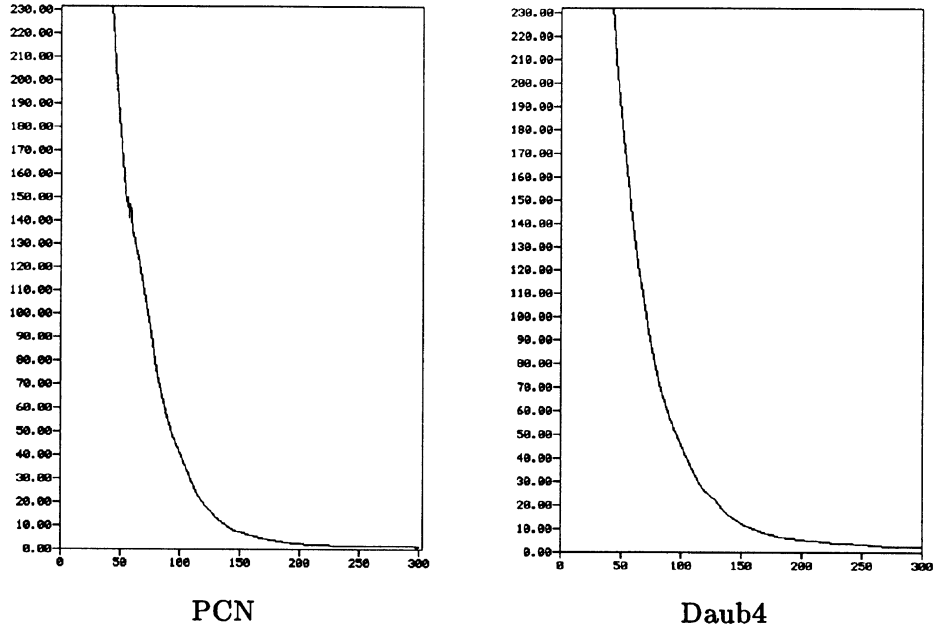

Fig. 7. Convergence of the neural classifier.

Table 3
Results of 3D object pose estimation

Object	Approach	Fail (%)	Error					
			Transl. (Pix)		Int.rot. (°)		Ext.rot. (°)	
			Mean	Max	Mean	Max	Mean	Max
(a)	PCN	0	1.1	2.8	1.1	3.4	2.3	6.8
	SOM	17	1.1	2.4	3.7	6.8	4.3	6.8
	STA.	0	1.1	2.5	1.3	3.4	1.3	6.8
(b)	PCN	4	0.8	1.5	1.7	6.8	3.2	6.8
	SOM	34	0.9	2.9	4.5	6.8	5.4	6.8
	STA	15	1.8	3.5	1.3	5.9	3.7	8.9
(c)	PCN	2	1.4	2.6	2.7	6.8	2.2	6.8
	SOM	23	1.7	4.4	4.3	6.8	5.7	6.8
	STA.	18	2.5	5.3	1.4	3.4	3.5	8.6

one described in Ref. [20] for pose estimation. On the same machine it takes about 6 s for feature extraction and localization of one of the objects by this approach, which is much time consuming than our neural approach.

We summarize the performance of our system as well as the other two approaches (named as SOM and STA, respectively) in tables as follows: Table 2 lists the recognition rate of the three objects on the test data using the two different feature extraction methods. The precision of pose estimation using all three different approaches is shown in Table 3. An estimation result is regarded as failure if the resulted translation error is more than 10 pixels or the rotation error is more than 9°.

As shown in Table 2, both PCN and SOM approach achieve their best recognition results when there are 56 neurons in the hidden layer. One can see no big difference on the recognition rate when the number of hidden neurons are between 30 and 56. This proves in somewhat the effectiveness of the Rprop training algorithm. At the same time, slight degradation of the recognition rate can also be seen when there are more than 56 neurons in the hidden layer. This complies with the generally believed weak generation resulting from an overfitting provided by too many hidden neurons in a net. In average the recognition rate is 97.1% by PCN and 94.3% by SOM. Under occlusion and noise, PCN achieved an average recognition rate of 77%, while SOM got near to 70%. As to pose estimation accuracy, the average estimated precision in translation is 1.1 pixel by PCN, 1.2 pixel by SOM and 1.8 pixel by STA. Based on PCN, SOM, and STA approach accordingly, there is in average 1.8°, 3.2° and 1.3° estimation error in the internal rotation parameters and 2.6°, 5.2° and 2.8° in the external rotation parameters. While SOM achieves comparable recognition rate with PCN approach, it is shown through this study that it has a poorer localization ability. Therefore, we argue that feature extraction through PCN appears to be a promising one. Also in run-time effectiveness, PCN outperforms the SOM as well as STA. Though we cannot see big performance difference on pose estimation exactness between the statistical and our neural approach from Table 3, the overall fail estimation of our approach is much lower. Also quite different from the STA method, whose localization performance changes when different object is presented, PCN shows a much lower performance variance on all three objects. Hence the neural approach proves itself to be relatively more robust and stable than the statistical one.

6. Conclusion

We developed an automatic system for the recognition and pose estimation of 3D objects viewed from arbitrary location and perspective angles, where NN theory is widely applied to model the object appearance. The effectiveness and accuracy of the proposed system is demonstrated in two experiments involving a data set of eight objects. Comparative studies with two other approaches, a neural and a statistical one, are carried out. It is shown that our system is more accurate, robust and faster than the other two approaches. It is concluded that with the proposed appearance based neural approach, recognition and pose estimation of 3D objects can be performed with low computational requirement and high accuracy. Currently we are applying this neural approach to a set of fourteen 3D objects (cup, plate, box, bottle, fork, spoon, knife, two kinds of cola dose, two kinds of stapler and three kinds of hole puncher) and have achieved a similar recognition rate. In the future, we will enhance the proposed method to cope with scenes with more than one object appearing before heterogeneous background.

References

[1] H. Boulard, Y. Kamp, Auto-association by multilayer perceptrons and singular value decomposition, Biological Cybernetics 59 (1988) 291–294.

[2] J.L. Chen, G.C. Stockman, Matching curved 3d object models to 2d images, CAD-Based Vision Workshop, 1994, pp. 210–218.

[3] S.W. Chen, A.K. Jain, Strategies of multi-view and multi-matching for 3d object recognition, CVGIP 57 (1) (1993) 121–130.
[4] P. Comon, Independent component analysis: a new concept?, Signal Processing 36 (1994) 287–314.
[5] K.I. Diamantaras, S.Y. Kung, Principal Component Neural Networks: Theory and Applications, Wiley, New York, 1996.
[6] S.A. Dudani, K.J. Breeding, R.B. McGhee, Aircraft identification by moment invariants, IEEE Transactions on Computer 26 (1) (1977) 39–46.
[7] S. Edelman, On learning to recognize 3-d objects from examples, IEEE Transactions on PAMI 15 (8) (1993) 833–837.
[8] J. Hornegger, H. Niemann, Statistical learning, localization, and identification of objects, ICCV95, 1995, pp. 914–919.
[9] A. Hornik, M. Strinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989) 359–366.
[10] I.T. Jolliffe, Principal Component Analysis, Springer, New York, 1986.
[11] P. Kral, Wavelet Transforms, Chair for Pattern Recognition, University of Erlangen-Nuernberg, 1995.
[12] B. Krebs, F.M. Wahl, Automatic generation of bayesian net for 3d object recognition, ICPR98, 1998, pp. 126–128.
[13] Y. Kuno, O. Takae, T Takahashi, Y. Shirai, Object recognition using multiple view invariance based on complex features, WACV96, 1996, pp. 129–134.
[14] H.M. Lakany, E.-G. Schukat-Talamazzini, H. Niemann, M. Ghoneimy, Object recognition from 2d images using kohonen's self-organized feature maps, Pattern Recognition and Image Analysis 7 (3) (1997) 301–308.
[15] D.G. Lowe, Object recognition from local scale-invariant features, ICCV99, 1999, pp. 1150–1157.
[16] S. Mallat, A. theory, for multiresolution signal decomposition: The wavelet representation, IEEE Transactions on PAMI 11 (1989) 674–693.
[17] H. Murase, S.K. Nayar, Visual learning and recognition of 3-d objects from appearance, IJCV 14 (1) (1995) 5–24.
[18] S.K. Nayar, S.A. Nene, H. Murase, Subspace methods for robot vision, IEEE Transactions on Robotics and Automation 12 (5) (1996) 750–758.
[19] J. Poesl, B. Heigl, H. Niemann, Color and depth in appearance based statistical object localization, Proceedings of the 10th IMDSP Workshop, 1998, pp. 71–74.
[20] J. Poesl, H. Niemann, Wavelet features for statistical object localization without segmentation, ICIP97, 1997, pp. 170–173.
[21] T. Poggio, S. Edelman, M. Fahle, Learning of visual modules from examples: a framework for understanding adaptive visual performance, CVGIP: Image Understanding 56 (1) (1992) 22–30.
[22] J. Ponce, A. Zissermann, M. Hebert, Object representation in computer vision II (ECCV96), Springer, Berlin, 1996.
[23] A.P. Reeves, R.J. Prokop, S.E. Andrews, F.P. Kuhl, Three-dimensional shape analysis using moments and fourier descriptors, IEEE Transactions on PAMI 10 (6) (1988) 937–943.
[24] T. Satonaka, T. Baba, T. Otsuki, T. Chikamura, T. Meng, Object recognition with luminance, rotation and location invariance, ICIP97, 1997, pp. 336–339.
[25] B. Schiele, J.L. Crowley, Probabilistic object recognition using multidimensional receptive field histograms, ICPR96, 1996, pp. 50–54.
[26] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 71–86.
[27] S. Ullman, High-level Vision: Object Recognition and Visual Cognition, MIT Press, Cambridge, MA, 1996.
[28] T.P. Wallace, P.A. Wintz, An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors, CGIP 13 (1) (1980) 99–126.
[29] A. Weingessel, H. Bischof, K. Hornik, Hierarchies of autoassociators, ICPR96, 1996, pp. 200–204.
[30] A. Zell, Simulation neuronaler Netze, Addison-Wesley, Bonn, 1994.