MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

3-D CURVED OBJECT RECOGNITION BASED ON PARAMETER NETS WEIGHTED WITH FEATURE SALIENCY

Hongbin Zha, Tadashi Nagata

Department of Computer Science and Communication Engineering
Kyushu University
6–10–1 Hakozaki, Higashi–ku
Fukuoka, 812, Japan

Abstract — *The paper describes a new method of model generation that can be used for 3-D curved object recognition by means of evidence collection through hierarchical parameter nets. The model database is composed of a set of high-dimensional interpretation tables with feature parameters as their indices. In the model generation process, the tables are registered with object hypotheses associated with pose-invariant features that are extracted from sample images or CAD data. The important aspect of the method is that feature saliency representing discriminating power of the features can be evaluated automatically and simultaneously with the feature registration. An algorithm for evaluating the feature saliency is proposed on the basis of learning mechanisms of neural networks. It is shown that the algorithm can be easily employed for modeling curved objects whose surface patches can be transformed into a unary and a binary parameter spaces. Results of some simulation experiments are reported.*

1 Introduction

Many high-level vision systems require a good use of the relevant knowledge of the objects to be recognized. In a *model-based vision system*, the knowledge is usually organized in a model base which plays a crucial role for the performance of the whole system. In general, the model generation process involves following two main parts:

- Extracting characteristic features of the model objects from sample training images or CAD data.
- Registering the extracted features into model database with a good structure so that they can be taken into recognition conveniently.

The feature extraction is a easy task since many efficient algorithms for the feature extraction are available ([3]). The *feature registration*, however, is much difficult to perform as compared with the feature extraction. In the most of model-based methods proposed until now, the feature registration is carried out by indexing model features into some *interpretation tables* that are used to produce, in the recognition phase, hypotheses according to sensory information. The table structures are usually very simple and the important point for reliable recognition is to use as many features as possible for setting up correct hypotheses.

On the other hand, the increase of the feature number will make the size of database very large, and in consequence, the cost for *hypothesis verification* increases drastically. A reasonable modification to the feature registration process is to associate with each indexed feature a coefficient showing the discriminating power of the feature for different recognition purposes. The discriminating power is usually referred as to *feature saliency* and its use will facilitate hypothesis verification by neglecting a large number of less potential hypotheses.

In the paper, we propose a new method for the saliency evaluation by utilizing pattern learning mechanisms of *neural networks*. It is assumed that, in the model generation process, sample images of model objects are shown in turn to the model-building modules that extract *pose-invariant features*. The features are then registered into some interpretation tables with the feature parameters as their indices. At the same time, a three-layers neural network is constructed for each interpretation table, with its input cells corresponding to table entries and its output cells to possible model objects. During the feature registration, the network is updated by using the well-known *back-propagation* algorithm with training and teacher patterns formed from the indexed tables. After the registration, input patterns corresponding each to a table entry are formed and shown to the trained network in sequence. The output signal is then used as value of the saliency coefficient for the considered feature-and-object pair.

The model base is most suited for the object recognition based on the parameter-nets approach, in which the evidence for globally consistent interpretations is collected through some hierarchical networks ([1], [2]). The computed saliency coefficients of the table entries can be used to specify the initial state of the networks to perform the evolving *constraint satisfication* process. Generally speaking, the main advantages of such an approach are the tolerance to image noises and occlusions as well as the fastness of the on-line recognition. It is noted that the model is fit also for *interpretation tree (IT)* approach if the required search process is planed by using the feature saliency coefficients ([6], [10]).

In our current implementation, the primitive pose-invariant features are chosen as unary and binary constraints for describing surface patches on the model object surfaces. Experimental results showing behaviors of the proposed method are presented.

2 Interpretation Tables for Modeling

Interpretation tables are a kind of data structures which associate interpretation hypotheses with model features to produce an efficient feature matching procedure ([5]). Corresponding to a specified feature type, an interpretation table is a high-dimensional table with the feature parameters as its indices. For a feature F with n parameters, an interpretation table can be constructed by specifying the indices of the table entries as $(k_{F1}, k_{F2}, \cdots, k_{Fn})$, or

simply $\mathbf{k}_F$, where k_{Fi} are quantified parameter values. The recorded items in the entries are hypotheses associated with the corresponding features. In object recognition, if a feature located at $\mathbf{k}_F$ belongs to model objects M_a, M_b, $\cdots$, the record of the entry can be written as

$$\mathcal{E}(\mathbf{k}_F) =< M_a, M_b, \cdots >, \tag{1}$$

where M_a, M_b, $\cdots$ indicate the hypotheses about object occurrence when F is found in sensor data. If multiple types of features are used, the same number of tables should be constructed.

In parameter-nets approach, the initial hypotheses generated from the tables can be sent to parameter networks for getting globally consistent interpretations via. a constraint satisfication process. If the hypotheses have the same possibility for supporting each interpretation, the constraint satisfication process usually costs a long computational time before reaching the stable state. In fact, however, each feature registered in the tables has a different discriminating power for the recognition of different objects. For example, the features belonging to only one object will provide much stronger evidence for occurrence of the object than the others, and it hence will play a more important role in recognizing the object. To evaluate the discriminating power, we argue that a saliency coefficient should be associated with each feature-and-object pair provided the set of model object is given. Accordingly, the record of the entry corresponding to feature F should be rewritten as

$$\mathcal{E}(\mathbf{k}_F) =< \{M_a, C_a\}, \{M_b, C_b\}, \cdots >, \tag{2}$$

where $C_a, C_b, \cdots$ are values of the saliency coefficients.

The use of feature saliency has received attention of many researchers ([6], [7], [8]). Some algorithms have been proposed for evaluating feature saliency for edge points or 3-D data points on the basis of comparison of edge or surface shapes ([11], [13]). However, there is little attention that has been paid for computing the coefficients for 3-D structural features. We have designed an algorithm for performing the computation for *unary constraints* on surface features ([12]), and, in the paper, we shall modify the algorithm to take into account the *binary constraints* on relations between, or among, features.

3 Pose-Invariant Scene Description

3.1 Patch-Based Object Description

A main factor dominating performance of model-based systems is the level of the used model features. In our method, we assume both sample and sensor images include only objects that can be described as a list of isolated surface patches. Suppose a sample image is segmented into m patches $\mathcal{P}_1$, $\mathcal{P}_2$, $\cdots$, $\mathcal{P}_m$. Then, the scene $\mathcal{S}$ projected to the image can be represented as

$$\mathcal{S} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}. \tag{3}$$

It has been shown that the patch-based description is a compact scene representation that has the following properties.

- A significant percentage of manufactured parts can be perfectly represented or well approximately by a small number of surface patches of simple shapes.
- The description is easy to obtain by using 3-D sensor data such as range images ([4], [8]).
- The description preserves many global properties of the described objects. It assures that absence of some surface points in a sensed image will not cause harm to tasks such as object identification.

Given isolated patches in a segmented image, we need also to describe them to get a complete description of the scene. In general, translation- and rotation-invariant (pose-invariant) descriptions are prefarable because they are not susceptible to the changes of object poses in a 3-D space. In the following, we shall discuss on some of such pose-invariant constraints on patch description.

3.2 Unary Constraints on Patches

To describe a scene informatively, it is necessary to specify the types and parameters of patches in the segmented image. In the method, we suppose that only four types of surfaces – planar, spheric, cylindrical, and conic surfaces – appear on the objects to be modeled. Then, a patch $\mathcal{P}_i$ in a sensor-oriented coordinate system can be described by the identifications of its primitive type, general type, and a triplet $P_i =< \mathbf{l}_i, \mathbf{o}_i, s_i >$, where $\mathbf{l}_i, \mathbf{o}_i$ and s_i indicate, respectively, its location and orientation vectors and its size.

Using the sensor-oriented description, we represent the pose-invariant unary constraints on the patch by a *feature vector* as

$$\boldsymbol{k}_u(\mathcal{P}_i) = (k_{i1}, k_{i2}, k_{i3}), \tag{4}$$

where $k_{ij} (1 \leq k_{ij} \leq N_{ij})$ are values of the parameters that are chosen to describe the patch. All of the combinations of k_{ij} forms the *feature space* of the unary patch description and its size Z is given by $Z = \prod_{i=1}^{3} N_{ij}$.

The feature vector is defined as follows.

- k_{i1}: **primitive type:**

$$k_{i1} = \begin{cases} 1: & \text{if } \mathcal{P}_i \text{ is a planar one;} \\ 2: & \text{if } \mathcal{P}_i \text{ is a spheric one;} \\ 3: & \text{if } \mathcal{P}_i \text{ is a cylindrical one;} \\ 4: & \text{if } \mathcal{P}_i \text{ is a conic one} \end{cases} \tag{5}$$

- k_{i2}: **general type:**

$$k_{i2} = \begin{cases} 1: & \text{if } \mathcal{P}_i \text{ is a convex one;} \\ 2: & \text{if } \mathcal{P}_i \text{ is a concave one.} \end{cases} \tag{6}$$

- k_{i3}: **patch size:**

$$k_{i3} = \Gamma(k_M \frac{s_i - s_N}{s_M - s_N}), \tag{7}$$

where $\Gamma(x)$ is a function quantifying x into integer, k_M the maximum for the size description, s_M and s_N the upper and low bounds of the size, respectively.

Fig.3 shows the sample images we used in our experiments and, by assuming $k_M = 4$, the Scene-3 in the figure is described as

$$\mathcal{S}_3 = \{\mathcal{P}_{31}, \mathcal{P}_{32}, \mathcal{P}_{33}\} = \{(1,1,2), (2,1,3), (3,1,3)\}. \tag{8}$$

3.3 Binary Constraints on Patches

Binary constraints describes relations between patches extracted from sample images. Suppose we have two patches $\mathcal{P}_i$ and $\mathcal{P}_j$, which are described as

$$P_i =< \mathbf{l}_i, \mathbf{o}_i, s_i >, P_j =< \mathbf{l}_j, \mathbf{o}_j, s_j >, \tag{9}$$

respectively, in the sensor-oriented coordinates. The pose-invariant binary constraints are then described as a *relation feature vector* as

$$\boldsymbol{k}_b(\mathcal{P}_i, \mathcal{P}_j) = (k_{ij1}, k_{ij2}), \tag{10}$$

where k_{ij1} is combination of the primitive types of the two patches, and k_{ij2} a pose-invariant parameter. The following shows a simple explanation of the definitions.

- k_{ij1}: **combination of primitive types:**

  The values of k_{ij1} are shown in Table 1, where plane is denoted as P, sphere as S, cylinder as Y, and cone as O.

- k_{ij2}: **relation parameters:**

$$k_{ij2} = \Gamma(k_M \frac{r_{ij} - r_N}{r_M - r_N}), \tag{11}$$

  where $\Gamma(x), k_M, r_M$ and r_N are defined in the same way as the size computation in the unary constraint description. r_{ij} is the distance between the locations, or angles between the orientation vectors, of the two patches. It has been shown that such parameters are pose-invariant and can be easily computed from the sensor-oriented description given by eq.(9).

TABLE I VALUES OF k_{ij1}

patch pair	P P	P S	P Y	P O	S S	S Y	S O	Y Y	Y O	O O
k_{ij1}	1	2	3	4	5	6	7	8	9	10

In general, the relational constraints can be extended to groups of more than two patches. We use only binary constraints here mainly because large groups of patches will lead to large numbers of the table dimensions and it will complicate the learning process for saliency computation.

4 Neural Saliency Computation

Many factors in an object description have influence on occurrence of model objects. Some researchers have attempted to compute saliency of object features on the basis of geometric or statistic properties of the features ([7], [11]), or from the viewpoint of patch visibility ([8]).

Theoretically, this saliency computation task can be considered as a learning process based on some given training patterns. We have designed an algorithm using the feature learning concept and implemented it with an artificial neural network. As shown in Fig.1, the network has three layers of cells. Each input cell is associated with each entry of the considered interpretation table. After the description of a training image is extracted, the input cells associated with the patches in the description are activated. That is to say, given the extracted image description, we set the input signals to the input cells related to the registered features to 1, and 0 otherwise. The output cells of the network are associated with the possible model objects. We know from the sample image what objects are occurring and the teacher signals showing the occurrence of the model objects can be then determined by the knowledge.

There are a training phase and a performance phase in the process of computing saliency coefficients of the table entries. The training phase is intended to memorize the patch occurrence in the training images by modifying the

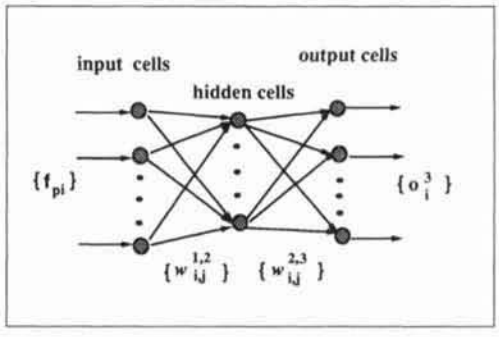

Fig.1. Neural Network Used in Saliency Computation.

weight coefficients of the net pathways. Let the weight coefficient of the path out from i-th cell of the $(k\text{-}1)$-th layer and to the j-th cell of the k-th layer be denoted by $w_i^{k-1}{}_j^k$. The cell output signals are computed by

$$o_i^k = f(\sum_j w_i^{k-1}{}_j^k o_j^{k-1}), \tag{12}$$

where

$$f(t) = \frac{1}{1 + \exp(-t/u_0)}. \tag{13}$$

Then, by using the teacher signals derived from the training image, the weight coefficients of the pathes are modified by the difference of the teacher signal y_i and the output signals o_j^3 from the output cells as

$$\Delta w_i^{k-1}{}_j^k = -\varepsilon d_j^k o_i^{k-1}; \tag{14}$$

$$d_j^3 = (o_j^3 - y_j) f'(i_j^3); \qquad d_j^k = (\sum_l w_j^k{}_l^{k+1} d_l^{k+1}) f'(i_j^k), \tag{15}$$

where ε is a small positive number specifying magnitude of each modification. When a training image is registered, the network starts the modification according to the rules given above repeatedly until the error signal between the output and teacher signals is less than a specified value. This learning algorithm is based on the typical back-propagation technique whose convergence is discussed in almost every text about neural network theory (eg. [9]).

In the performance phase, the saliency coefficients are calculated by the forward performance of the trained network. Assume the table has n_m entries. Then, n_m input patterns are formed so that only one input cell is activated in each pattern. The input patterns are then shown to the input layer, and the output signals are taken as values of the saliency coefficients of the corresponding entries for recognizing the considered model objects.

Whenever a new model object is taken into account, we have to establish one more output cell in the network and restart the training phase all over again. Since the model generation process is usually carried out by off-line operations, the computational time required in the restarted process is not problematic.

5 Exeperimental Results

We have made some experiments on synthetic scene descriptions to show the feasibility of the proposed method. Results of one of them are illustrated in the following.

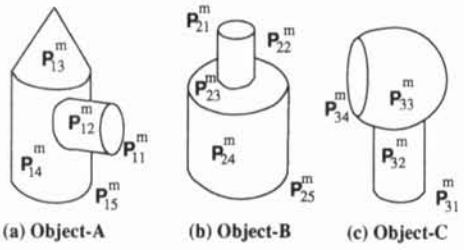

Fig.2. Model Objects Used in the Experiments.

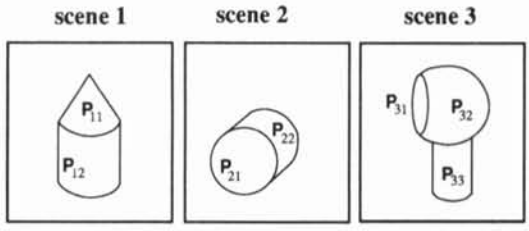

Fig.3. Three Training Images Used in the Experiments.

The model objects supposed to appear in the observed scenes are Object-A, B, and C shown in Fig.2. Six training scenes, three of which are shown in Fig.3, are formed to carry out the training of the saliency computation network. In the example, we consider only the learning for the interpretation table constructed for unary constraints. The number of the input and output cells are chosen as 32 and 3, respectively. The number of the hidden cells are 16 and the parameter u_0 in eq.(13) is 1.0. After the training patterns are shown to the network one after another, the network reaches its stable state after 238 times of modification of the weight coefficients.

After the training process, the outputs of the network for the input patterns corresponding to each table entry is measured. The saliency coefficients of the patch-and-object pairs are derived as illustrated in Table 2, 3, and 4, respectively.

The results show that the saliency coefficients related to the patches belonging to only one model object have much larger values than some others. This is consistent with the meaning the coefficients are expected to have.

6 Conclusion

The work is aimed at solving the problem of feature saliency computation involved in an automatic model generation. An algorithm based on neural networks has been proposed for accomplishing the computation.

Feature saliency computation is a very useful technique in dealing with 3-D object recognition problem. By using the method, it is sufficient, in many recognition tasks, to match a small number of patches of greater saliency for determining occurrence of specified model objects. In particular, the saliency concept is very suitable for interpreting images containing occluded surfaces since the absence of some less important patches does not hinder the whole recognition task. In the future work, we shall try to apply the saliency computation algorithm to other types of invariant features, such as 2-D edges. It is also worth while to investigate how to use the computed saliency coefficients in general constraint satisfication parameter nets.

TABLE II SALIENCY COEFFICIENTS FOR PATCHES ON OBJECT A

	$\mathcal{P}^m_{11}$	$\mathcal{P}^m_{12}$	$\mathcal{P}^m_{13}$	$\mathcal{P}^m_{14}$	$\mathcal{P}^m_{15}$
$Object-A$	0.80	0.99	0.99	0.80	0.43
$Object-B$	0.33	0.06	0.02	0.05	0.02
$Object-C$	0.00	0.00	0.01	0.04	0.37

TABLE III SALIENCY COEFFICIENTS FOR PATCHES ON OBJECT B

	$\mathcal{P}^m_{21}$	$\mathcal{P}^m_{22}$	$\mathcal{P}^m_{23}$	$\mathcal{P}^m_{24}$	$\mathcal{P}^m_{25}$
$Object-A$	0.80	0.07	0.06	0.05	0.06
$Object-B$	0.33	0.72	0.89	0.95	0.89
$Object-C$	0.00	0.02	0.02	0.00	0.02

TABLE IV SALIENCY COEFFICIENTS FOR PATCHES ON OBJECT C

	$\mathcal{P}^m_{31}$	$\mathcal{P}^m_{32}$	$\mathcal{P}^m_{33}$	$\mathcal{P}^m_{34}$
$Object-A$	0.43	0.80	0.00	0.43
$Object-B$	0.02	0.05	0.26	0.02
$Object-C$	0.37	0.04	0.90	0.37

REFERENCES

[1] D. H. Ballard, "Parameter Nets", *Artificial Intell.*, vol.22, pp.235–267, 1984

[2] R. M. Bolle, A. Califano, and R. Kjeldsen, "A Complete and Extendable Approach to Visual Recognition", *IEEE Trans. PAMI*, vol.14, pp.534–548, 1992

[3] R.T. Chin, and C.R. Dyer, "Model-Based Recognition in Robot Vision", *Computing Surveys*, vol.18, pp.67–108, 1986

[4] T.J. Fan, G. Medioni, and R. Nevatia, "Matching 3-D Objects Using Surface Descriptions", *Proc. ICRA*, pp.1400–1406, 1988

[5] P. J. Flynn, and A. K. Jain, "3D Object Recognition Using Invariant Feature Indexing of Interpretation Tables", *CVGIP: Image Understanding*, vol.55, pp.119–129, 1992

[6] W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, 1990

[7] D. G. Lowe, "Four Steps toward General-Purpose Robot Vision", *Robotics Research: The Fourth Int. Symposium*, (eds. R. Bolles and B. Roth), MIT Press, pp.221–228, 1988

[8] T. Nagata, and H. B. Zha, "Recognizing and Locating a Known Object from Multiple Images", *IEEE Trans. RA*, vol.7, pp.434–448, 1991

[9] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley Publishing Company, Inc., 1989

[10] T. Skordas, and R. Horaud, "Planning a Strategy for Recognizing Partially Occluded Parts", in *Proc. 8th IJCPR*, pp.1080–1083, 1986

[11] J. L. Turney, T. N. Mudge, and R. A. Volz, "Recognizing Partially Occluded Parts", *IEEE Trans. PAMI*, vol.PAMI-7, pp.410–421, 1985

[12] H. B. Zha, T. Nagata, and K. Kumamaru, "Automatic Generation of 3-D Curved Object Models by Using the Learning Mechanism of Neural Networks", *Engineering Systems with Intelligence* (ed. S. G. Tzafestas), Kluwer Academic Publishers, pp.213–220, 1991

[13] H. B. Zha, T. Nagata, and K. Kumamaru, "Quantifying Saliency of Feature Points on 3D Curved Surfaces from Range Images", *Proc. IEEE ICRA*, pp.1695–1700, 1992