AN ADAPTIVE MODEL FOR VIEWPOINT-INVARIANT OBJECT RECOGNITION

Peter A. Sandon
Department of Math. and Computer Sciences
Dartmouth College

Leonard M. Uhr
Computer Sciences Department
University of Wisconsin - Madison

INTRODUCTION

When we look at a familiar object from a novel viewpoint, we are usually able to recognize it. We are interested in developing a model of vision which can efficiently represent the invariant information required to recognize objects from various viewpoints, and which is capable of acquiring this information through experience. In the work described here, objects are modeled in terms of 2-D shape features. Using a hierarchical decomposition of object shapes allows parallel extraction of sub-shapes, provides storage economy for object models, and facilitates generalization of learned knowledge from one object to another.

The problem of viewpoint invariant vision can be stated as follows: The shape of an object as seen from an arbitrary viewpoint is some rigid-body transformation of visible shape features of some canonical shape of the object. Given a particular shape in an image, the vision system must recognize the object in the image as an instance of the canonical shape. To solve the recognition problem, viewpoint invariant features must be computed from the image data. Such features we refer to as being object-centered, since they have meaning relative to the object itself rather than to their image appearance. The use of object-centered features simplifies the recognition process by representing aspects of the object that do not depend on imaging parameters. In addition, object-centered representation allows more powerful generalization capabilities due to the similarity of representation within an object class.

The model is implemented in a connectionist network, in which nodes, singly or in groups, represent shape features and links represent evidential relations among shape features. We use an error correction learning method to train the network by example. The most commonly used multi-layer algorithm is the generalized delta rule (GDR). Since this algorithm is weak in a number of ways, a number of modifications to this rule have been developed and applied to this task. These modifications involve the addition of local constraints to the global error reduction constraint normally used to drive the learning. The details of these *error modification* (ErrMod) methods are reported elsewhere [Sandon 1987].

In the following sections we review some work related to our own, and then present the network model, and some simulation results. The main result involves the training of the upper layers of the network, where the desired generalization across translations of objects is demonstrated. This generalization capability eliminates the need to expose the system to every object under every transformation in order to obtain complete recognition.

RELATED WORK

Two related models that have been described in recent years [Hinton 1981, Ballard 1984] attempt to cooperatively identify the transformation and recognize the canonical shape simultaneously. A partial identification of the transformation can be used to constrain the possible interpretations of the object shape and vice versa. In the present work, we combine the representation of transformation information suggested in the Hinton and Ballard models with the hierarchical representation of shape used by Uhr [1972]. A major constraint on the resulting model is that it be amenable to learning of both shape and transformation.

Uhr's Recognition Cone

The recognition cone is an example of a parallel-hierarchical vision model. The recognition cone consists of layers of *transforms*. Each transform produces an output value that is some characteristic function of its input values. In general, the output values of each transform are used as input values to other transforms. The transforms in the lowest layer compute properties of the image pixels, those in the next lowest layer compute properties of these lowest layer transform properties, and so forth. Higher layers in the recognition cone are logarithmically smaller than lower layers, giving a hierarchical, pyramidal structure. Information is converged as it proceeds up the cone, so that higher level transforms, performing local operations on the information below, compute more successively global properties of the image than do transforms at lower layers. The properties computed by any given transform are stored locally, and/or passed up to be accessible to transforms at higher layers. In addition to the bottom-up processing, the property computed by a transform can be passed down the pyramid to provide feedback to earlier layers.

The recognition cone has much in common with the connectionist paradigm. It can be described as a connectionist network by making the following correspondence. Each transform is implemented by a single node in the network. A given transform accesses the result of another transform through a connection. The connectionist interpretation is more restricted than the original recognition cone in the complexity of transforms that can be directly implemented, in the use of local memory and in the control structure that is used. The local and layered nature of the recognition cone transforms, however, makes them amenable to this connectionist interpretation.

The Hinton / Ballard model

An approach which *explicitly* represents both the transformation invariant shape and the transformation itself is that proposed by Hinton [1981] and extended by Ballard [1984]. In this connectionist model, processing units are grouped into three distinct sets (see Figure 1), referred to as the retina-based frame, the object-based frame and the mapping units. The retina-based units represent features extracted from the image with spatial relations represented relative to the imaging device. The object-based units represent spatial relations relative to the object, without regard to the particular image representation. The mapping units represent the

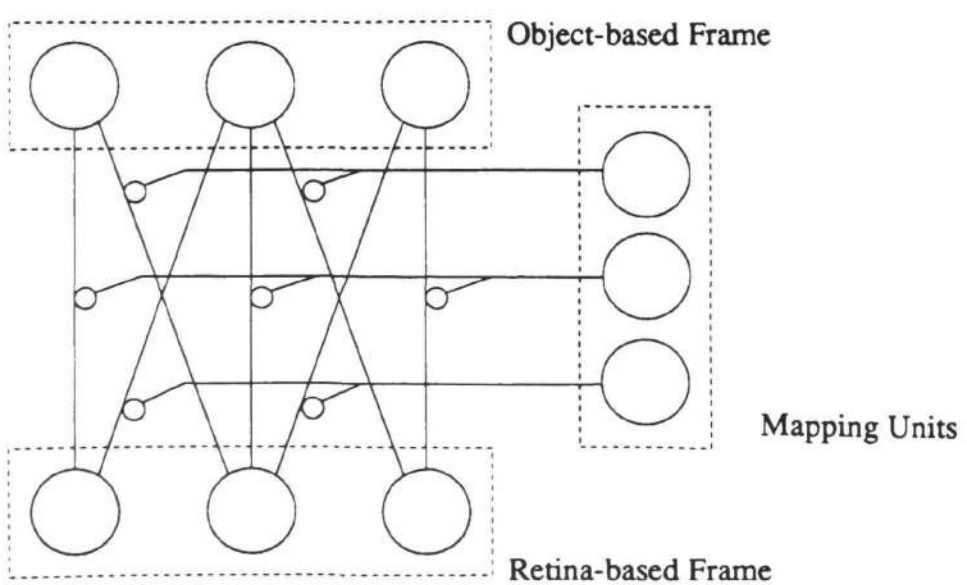

Figure 1

transformation between the retinal shape and the canonical coordinate frame represented in the object-based frame.

By explicitly representing the one transformation associated with an object, the network implements what Hinton refers to as the single viewpoint constraint. Retina-based units compute features based on properties of the image itself. Object-based units combine retina-based and mapping unit features to compute object-centered features. The small circles in Figure 1 represent the conjunctive modification of the retina-based to object-based connection by the mapping units. Similarly, mapping units use information from both retina-based and object-based units to compute the current object transformation. This computation is not represented in the figure. The interdependence of the object-based and mapping units requires a cooperative computation in which a partial result in one set of units improves the result in the other set.

NETWORK MODEL

The basic structure of our model is that of a pyramid-like primary network in which shape features are hierarchically represented, augmented with a secondary 'context' network in which transformation information is represented. The transformation information is conjunctively combined with shape features, at various levels of the pyramid, to produce representations of shape that are successively more object-centered in higher layers of the network. The gradual transition from strictly retina-based to strictly object-based features has two advantages. First, the increased connectivity due to conjunctive combination is spread over several layers. Second, local connectivity is maintained, which allows the recognition cone to discriminate features of locally, and successively more globally, interacting objects. We refer to all layers of the shape pyramid below the point where context information is introduced as retina-based layers. All layers above any use of context information are object-centered layers. Those layers in between are transition layers.

The particular instantiation of this model that has been simulated is a 2-D translation network which succeeds in recognizing various stick-figure patterns under all translations within a small image plane. (see Figure 2). Layer 0 is the input image. Shape features are represented in layers 1 through 4, with layer 5 representing objects to be recognized. The transformation (in this case, translation) of the object is represented in layers 1^c through 4^c. The outputs of the upper two layers of the context network are conjunctively combined with shape features at layers 3 and 4 in the shape hierarchy.

In the Hinton model, it is suggested that both shape and transformation can be extracted from the image. This requires that the identity of the shape be used to define the transformation, and that the identity of the transformation be used to define the shape. These mutually dependent definitions of shape and transformation require feedback paths and a relaxation process for computation by a network. In order to maintain the feedforward structure of the network, we do not include the feedback term from the object-centered features in computing the transformation. For the objects used in our simulations, the network is able to compute the transformation without this feedback term.

Each layer in the network consists of a compact square array of competing node clusters. A cluster is composed of a number of nodes, typically from 2 to 9, which compete for adjustments to their connection weights through the error modification learning mechanism. Each node is implemented as a logistic processing element [Rumelhart, et. al. 1986]. The input image is 15 pixels on a side. Layers 1-5 and 1^c to 4^c are composed of 13, 11, 5, 3, 1, 9, 7, 3 and 3 clusters on a side, respectively. The size of the cluster in the output layer (layer 5) ranges from 6 to 50 in various simulations.

Due to the complexity of this network and the various characteristics of the network structure and learning algorithms to be demonstrated, we have simulated separate pieces of the network, which we now describe.

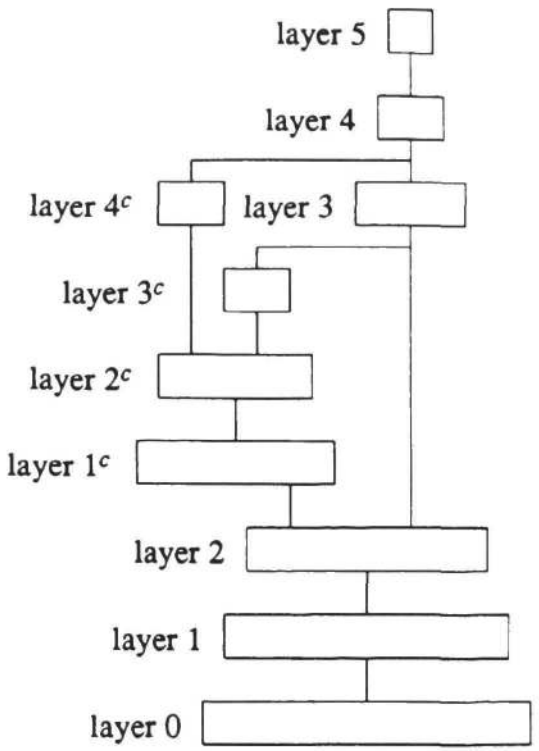

Figure 2

EXPERIMENTS

Learning object-centered representations

The first learning simulation uses the top three layers of the shape pyramid (layers 3-5), and the top layer of the context network (layer 4^c). This simulation demonstrates that object recognition can be learned through the development of object-centered feature detectors and explicit representations of location information.

To simulate this 3-layer network, we provide input directly to layers 3 and 2^c. Patterns are provided to various intermediate layers of the network in the experiments described. The patterns chosen are consistent with a particular interpretation that can be represented by the nodes at that layer, though many other representations are possible. One of six 'letter' patterns (see Figure 3) in one of nine locations is presented in layer 3. A single activation of one of nine nodes in layer 2^c represents the location information for the context subnetwork. Using these six shape patterns and nine locations yields a set of 54 images to be presented to the network. These are split into a set of 27 training patterns and 27 test patterns. A response is considered correct if the maximally active output node corresponds to the correct pattern class and has an output value greater than .5.

In the first experiment, learning of the 27 training patterns proceeds quickly, yielding 89% performance in 500 cycles using GDR. However, when the remaining 27 patterns are presented to the network as a test set, only 4 are correctly classified. This combination of relatively fast learning and weak generalization indicates that the capacity of the network to store patterns is high compared to the number of patterns to be stored. This allows the network to perform well by representing each pattern individually, rather than as a set of shared features based on the regularities intrinsic to the pattern collection. The result is rote learning, which lacks generalization capability. To overcome this problem, the capacity of the network is limited in the succeeding experiments by using only six of the eighteen nodes of layer 4.

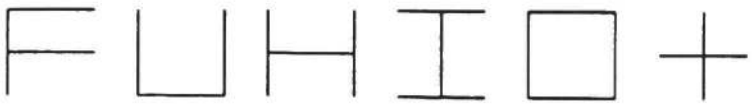

Figure 3

Repeating the training experiment using GDR on the 27 training patterns results in 75% recognition performance after 5000 cycles. Error modification improves this result to 96% performance after 5000 cycles. The reduced capacity of the network has increased the difficulty of the learning task, as expected. However, in the simulation using error modification, only two of the nine context nodes are utilized, in the sense that they are active for some patterns and not for others. Since the location information is implicit in the image, the primary network is capable of solving much of this task without the explicit representation of location supplied by the context network. This will not lead to the desired object-centered features, however, which are generalizations over object locations. The results of applying the adapted network to the test patterns are again 4 of 27 correct.

In the next experiment we use the full set of 50 shape patterns and 9 locations for a total of 450 image patterns. In order to achieve good recognition performance on this task, the network must represent pattern features more efficiently than in the previous experiment. We use a training set of 150 patterns leaving 300 patterns as a test set. The set of 150 training patterns includes 3 presentations of each of the 50 fixed patterns in three different locations.

The increased number of patterns makes this task more difficult than the previous one. Using GDR, only 12% of the patterns are correctly recognized after 20000 cycles of training. Using error modification, performance on the training set reaches 100% after 12000 cycles (80 presentations per pattern). Table 1 presents the results of running a set of 10 testcases using GDR and error modification. In the GDR simulation, the network did not learn to correctly classify enough of the training patterns to demonstrate any significant generalization. In the error modification simulation, seven of the ten testcases resulted in perfect classification of the training patterns, and each of those testcases demonstrated a strong generalization to the test patterns.

Table 1 - Learning performance on training and test patterns
% performance after 20000 cycles

Testcase	GDR		ErrMod	
	training	test	training	test
A	37	6	100	100
B	40	6	90	44
C	35	6	76	23
D	38	9	100	99
E	31	11	100	100
F	38	8	100	100
G	42	6	92	40
H	31	10	100	100
I	30	8	100	91
J	39	11	100	100

Generalization allows unknown patterns to be correctly classified, as demonstrated above. In addition, generalization leads to efficient learning since some knowledge of learned patterns is transferred to related unknown patterns. There are a number of ways to demonstrate this transfer. Table 2 presents the number of cycles needed to reach maximum performance for various training set sizes, using testcase A. If no generalization takes place, we expect the time needed to train the network to increase as the size of the training set increases. However, Table 2 shows that the time needed to train the network on all 450 patterns is less than that needed to train on 100 patterns. This indicates strong generalization among patterns which enhances the learning of additional patterns which fit the generalization.

The set of experiments described in this section demonstrates the key network capability that is desired for recognizing familiar objects from novel viewpoints.

Other sub-networks

We now describe additional simulations that demonstrate the behaviors of other sub-networks.

Context. To demonstrate the learning of context information from shape features we simulated the network composed of layers 0, 1, 2, 1^c and 2^c of the 2-D Translation network. Patterns are presented as activations of the input nodes of layer 0. The weights of the nodes in layers 1 and 2 are predetermined to extract simple line features from the image. The input patterns presented are those corresponding to 17 of the 50 shapes and 25 of the 49 locations used in the previous simulation. The desired output, at layer 2^c is to have a single active node corresponding to one of 49 locations of the shape in the image.

This two-layer learning task turns out to be fairly easy for the GDR algorithm. The network is able to reach 100% performance in 4500 cycles.

Retinotopic. The second simulation involves the lower layers of the network, where feature representations are purely retina-based. These layers are 4 and 5 layers removed from direct training. This results in very weak training signals from the layers above. For this reason, we apply the methods of error augmentation [Sandon 1987], which combines top-down error driven learning with bottom-up stimulus driven learning.

For this simulation, we adapt the first four layers of the shape network. Input patterns are presented at layer 0. At layers 1 and 2, an error augmentation algorithm is used to adapt the weights. At layers 3 and 4, GDR is used. Training input is provided directly to layer 4.

Using a set of 100 patterns, each consisting of one of 47 shapes at one of 25 locations, the performance of this network reaches 61% after 20000 cycles. The significance of this simulation is in demonstrating that the low level features, developed mostly through a bottom-up process, are sufficient to produce the features at higher layers that are required by the translation-invariant recognition task. Using the self-organizing component alone in the lower two layers yields a performance of only 11%.

Transition. The final simulation concerns the gradual transition of retina-based to object-based features. For this purpose, we simulate layers 2, 3, 4, 2^c, 3^c and 4^c of the 2-D Translation network. Inputs are supplied

Table 2 - Learning performance of ErrMod
Maximum performance
and cycles needed to achieve it

size	% performance	cycles(x1000)
50	99	20
100	96	20
150	100	12
200	100	14
450	100	10

directly to layers 2 and 2^c, while training input is provided to layer 4.

Learning in this sub-network is difficult because it involves two context layers comprising one two-layer and one three-layer backpropagation path. Most previous work using error-correction learning in layered networks has been applied to two-layer networks without conjunctive connections. Error modification is used at both context layers to obtain sufficient differentiation of function among the nodes to allow learning to take place. In addition, error augmentation is used in the context layers due to the length of the backpropagation path.

This simulation uses 100 input patterns each consisting of one of 47 shapes and 9 locations. After 20000 cycles, network performance reaches 80%, but it does not improve during an additional 10000 cycles. Although the combination of learning algorithms leads to only 80% performance on this task, the result is encouraging considering the difficulty of the task.

CONCLUDING REMARKS

We have described a network model of shape classification that is capable of learning to recognize objects under various translations, including novel ones. The 2-D translation network successfully learns to "recognize familiar objects from novel viewpoints" by developing object-centered representations of the shapes through the explicit representation of location. These transformation-invariant features support the necessary generalization of shape information across locations.

In the simulations that have been described, we have had to apply input and training patterns at intermediate layers of the network. Our ability to define such patterns may imply an understanding of this particular problem which obviates the need for learning. However, we hypothesize, and intend to demonstrate, that the same network structure can be applied to a problem involving non-rigid transformations, such as recognition of handwritten letters. In such a task, the choice of *a priori* representations for each layer of the network would be extremely difficult, making learning a crucial part of the modeling process.

REFERENCES

. Ballard, D. H., "Parameter Nets," *Artificial Intelligence* **22** pp. 235-267 (1984).

. Hinton, G. E., "A Parallel Computation that Assigns Canonical Object-Based Frames of Reference," *Proc. 7th IJCAI*, pp. 683-685 (1981).

. Rumelhart, D. E., G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," pp. 318-362 in *Parallel Distributed Processing Volume 1*, ed. D. E. Rumelhart and J. L. McClelland (eds.),Bradford Books, Cambridge, MA. (1986).

. Sandon, P. A., "Learning Object-Centered Representations," PhD. Dissertation, Univ. Wisconsin - Madison (August 1987).

. Uhr, L., "Layered 'recognition cone' networks that preprocess, classify and describe," *IEEE Trans. Comput.* **21** pp. 758-768 (1972).