VISOR: Schema-based Scene Analysis with Structured Neural Networks *

Wee Kheng Leow and Risto Miikkulainen
Department of Computer Sciences
The University of Texas at Austin
Austin TX 78712 USA
Email leow,risto@cs.utexas.edu

Abstract

A novel approach to object recognition and scene analysis based on neural network representation of visual schemas is described. Given an input scene, the VISOR system focuses attention successively at each component, and the schema representations cooperate and compete to match the inputs. The schema hierarchy is learned from examples through unsupervised adaptation and reinforcement learning. VISOR learns that some objects are more important than others in identifying the scene, and that the importance of spatial relations varies depending on the scene. As the inputs differ increasingly from the schemas, VISOR's recognition process is remarkably robust, and automatically generates a measure of confidence in the analysis.

1 Introduction

The basic paradigm in many image understanding systems is the "hypothesize and test" technique [3, 14]. The system contains a large number of models that represent typical objects and scenes. Based on the input image, the system invokes models that are likely to match the scene well—a process called hypothesis activation. In the process, expectations or beliefs are generated about the scene, which may in turn activate other hypotheses. The process continues until the system discovers a set of consistent hypotheses that best matches the scene. This set constitutes an interpretation of the scene.

The "hypothesize and test" technique involves tedious and complex search procedures. Alternatively, it can be formulated in terms of interactions among visual schemas [6, 8]. The models are replaced by object and scene schemas that cooperate and compete to match the inputs. Data-driven and expectation-driven activation of hypotheses becomes simple bottom-up and top-down activation of schemas.

This paper describes how such cooperation, competition, and parallel bottom-up and top-down processing can be implemented in neural networks, making use of their natural robustness and learning properties.

2 The VISOR System

The VISOR system (VIsual Schemas for Object Representation; [11, 12]) consists of three main components: (1) the Low-Level Visual Module extracts positional and featural information from the scene, (2) the Schema Module matches schemas with the inputs, and (3) the Response Module generates object and scene labels as the recognition result (Fig. 1).

*This research was supported in part by NSF grant #IRI-9309273.

To appear in the *Neural Processing Letters* 1:18–23, 1994.

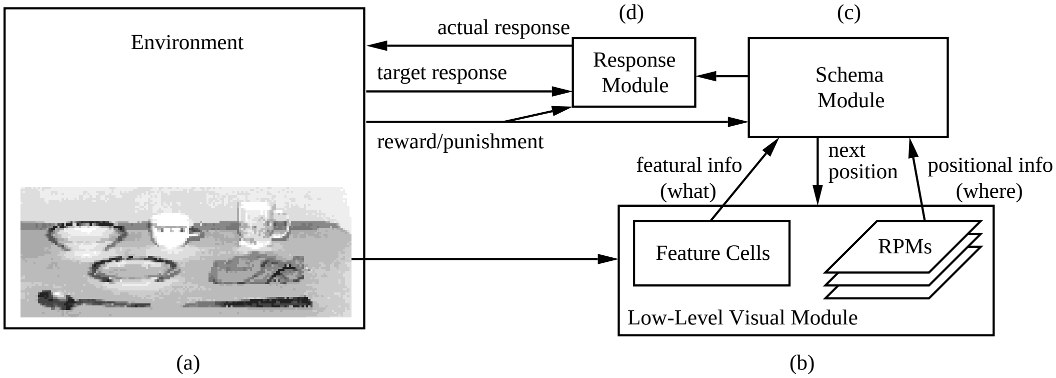

Figure 1: VISOR consists of the Low-Level Visual Module (LLVM, b), the Schema Module (c), and the Response Module (d). LLVM extracts "what" and "where" information from the scene (represented symbolically). The Schema Module performs scene analysis and the Response Module produces scene labels for the environment.

The Low-Level Visual Module (LLVM) has the task of providing input to the Schema Module. Because VISOR research concentrates on mechanisms of schema learning and processing, the LLVM is currently simulated procedurally, and it receives a symbolic representation of the real-world scene as input (consisting of a list of locations and shape attributes for the components in the scene). The LLVM focuses attention at one component of an object at a time and produces two outputs: the shape of the component encoded as a distributed activity pattern over the shape feature cells, and the position of the component represented in the Relative Position Maps (RPMs; Fig. 1). Following Biederman's theory of human object perception [2], the feature pattern describes shape attributes such as length, breadth, closure, vertical tilt, horizontal tilt, degree of expansion, and curvature. Each attribute is represented by a set of feature cells that are maximally sensitive to different ranges of values (e.g., small, medium, large). If the focused object is a region with approximately uniform texture but no specific shape (such as a grass patch), the texture of the region is extracted and encoded in the texture feature cells (Fig. 2). Based on the model of Bovik et al. [4], each texture feature cell corresponds to the output of a filter channel that is maximally sensitive to a particular type of texture.

The featural and positional information extracted by the LLVM are fed into the Schema Module (Fig. 1c) where it is matched with the schema representations. The Schema Module consists of a multi-layer network of schema representation nets, or schema-nets for short (Fig. 2). Each layer of schema-nets corresponds to a level in the schema hierarchy, with the scene schemas at the top and the object schemas at the bottom. Consider the representation of the spoon (Fig. 2a), which consists of two components: an elliptical part and a long, slim handle. The spatial layout of the spoon is represented by a two-dimensional map called the Sub-schema Activity Map (SAM) in the spoon schema-net. For example, the location of the spoon's handle is represented by the right middle SAM unit. The shape feature cells connect to the right middle SAM unit, and the weights of the connections encode the distributed pattern of a long, slim rectangle in the same manner as the LLVM's feature cells. Similarly, the weights of the connections to the left middle SAM unit encode a moderately large ellipse.

The objects in a table scene, such as a dining table (Fig. 1a), may be located anywhere in the scene. The SAM of such a scene schema encodes only one position representing the entire scene, but each position can contain more than one object (shown as a SAM column in Fig. 2). The connection from the spoon schema's output unit to the SAM unit of the dining table schema indicates that the dining table may contain a spoon anywhere in the scene. An object, such as a knife, may appear on both a dining table as well as a workbench. Therefore, the knife schema's output unit connects to both the SAM units of the dining table and the workbench schemas (Fig. 2).

The objects and regions in an outdoor scene, such as a road scene, each occupy a typical area of the scene, but there may be several different objects in each of these areas. The spatial layout of such

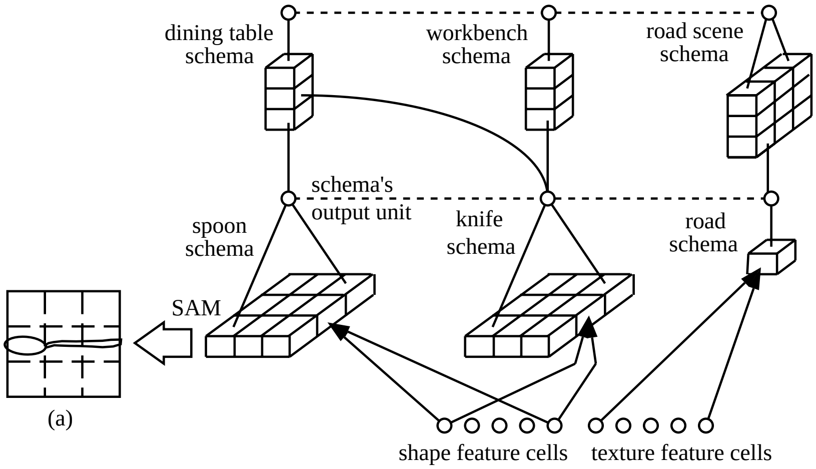

Figure 2: The hierarchy of schema-nets. Schemas' output units are represented as circles, and the units of the Sub-schema Activity Maps (SAMs) as cubes. Arrows represent one-way connections, solid lines represent both the bottom-up and top-down connections (which are different), and dotted lines denote the inhibition among the schemas' output units. (a) The SAM units representing the relative positions of the spoon's components.

a scene is encoded in a 3-D SAM which, for outdoor scenes, reduces to 3 vertical positions representing the sky, the middle portion of the scene, and the ground level (Fig. 2). Each position contains a SAM column that encodes the objects or regions that may appear in that portion of the scene. For example, Fig. 2 shows a road scene schema containing a road region at the bottom of the scene.

Let us see how VISOR processes a scene such as Fig. 1a. Given the scene representation as input, the LLVM focuses attention at one of the components in the scene, say the elliptical part of the spoon, and generates a feature representation of its shape in the shape feature cells. Next, the activation propagates up to the SAMs of the object schemas. Only those SAM units that match the current position in the Relative Position Maps are enabled (in this case the left middle SAM units), and they compare the featural inputs with their input weights by computing their weighted sum. The result is encoded in the SAM units' activities, which then propagate to the object schemas' output units. These units indicate how well the entire schemas match the inputs. In this example, the spoon schema is most strongly activated since its left middle SAM unit best matches the left component of the input object. In other words, VISOR believes that the input object is probably a spoon. After processing the inputs at the current position, the Schema Module selects the next position of attention where other components are expected, in this case the right middle position. The LLVM shifts attention to the new location, and the process repeats.

Schemas are activated both bottom-up and top-down. The spoon schema's activity is propagated upwards to activate the dining table schema. At the same time, the dining table schema propagates its activity back to the spoon schema to indicate that the spoon is indeed expected on the dining table. Receiving both bottom-up and top-down inputs, the spoon schema becomes more active and tends to decrease the activities of other schemas through inhibitory connections between schemas' output units. In this way, the schemas cooperate and compete to determine which one best matches the inputs.

The environment does not have to peek into the Schema Module to determine the recognition results. Instead, it receives the output response (a label) generated by the Response Module (Fig. 1d) based on the current activation of the schema hierarchy. The Response Module plays an important role in learning new schemas. For example, let us consider how VISOR learns to encode the spoon schema. The environment presents the image of a spoon to the LLVM and the *spoon* label to the Response Module as the target response (Fig. 1). As a result of the interactions among the schema-nets, one of them becomes most strongly activated. There are three different learning situations:

1. If the most active schema-net has not yet encoded any schema, the Response Module will produce no output response, and the environment will deliver a reward signal to VISOR. The schema-net

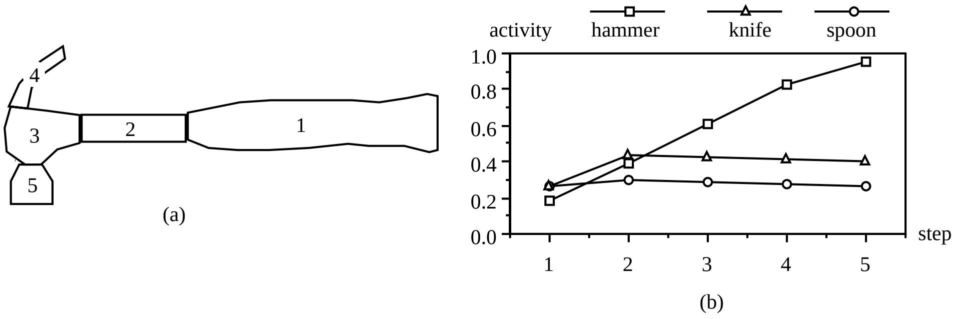

Figure 3: Processing a hammer image. (a) The sequence of positions of attention. (b) Activities of the object schemas. Initially, VISOR was unable to determine what the input object was. However, after looking at all the components, VISOR confidently concluded that it was a hammer.

weights adapt to encode the spatial structure of the spoon, and the Response Module learns to associate the activation of the newly formed spoon schema with the target label *spoon*.

2. If the most active schema-net happens to be the newly formed spoon schema, the Response Module will produce the correct *spoon* label as the response. The environment will deliver a reward signal to VISOR and weight adaptation takes place as in the first case.

3. If another schema, such as the knife schema, becomes most active, the Response Module will produce the *knife* label which is incorrect. In this case, the environment will deliver a punishment signal to VISOR, suppressing the knife schema-net's activation so that a different schema-net can become most active. The punishment signal is analogous to the mismatch-reset signal in the ART network [5]. It tells the Schema Module to find a different schema-net for the spoon without specifying which one.

VISOR's method of matching inputs through cooperation, competition, and parallel bottom-up and top-down activation of schemas gives rise to a particularly robust scene analysis process, as will be described below.

3 VISOR's Behavior and Properties

Several important properties automatically emerge from VISOR's neural network architecture. VISOR tolerates minor variations in the object structure, yet is able to distinguish between similar objects when necessary. VISOR can reliably recognize ambiguous scenes and indicate confidence of its analysis. In effect, VISOR uses schemas as soft constraints guiding the scene analysis process.

First, let us see how the schemas compete to match an input object. VISOR was shown the image of a hammer, and was told to start by focusing attention at its handle (Fig. 3). Since the shape of the handle looks similar to that of the knife and the spoon, VISOR's process began in a highly ambiguous state, and the schemas were activated by roughly equal amounts (step 1). The second component of the hammer resembles the knife's blade more than the spoon's elliptical part. As a result, at step 2, both the activities of the hammer and the knife schemas increased more than that of the spoon schema. However, the hammer schema's activity was still lower than knife's, because it needed more components. Disambiguation occurred at step 3 when VISOR focused attention at the claw, a component that exists only in the hammer. As VISOR focused successively at the remaining parts, the hammer schema's activity increased while those of the knife and the spoon schemas decreased due to competition among the schemas. After looking at all components, VISOR concluded that the image was most likely a hammer.

In real images, the actual inputs often differ slightly from prototypical examples due to normal variation, occlusion, damage to the input objects, noise, and so on. VISOR can recognize objects

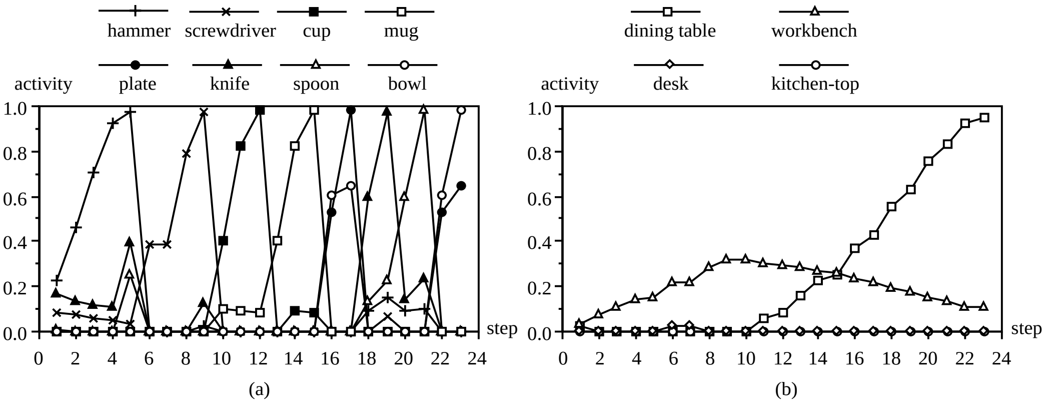

Figure 4: Processing a dining table scene with a hammer and a screwdriver. (a) The activities of the object schemas, and (b) those of the scene schemas.

despite such variations and also indicate how well the object matches the schema. During the training phase, VISOR learned to encode the hammer based on the perfect image shown in Fig. 3a. Variations of the hammer were then presented to VISOR for identification. When shown the perfect hammer, the hammer schema attained an activity level of 0.96 (Fig. 3a). Its activity remained at 0.96 in response to a hammer with an additional claw because VISOR simply ignored the extra component. When the claw of the hammer was straight instead of curved as in Fig. 3a, the hammer schema's activity was lowered slightly to 0.93. When the claw was missing entirely, the match between the input and the schema was not as good, and the schema's activity dropped to 0.83. Thus, VISOR can not only recognize variations of input objects but also indicate the confidence of its analysis.

However, when the spatial layout of the object differs too much from that encoded in the schema, VISOR will not regard the object as an acceptable variant. For example, if the entire handle of the hammer is missing, the object becomes much shorter and its spatial layout cannot be matched with that of the schema. VISOR would not recognize it as a hammer, which is usually a reasonable conclusion in a situation where a major component is missing.

Even though VISOR is quite insensitive to variations, it can also learn to pay attention to even minor variations if they are important in distinguishing between objects. For example, the standard pliers and the long-nose pliers differ only in the shapes of the jaws, making their input representations extremely close. VISOR encodes the difference in the schemas for these objects, which gives the correct schema an activation advantage. This difference is magnified through competition, and in the end, VISOR is able to clearly indicate the recognition result in the schemas' output values.

Let us now see how VISOR processes an entire scene, and how the cooperation and competition takes place among the object and scene schemas. During training, VISOR learned about dining table, workbench, kitchen top, and desk scenes and the objects in them. An unusual dining table scene was then presented to VISOR, containing a hammer and a screwdriver, which normally appear only in the workbench scene. VISOR was given a false lead by forcing it to focus attention at the hammer first. At step 5 (Fig. 4a), the hammer was identified, and VISOR believed that the input scene was probably a workbench. When the screwdriver was recognized at step 9, VISOR's initial belief was strengthened. After identifying the cup and the mug at steps 12 and 15, the activity of the dining table schema started to increase, and consequently reduced the workbench schema's activity through competition. At step 15, VISOR was very confused: the dining table and the workbench schemas were equally active. However, after recognizing the plate (step 17), the knife (19), the spoon (21), and the bowl (23), the activity of the dining table schema gradually increased, and the belief in the workbench schema faded away in the competition with the dining table. In the end, VISOR was very confident that the input scene was actually a dining table.

Some objects are more important than others in activating a schema. For instance, a hammer is a more important indicator for workbench than a knife because it appears more often in the scene. VISOR learns to encode object importance in the weights of the feedforward connections from the SAM units to the schema's output units. For example, if a hammer appears in a workbench scene 90% of the time and a knife 50% of the time during training, the workbench schema will receive 9/5 times more activation from hammer than from the knife. In other words, VISOR can judge how important an object is in identifying a scene, which further adds to its capability of representing "soft" schemas.

4 Discussion

The main research goal with VISOR was to develop methods for representing, learning, and applying visual schemas in neural networks. Connectionist schema systems have been proposed before [1], and there are systems that simulate parts of the process such as hierarchical structures [7, 9], cooperation and competition [13], and dynamic bindings [10]. An important contribution of VISOR is to show how schemas can be learned from examples, which is a problem that to our knowledge has not been addressed before.

The scale-up prospects of the approach seem promising. Obviously, the system has to have different schema-nets for each possible schema, but high-level schemas can share lower-level schemas as components, and each schema can recognize a large set of variations from the prototype. The competitive and learning processes do not seem to break down when the number of schema-nets is increased, because at any time, only a small number of nets are active participants in the competition and learning. In the experiments reported above, VISOR started with thirty $8 \times 8 \times 1$ bottom-level schema-nets and eight $3 \times 1 \times 10$ top-level nets, and learned to recognize 28 objects and 6 scenes from only 1 example of each scene. In our experience, the system is also very insensitive to parameters such as the learning rate as well as the inhibition and feedback coefficients, as long as they are within a reasonable range.

In comparison, a standard 3-layer backpropagation network was found to require over 100 times more presentations and an order of magnitude more input units to learn to classify the same set of input objects. Backpropagation was also very sensitive to parameters and initial weights, and could not generalize well to object variations where a component was missing. It would also be very difficult to extend the backpropagation approach to recognizing scenes where objects can occur in any location.

5 Conclusion

VISOR presents a natural neural network approach to object recognition and scene analysis based on visual schemas. The spatial structure of objects and scenes is learned from examples by a hierarchical network of schema-nets that cooperate and compete to match the inputs. Several interesting properties emerge automatically from such architecture. VISOR tolerates variations and ambiguity, indicates confidence of its analysis, learns that some objects are more important than others in identifying the scene. With these properties and behavior, VISOR is a promising first step towards a general vision system that could be used in different applications after learning the application-specific schemas.

References

[1] M.A. Arbib. *The Metaphorical Brain 2: Neural Networks and Beyond*, Wiley, New York, 1989.

[2] I. Biederman. Recognition-by-components: A theory of human image understanding, *Psychological Review*, vol. 94, pp. 115–147, 1987.

[3] T.O. Binford. Survey of model-based image analysis systems, *Robotics Research*, vol. 1, pp. 18–64, 1982.

[4] A.C. Bovik, M. Clark and W.S. Geisler. Computational texture analysis using localized spatial filtering, In *Proceedings of the Workshop on Computer Vision*, pp. 201–206, 1987.

[5] G.A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine, *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 54–115, 1987.

[6] B.A. Draper, R.T. Collins, J.Brolio, A.R. Hanson and E.M. Riseman. The Schema System, *International Journal of Computer Vision*, vol. 2, pp. 209–250, 1989.

[7] J.A. Feldman. Four frames suffice: A provisional model of vision and space, *Behavioral and Brain Sciences*, vol. 8, pp. 265–313, 1985.

[8] A.R. Hanson and E.M. Riseman. VISIONS: A computer system for interpreting scenes, In A.R. Hanson and E.M. Riseman, editors, *Computer Vision Systems*, Academic Press, New York, 1978.

[9] G.E. Hinton. Mapping part-whole hierarchies into connectionist networks, *Artificial Intelligence*, vol. 46, pp. 47–75, 1990.

[10] J.E. Hummel and I. Biederman. Dynamic binding in a neural network for shape recognition, *Psychological Review*, vol. 99, pp. 480–517, 1992.

[11] W.K. Leow. *VISOR: Learning Visual Schemas in Neural Networks for Object Recognition and Scene Analysis*, PhD thesis, Department of Computer Sciences, The University of Texas at Austin, 1994.

[12] W.K. Leow and R. Miikkulainen. Priming, perceptual reversal, and circular reaction in a neural network model of schema-based vision, In *Proceedings of the 16th Annual Conference of the Cognitive Science Society*, 1994.

[13] D.E. Rumelhart, P. Smolensky, J.L. McClelland and G.E. Hinton. Schemata and sequential thought processings in PDP models, In J.L. McClelland and D.E. Rumelhart, editors, *Parallel Distributed Processings*, MIT Press, Cambridge, Massachusetts, 1986.

[14] J.K. Tsotsos. Image understanding, In S.C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, Wiley, New York, 1987.