Visual Schemas in Object Recognition and Scene Analysis *

Risto Miikkulainen and Wee Kheng Leow
Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712

1 INTRODUCTION

Humans have the ability to rapidly and accurately recognize objects in a scene. We perform this task by matching visual inputs with object representations in our memory. These representations are not simply raw images in terms of light and dark pixels, but describe the spatial structure of objects. In many computational models, such representations are implemented as *visual schemas*, which are active functional units that cooperate and compete to determine which representation best matches the object. This article focuses on how visual schemas can be implemented in neural networks and how they can be used to model human object recognition and scene analysis.

2 BASIC CONCEPTS

Visual schemas describe objects in terms of the physical properties and the spatial arrangements of their components. Consider the schema for a deer for example. The real-world deer has several parts such as a head, a neck, a body, and four legs. Each part has a characteristic shape and size: the head is a small triangular block, the neck a long cylinder, the body a large rectangular block, and the legs are long and slim cylinders. The parts are arranged in more or less specific locations. The head is attached to one end of the neck, and its other end is attached to the body. The four legs are attached to the bottom of the body to support it. All such information must be represented in the schema for a deer.

*To appear in M. A. Arbib (Ed.), The Handbook of Brain Theory and Neural Networks, Cambridge, MA: MIT Press, 1995.

Recent psychological theories of human object recognition (Biederman, 1987) suggest that the shapes of the object components can be classified into about 36 different categories called *geons*. Geons are similar to the generalized cylinders used in machine vision for describing 3-D shapes (Marr, 1982). Whereas a generalized cylinder is modeled mathematically by sweeping a 2-D cross-section along a curvilinear axis, a geon is identified by a set of characteristics such as the shape of the cross-section, whether the size of the cross-section is uniform or expanding along the axis, whether the axis of symmetry is straight or curved, and so on. The sizes need not be very accurately measured, and they can be quantized into discrete intervals. Although the geon theory does not address certain aspects of human visual perception such as the recognition of faces and natural scenes, it captures many aspects of human object recognition and serves as psychological motivation for modeling object recognition in terms of schema systems.

Much of the schema theory has been developed in the symbolic framework (Arbib, 1989; Rumelhart, 1980). Based on the experience with symbolic schema systems such as VISIONS (Draper et al., 1989; Hanson and Riseman, 1978), it is possible to summarize the three main functional characteristics that visual schema implementations should have:

1. Visual schemas are organized into a *schema hierarchy*, with scene schemas at the topmost level, object schemas at the next level, and so on.
2. Schema instances representing the components of an object *cooperate* to support the instantiation of the object schema.
3. Schema instances representing different objects (or scenes) *compete* to determine which one best matches the inputs.

It turns out that these characteristics are very naturally implemented in neural networks. Moreover, the standard learning mechanisms of neural networks make it possible to build systems that learn schemas from examples, which is difficult to do in the symbolic framework.

3 VISUAL SCHEMAS IN NEURAL NETWORKS

Schemas are inherently structured representations, and representing structure in general is difficult with neural networks. There are three key ideas that allow us to approach the problem of representing visual schemas. First, part-whole relationships between an object and its parts can be represented by connections between localist units (Hinton, 1988). When a unit representing a part of an object is activated, its activity propagates to the unit representing the whole object. This process corresponds to bottom-up input. Conversely, through feedback connections from the object unit to the part units, the object can activate its parts, corresponding to top-down expectation.

Second, cooperative and competitive relationships among the schemas can be represented by connections among the part and schema units. This idea is adopted in the distributed schema system of Rumelhart et al. (1986). There are parts that can be found in several high-level schemas while others belong only to a single schema. For example, both dining room and kitchen may contain a table, but a bed typically exists only in a bedroom. Part units that belong to the same schema are connected with positive weights and cooperate to support the schema. Part units that never exist in the same schema are connected with negative weights and compete, trying to activate different schemas. Given that certain components, say a table and a bed, are activated by the input, the activity propagates through the connections and eventually stabilizes. The resulting activity over the network represents the activation of the best matching schema, such as the bedroom schema. Since the network contains only part units and has no distinct units for the different schemas, only one schema can be distinctly activated at any one time in this model.

The third idea concerns the so-called *binding problem* which occurs when a part unit is connected to several schema units that contain the same component. Several distinct objects in the scene may contain different instances of the same part. When the part unit is activated, the system has to determine to which object the part instance belongs. The binding problem can be solved by focusing attention at one component of an object at a time (Didday and Arbib, 1975), and cumulating the recognition results in the object schema hierarchy.

4 IMPLEMENTATION IN VISOR

VISOR (Leow, 1994; Leow and Miikkulainen, 1994) is a concrete implementation of the above three ideas in the domain of object recognition and scene analysis, and one of the few explicitly schema-based neural network vision systems built to date (see also Arbib, 1989; Feldman, 1985; Hummel and Biederman, 1992). It consists of three main modules: the Low-Level Visual Module, the Schema Module, and the Response Module. The Low-Level Visual Module (LLVM, currently simulated procedurally) focuses attention at one component of an object, such as the triangular roof of the arch, at a time, extracts its shape (what) and relative position (where), and sends this information to the Schema Module. The schema that best matches the input suggests shifting attention to a location where another component, such as the left pillar, is expected. The LLVM then shifts attention to the new position, and the process repeats until VISOR has looked at all the components in the scene.

In the Schema Module, schema representations are organized in two levels (Figure 1). The topmost level consists of scene schemas that receive inputs from lower-level object schemas. Object schemas in turn receive inputs from shape units that represent rough categories of shape and size modeled after Biederman's theory. The spatial structure of an object, such as

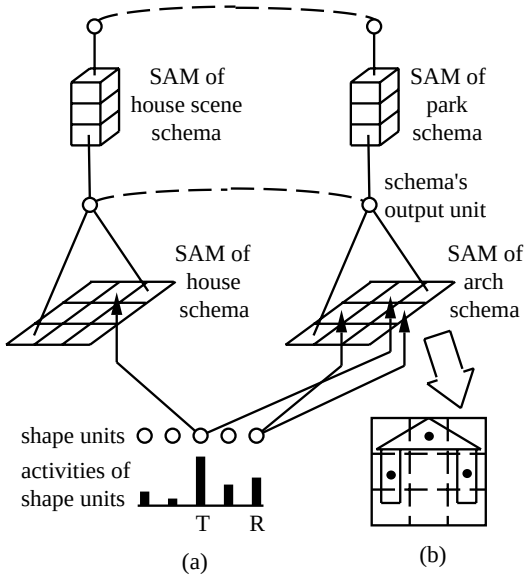

Figure 1: **The Schema Module of VISOR.** (a) The schemas are organized into two levels: objects and scenes. Arrows represent one-way connections from low-level inputs, the solid lines represent both the bottom- up and top-down connections (which are different) in the schema hierarchy, and dashed lines indicate inhibition. The shape unit marked "T" is sensitive to flat triangles, and the one marked "R" to vertical rectangles. (b) The arch image encoded by the arch schema. The grid represents the Sub-schema Activity Map (SAM). The black dots denote those SAM units that represent the arch components.

an arch (Figure 1b), is represented in a 2-D array of units called the Sub-schema Activity Map (SAM). Each component of the object is represented by a SAM unit at the corresponding position. The connections between the shape units and the SAM units encode what VISOR expects to find at each SAM position. For example, the arch, which consists of a triangle on top of two rectangles, is encoded in a 3×3 SAM: the top-center unit is strongly connected to the triangle-sensitive shape unit, and the two units in either side are most strongly activated by the same rectangle-sensitive shape unit.

At the scene level, spatial structure is often less rigid. For example, in a park scene, there may be several instances of the tree, and they may appear anywhere in the scene. The schema for such a scene consists of a column of SAM units without spatial relationships (Figure 1). The connection from an object schema to a SAM unit indicates that several instances of the object may appear anywhere in the scene. A rough spatial structure, such as objects appearing at the top, center, or bottom sections of the scene can be represented by multiple spatially organized columns.

In analyzing a park scene that contains e.g. an arch surrounded by two trees, VISOR

may begin by focusing at the center of the image, that is, at the triangular roof of the arch. The LLVM finds this component to be located at the top-center position of the object and enables the object schemas' top-center SAM units. It also activates the triangle-sensitive shape unit (Figure 1a). Because the arch schema expects a triangle at that relative position, its top-center SAM unit becomes highly activated. As VISOR looks at other components in the scene, the corresponding SAM units' activities are updated, and indicate how likely the other components of the schema are to be present in the scene. The schema's output unit sums up the component activities and indicates how well the entire schema matches the input. In other words, the components cooperate in supporting the schema activation. The output unit then sends activation to the SAM units of higher-level schemas, indicating for example that finding an arch in the scene suggests that the entire scene might depict a park.

Different schemas may share identical or similar parts. For instance, the roof of an arch may look like that of a house. In this case, the triangle-sensitive shape unit has a strong connection to SAM units in both the arch and the house schemas (figure 1.). If the triangles appear in the same relative position, as is the case with the arch and house, then the activation of the triangle- sensitive unit propagates to both arch and house SAMs. This way, whenever VISOR focuses at a new location, all schemas that match the input at that location are simultaneously activated. VISOR keeps shifting attention to other positions and accumulating activation in its schema hierarchy until it has seen all of the important inputs in the scene. In the arch example, the arch schema eventually develops a larger output activity because it matches the input object better than the house schema. It also inhibits the house schema through inhibitory connections between their output units, thus enhancing the difference. Thus, the schemas compete to determine which one best matches the focused object.

VISOR can also learn to encode schemas from examples. Initially all SAM weights have small positive random values, indicating no specific spatial structure. Given an input object such as a house, the schemas become randomly activated in the scene analysis process. The schema that happens to match the input best modifies its weights through variations of Hebbian rules, and gradually learns to encode the spatial structure of the house. At the same time, the Response Module learns to associate the current schema hierarchy activation with the house label provided by the environment. In case that best matching schema happens to already encode a different object, such as the arch, the Response Module will produce an incorrect label. The environment delivers a punishment signal which suppresses the activation of the arch schema, and another schema will become most active. If it is a new schema, no incorrect label will be produced, and learning proceeds as above. This punishment signal is analogous to the mismatch-reset signal in the ART network (Carpenter and Grossberg, 1987). It tells VISOR to find a different schema to encode the house without specifying which one.

5 DISCUSSION

The VISOR implementation of visual schema representation, application, and learning can give a computational account to several phenomena in human object recognition and scene analysis (Leow and Miikkulainen, 1994), but it is still a long way from capturing the full variety and complexity of real world object recognition and scene analysis. Some objects have flexible or movable components that can appear in different spatial relations with each other, such as the limbs and body of a human reaching up or picking up something from the ground. For such objects, topological relationships such as "connected-to" would be more appropriate than rigid spatial relationships (Biederman, 1987). Many objects would need to be represented in 3-D rather than as 2-D projections, and it should be possible to recognize them from different viewpoints and also in different scales and orientations (Hummel and Biederman, 1992; Leow, 1994; Olshausen et al., 1993). Also, segmentation of scenes to their components and separation from the background is perhaps not possible strictly bottom-up as VISOR currently assumes, especially when the objects can be occluded. So far, it has been possible to give only partial answers to some of these questions and others remain wide open.

REFERENCES

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

Biederman, I., 1987, Recognition-by-components: A theory of human image understanding, Psychological Review, 94:115-147.

Carpenter, G.A. and Grossberg, S., 1987, A massively parallel architecture for a self-organizing neural pattern recognition machine, Computer Vision, Graphics, and Image Processing, 37:54-115.

Didday, R.L. and Arbib, M.A., 1975, Eye movements and visual perception: A "two visual system model," Int. J. Man-Machine Studies, 7:547-569.

Draper, B.A., Collins, R.T., Brolio, J., Hanson, A.R., and Riseman, E.M., 1989, The schema system, Int. J. Computer Vision, 2:209-250.

Feldman, J.A., 1985, Four frames suffice: A provisional model of vision and space, Behavioral and Brain Sciences, 8:265-313.

Hanson, A.R. and Riseman, E.M., 1978, VISIONS: A computer system for interpreting scenes, in Computer Vision Systems, (A.R. Hanson and E.M. Riseman, Eds.), New York: Academic Press.

Hinton, G.E., 1988, Representing part-whole hierarchies in connectionist networks, in Proc. 10th Annual Conf. Cog. Sci. Soc., 48-54.

Hummel, J.E. and Biederman, I., 1992. dynamic bindings in a neural network for shape recognition, Psychological Review, 99:480-517

Leow, W.K. and Miikkulainen, R., 1994, Priming, perceptual reversal, and circular reaction in a neural network model of schema-based vision, in Proc. 12th Annual Conf. Cog. Sci. Soc.

Leow, W.K., 1994, VISOR: A Neural Network System That Learns to Represent Schemas for Object Recognition and Scene Analysis, Ph.D. dissertation, Dept. of Computer Sciences, The Univ. of Texas at Austin, May 1994.

*Marr, D., 1982, Vision, San Francisco:Freeman.

Olshausen, B.A., Anderson, C.H., and Van Essen, D.C., 1993, A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information, Neuroscience, 13:4700-4719.

Rumelhart, D.E., 1980, Schemata: The building blocks of cognition, in Theoretical Issues in Reading Comprehension, (R.J. Spiro, B.C. Bruce, and W.F. Brewer, Eds.), New York: Wiley.

Rumelhart, D.E., Smolensky, P., McClelland, J.L., and Hinton, G.E., 1986, Schemata and sequential thought processing in PDP models, in Parallel Distributed Processing, (J.L. McClelland and D.E. Rumelhart, Eds.), Cambridge, Massachusetts: MIT Press.