Neural Networks for Data Compression and Invariant Image Recognition

Sheldon Gardner
Naval Research Laboratory
Washington, DC

SUMMARY

An approach to invariant image recognition \([I^2R]\), based upon a model of biological vision in the mammalian visual system \([MVS]\), is described. The complete \([I^2R]\) model incorporates several biologically inspired features: exponential mapping of retinal images, Gabor spatial filtering, and a neural network associative memory. In the \([I^2R]\) model, exponentially mapped retinal images are filtered by a hierarchical set of Gabor spatial filters \([GSF]\) which provide compression of the information contained within a pixel-based image. A neural network associative memory \([AM]\) is used to process the GSF coded images. We describe a 1-D shape function method for coding of scale and rotationally invariant shape information. This method reduces image shape information to a periodic waveform suitable for coding as an input vector to a neural network AM. The shape function method is suitable for near term applications on conventional computing architectures equipped with VLSI FFT chips to provide a rapid image search capability.

INTRODUCTION

Neural networks offer a potential for technology innovation to provide the next generation of on-board processing \([OBP]\) capability in space-based systems for strategic defense and surveillance as well as other non-military space applications such as remote sensing of the environment. The data collection capabilities of space-based imaging sensors are expected to continue to improve dramatically, further outstripping the ability of operators to exploit image data in real time. One of the goals of the Image Processing Research \([IPR]\) Program at the NRL Naval Center for Space Technology is to develop applications for neural network-based invariant image recognition \([I^2R]\)\([1-4]\).

The encoding of images by the mammalian visual system \([MVS]\) is a subject which has challenged vision researchers for centuries. In the past several years significant progress has been made by Daugman and others towards an understanding of how images are processed within the MVS \([5-12]\). The basic architecture for invariant image recognition is shown in Figure 1. We assume that the MVS performs a sequence of space and space-time mappings which we call scale-space transformations \([SST]\) \([1,2]\). The first SST to occur in the MVS is a logarithmic spatial mapping which occurs in the retina in the vicinity of the fovea. This
mapping, which we call the LZ-SST, produces scale and rotational
invariance in the foveal image [14,15]. A second SST, which we call the
cortical filter SST, or CF-SST, occurs throughout the lateral
geniculate nucleus and the striate cortex. The function of the CF-SST
is to provide a coded representation of the image for associative
memory processing which takes place in higher cortical areas. We have
suggested that, among other operations, the CF-SST includes a
hierarchical network of Gabor filters to map the retinal image into a
four-dimensional function of two spatial variables and two spatial-
frequency variables. Functionally, this mapping is equivalent to
computation of the 4-D Cross-Wigner Distribution [CWD][1,12,13]. These
complex spatial filtering operations occur within the the second block
shown at the top of Figure 1. The encoded image features are then
processed by the neural network associative memory [AM] as shown in
the third block of Figure 1.

In the next section we describe the shape function method for
coding of scale and rotationally invariant shape information into a
scalar waveform. This method can reduce line object shape information
to a scalar waveform suitable for processing by a VLSI FFT array or for
coding as an input vector to a neural network AM.

CODING OF SHAPE FUNCTIONS

Motivated by the properties of the MVS, we can represent a static
image by means of a hierarchical relational graph [HRG][4]. At each
level of the hierarchy, we constructed a set of nodes (simple objects),
and a relational graph (complex object) based upon the relations
between the nodes. At the next lowest level in the hierarchy (finer
resolution), each node is treated as a complex object, composed of its
own set of connected simple objects. Although, we describe the HRG
structure in a top-down manner, in the MVS data flow actually takes
place in a bottom-up manner, since image information is first processed
in the visual cortex, then sent to higher areas of the brain, such as
the cerebral cortex. Recognition of a face can be used as a simple
example of this process. Starting with the placement of features (e.g.
eyes, nose, etc.) we recognize a face as a complex object composed of
simple objects (features). On the next hierarchical level we examine
individual facial features. Fig. 2 illustrates the hierarchical
representation of object shape. The complex object F1[·], shown in
Figure 2, can be represented in terms of a three-level hierarchical
notation F1[G1[H1], G2[H2]].

Figure 3 illustrates a two-step process which can be used to
obtain the shape features of a broad-band multi-level image. The
nonlinear trace operation shown in Figure 3(b) converts a bit-mapped
image into a set of objects. An example of this type of trace operation
can be found in commercial microcomputer software (e.g. Digital
Darkroom®).

Shape information can be used in the construction of object
features vectors useful for object recognition. We illustrate how,
after posterization and tracing between fixed grey levels, shape
information can be coded into a scalar shape function which characterizes a line object. For high speed applications which require special purpose hardware, such as VLSI array processors implementing FFT algorithms, these shape functions can be processed with conventional computers (e.g. a Hypercube® or a Connection Machine®). In the future, when massively parallel neural network computers become available, shape functions can be coded into feature vectors for input to a neural network AM.

As an illustration of the shape function process, an aircraft line object is shown in Figure 4(a) together with the corresponding shape function shown in Figure 4(b). To compute the shape function, we first select a suitable centroid within the object boundary. The shape function is then defined as the distance from this centroid to the object contour measured as a function of distance around the object perimeter. Figure 4(b) is a plot of the aircraft shape function measured from the nose (top). Individual features, such as the engines, can be clearly identified. Figures 5 and 6 show line objects and shape functions for two other aircraft of different types. Figures 7 and 8 show the data for two of the aircraft with a 10 db S/N. The identifying features of each aircraft are still clearly visible in the shape functions. In practice, a sequence of noisy images will usually be available for processing. If the spatial noise background between images in the sequence is uncorrelated, an improvement in S/N will occur when averaging over multiple frames.

CONCLUDING REMARKS

A model for invariant image recognition, based on the properties of the MVS, has been described. The model includes a hierarchical representation of shape information for complex objects. Each level in the hierarchy is represented by a collection of line objects. Through a nonlinear tracing operation the pixel image of each objects is converted to a shape contour. This contour is then represented by a scalar shape function defined as the distance from a centroid within the object to the contour expressed as a function of distance around the object perimeter. This scalar shape waveform uniquely represents object features and can be processed with conventional FFT hardware. Simulations are used to demonstrate the viability of the approach.

REFERENCES


Figure 1. Architecture for invariant image recognition.

Figure 2. A hierarchical representation for object shape
Figure 3. Steps in obtaining shape features from a broad-band image.
Figure 4(a) Aircraft 1 line object

Figure 4(b) Aircraft 1 shape function.
Figure 5(a) Aircraft 2 line object

Figure 5(b) Aircraft 2 shape function.
Figure 6(a) Aircraft 3 line object

Figure 6(b) Aircraft 3 shape function.
Figure 7(a) Aircraft 1 line object (10 db S/N)

Figure 7(b) Aircraft 1 shape function (10 db S/N)
Figure 8(a) Aircraft 2 line object (10 db S/N)

Figure 8(b) Aircraft 2 shape function (10 db S/N)