Dr. R. S. Michalski

88-22

2-D Invariant Object Recognition Using Distributed Associative Memory

HARRY WECHSLER, SENIOR MEMBER, IEEE, AND GEORGE LEE ZIMMERMAN, STUDENT MEMBER, IEEE

Abstract—This paper describes an approach to two-dimensional object recognition. Complex-log conformal mapping is combined with a distributed associative memory to create a system which recognizes objects regardless of changes in rotation or scale. Recalled information from the memorized database is used to classify an object, reconstruct the memorized version of the object, and estimate the magnitude of changes in scale or rotation. The system response is resistant to moderate amounts of noise and occlusion. Several experiments, using real, gray scale images, are presented to show the feasibility of our approach.

Index Terms—Complex-log mapping, distributed associative memory, invariance, pattern recognition, space variant filtering.

I. Introduction

THE challenge of the visual recognition problem stems from the fact that the projection of an object onto an image can be confounded by several dimensions of variability such as uncertain perspective, changing orientation and scale, sensor noise, occlusion, and nonuniform illumination. A vision system must not only be able to sense the identity of an object despite this variability, but must also be able to characterize such variability—because the variability inherently carries much of the valuable information about the world. For example, assume that a computer vision system receives a series of motion images of a ship from a remote sensor. If the image of the ship expands but does not translate in each successive frame then a collision with the remote sensor is iminent. The survival of the remote sensor may depend on the ability of the vision system to both recognize the object and extract how the object is changing in time. Once the variability has been characterized, action can be taken to prevent the collision.

Our goal is to derive the functional characteristics of image representations suitable for invariant recognition using a distributed associative memory. The main question is that of finding appropriate transformations such that interactions between the internal structure of the resulting representations and the distributed associative memory yield invariant recognition. As Simon [1] points out, all mathematical derivation can be viewed simply as a change of representation, making evident what was previously true but obscure. This view can be extended to all problem solving. Solving a problem then means transforming it so as to make the solution transparent.

The seminal work of Marr [2] considers computational vision as an information processing task. He defines three levels at which any machine vision system must be understood. First, the basic computational theory specifies what is the task, why is it appropriate, and what is the strategy by which it can be carried out. Second, the representation and algorithm specify how the computational theory can be implemented in terms of input, output, and transformations. It is apparent that the visual task determines the mixture of representations and algorithms. Third, the hardware specifies the actual implementation.

There are several major ways to handle the issue of image variability. The approaches can be distinguished according to how memorized patterns are matched against input image representations. The interaction may occur along several different dimensions of the representation. There are viewer-centered and object-centered representations. A viewer-centered representation is viewpoint dependent and lacks generality but it might be necessary for navigation tasks. An object-centered representation is a description given in terms of a coordinate system which is attached to the object in space. One example of such an object representation is the generalized cylinders [3]. Representations can also vary along the dimensions of multiprototype versus a canonical representation and complete versus incomplete representation. A canonical representation can characterize the input pattern with a single template. A complete representation encodes sufficient information to allow detection under geometric distortions, noise, and partial occlusion. Our recognition system requires that variability be dealt with by specifying canonical, complete, object-centered representations. The memory component must be able to account for occlusion and noise, partial key indexing and reconstruction. It should also yield the entire output vector even if the input is noisy or partially present. Distributed associative memories [4] provide this capability.

We approach the problem of object recognition with three requirements: classification, reconstruction, and characterization. Classification implies the ability to distinguish objects that were previously encountered. Reconstruction is the process by which memorized images can

Manuscript received November 5, 1986; revised October 27, 1987. Recommended for acceptance by C. Brown. This work was supported in part by the National Science Foundation under Grant ECS-8310057 and by a grant from the Microelectronics and Information Science (MEIS) Center of the University of Minnesota.

H. Wechsler is with the Department of Computer Science, George Mason University, Fairfax, VA 22030.

G. L. Zimmerman is with the Department of Electrical Engineering, University of Minnesota, Minneapolis, MN 55455.

IEEE Log Number 8823855

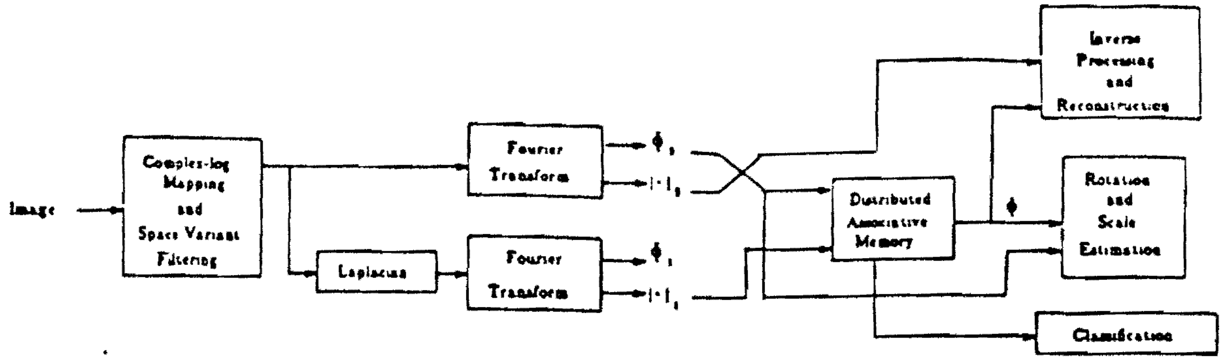


Fig. 1. Block diagram of the system.

be drawn from memory given a distorted version exists at the input. Characterization involves extracting information about how the object has changed from the way in which it was memorized. Our goal in this paper is to discuss a system which is able to recognize memorized two-dimensional objects regardless of geometric distortions like changes in scale and orientation, and can characterize those transformations. The system also allows for noise and occlusion and is tolerant of memory faults.

Sections II and III describe the various components of the system in detail. Section IV presents the results from several experiments we have performed on real data. The paper concludes with a discussion of our results and their implications for future research.

II. Invariant Representation

The goal of this section is to examine the various components used to provide the vectors which are associated in the distributed associative memory.

The block diagram which describes the various functional units involved in obtaining an invariant image representation is shown in Fig. 1. The image is complex-log conformally mapped so that rotation and scale changes become translation in the transform domain. Along with the conformal mapping, the image is also filtered by a space variant filter to reduce the effects of aliasing. The conformally mapped image is then processed through a Laplacian in order to solve some problems associated with the conformal mapping. The Fourier transform of both the conformally mapped image and the Laplacian processed image produce the four output vectors. The magnitude output vector $|\bullet|_1$ is invariant to linear transformations of the object in the input image. The phase output vector Φ_2 contains information concerning the spatial properties of the object in the input image.

A. Complex-Log Mapping and Space Variant Filtering

The fist box of the block diagram given in Fig. 1 consists of two components: complex-log mapping and space variant filtering. Complex-log mapping transforms an image from rectangular coordinates to polar exponential coordinates. This transformation changes rotation and scale into translation. Fig. 2 shows vertical lines and 45 degree lines and their respective complex-log mapped images.

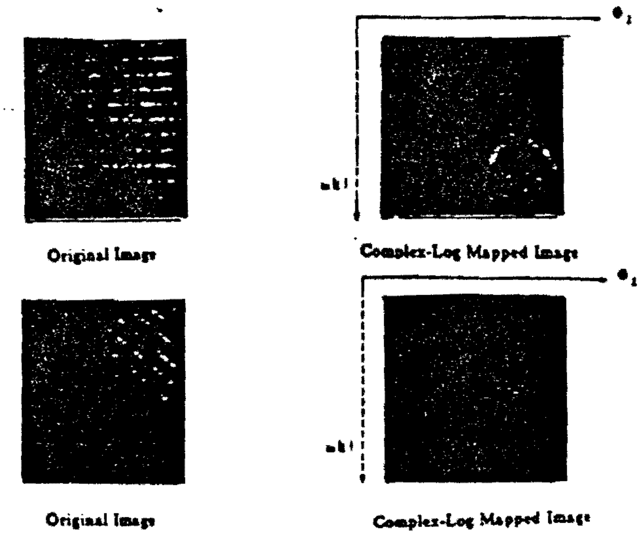


Fig. 2. Rotation in the complex-log mapped domain.

Notice that the rotation in the image space corresponds to a translation along the x-axis in the complex-log space.

Fig. 3 shows an image of concentric white circles and the corresponding complex-log mapped image. Although the distance between the edges of the white circles becomes larger with eccentricity, the distance between the layers of its complex-log mapped image stays the same—scale changes are thus transformed into vertical translation.

The complex-log mapping transforms radial lines into vertical lines and concentric circles into horizontal lines. If the image is mapped into a complex plane then each pixel (x, y) on the Cartesian plane can be described mathematically by $z = x + jy$. The complex-log mapped points w are described by

$$w = \ln(z) = \ln(|z|) + j\theta_z \tag{1}$$

where $|z| = (x^2 + y^2)^{1/2}$ and $\theta_z = \tan^{-1}(y/x)$.

Our system sampled 256×256 pixel images to construct 64×64 complex-log mapped images. Samples were taken along radial lines spaced 5.6 degrees apart. Along each radial line the step size between samples increased by powers of 1.08. These numbers are derived from the number of pixels in the original image and the number of samples in the complex-log mapped image. An excellent examination of the different conditions involved

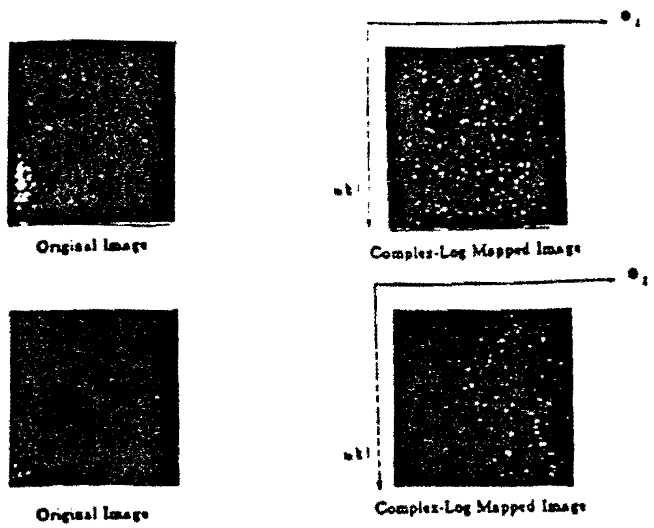


Fig. 3. Scaling in the complex-log mapped domain.

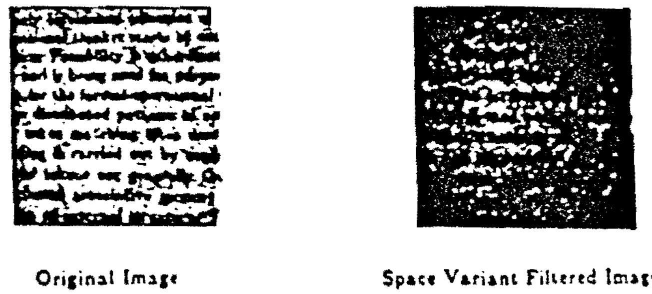


Fig. 4. Space variant filtering.

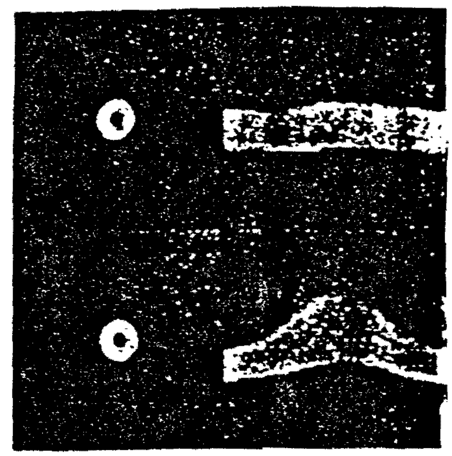


Fig. 5. Center misalignment effects on complex-log mapping.

in selecting the appropriate number of samples for a complex-log mapped image is given in [5]. The nonlinear sampling can be split into two distinct parts along each radial line. Toward the center of the image the samples are dense enough that no antialiasing filter is needed. Samples taken at the edge of the image are large and an antialiasing filter is necessary. The image filtered in this manner has a circular region around the center which corresponds to an area of highest resolution. The size of this region is a function of the number of angular samples and radial samples. An example of such filtering is shown in Fig. 4. Notice in Fig. 4 that the area of highest resolution encircles the word "pattern" and that the image is greatly blurred beyond that region. The filtering is done, at the same time as the sampling, by convolving truncated Bessel functions with the image in the space domain. The width of the Bessel functions main lobe is inversely proportional to the eccentricity of the sample point.

There are several problems associated with the complex-log mapping. *First*, because the system samples from high resolution to low resolution, the image reconstructed from the samples will not carry all the information from the original. In particular, details close to the edge of the original image will be smeared by sampling and reconstruction. This smearing is shown clearly in Figure 4. The size of the objects used in our experiments is small compared to the size of the whole image so that most of the object details fit inside the region of highest resolution.

A *second* problem is sensitivity to center misalignment of the sampled image. Small shifts from the center causes dramatic distortions in the complex-log mapped image. This is shown in Fig. 5.

Our system assumes that the object is centered in the image frame. Slight misalignments are considered noise. Large misalignments are considered as translations and could be accounted for by changing the gaze in such a way as to bring the object into the center of the frame. The decision about what to bring into the center of the frame is an active function and should be determined by the task. An example of a system which could be used to guide the translation process was developed by Anderson *et al.* [6]. Their pyramid system analyzes the input image at different temporal and spatial resolution levels. Their smart sensor was then able to shift its fixation such that interesting parts of the image (i.e., something large and moving) was brought into the central part of the frame for recognition.

A *third* problem that occurs in the complex-log mapping is related to its size invariant aspect—a change in scale does not appear as a direct translation in practice. When an image is scaled from smaller to larger a translation occurs in the complex-log mapped image but the points left vacant by the translation are filled with more samples from the center of the image. If the object in the image has no hole in its center the new samples which take the place of the translating points will in general be very similar to those translating points. This has the effect of stretching, not simple translation in the complex-log mapped image. Fig. 6(a) shows the radial dimension of an object that has a hole in the center and a scaled version of the same object. Fig. 6(b) is the complex-log mapped version of these images. Notice that scaling in the image domain corresponds directly to translation in the complex-log mapped domain. Fig. 6(c) shows an object whose center is not like the background (i.e., no hole) and a scaled version of this object. Fig. 6(d) is the complex-log mapped version of these objects. In this case, expansion in the image domain does not correspond to translation in the complex-log domain, but instead to stretching. The problem is solved by convolving the complex-log mapped image with a Laplacian which sharpens the edges and ze-

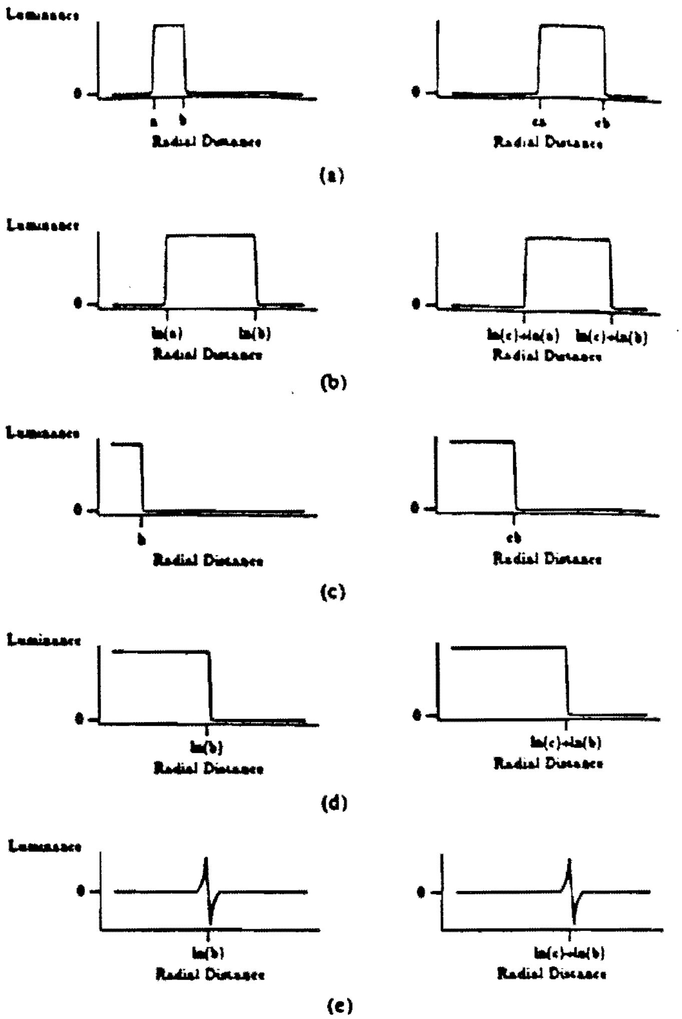


Fig. 6. Translation and stretching. (a) Original images. (b) Complex-log mapped images. (c) Original images—no hole. (d) Complex-log mapped images. (e) After Laplacian.

roes regions that vary slowly as shown in Fig. 6(e). The Laplacian is the derivative of a bandpass filter so high frequency variations due to textured central region of an object will also be smoothed and set to zero. The range of variations which are accentuated are determined by the size of the Laplacian channel. Determining the optimal size of the channel parameter is not addressed in this paper. A description of the Laplacian and its use is discussed in more detail later.

B. Fourier Transform

The second box in the block diagram of Fig. 1 is the Fourier transform. The Fourier transform of a two-dimensional image $f(x, y)$ is given by

$$F(u, v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y) e^{-j(ux + vy)}\, dx\, dy \quad (2)$$

and can be described by two two-dimensional functions corresponding to the magnitude $|F(u, v)|$ and phase $\Phi_F(u, v)$. The magnitude component of the Fourier transform which is invariant to translation, carries much of the contrast information of the image. The phase component of the Fourier transform carries information about how things are placed in an image. Translation of $f(x, y)$ corresponds to the addition of a linear phase component. The complex-log mapping transforms rotation and scale into translation and the magnitude of the Fourier transform is invariant to those translations so that $|\bullet|$ will not change significantly with rotation and scale of the object in the image.

The representation system we have developed is similar to the Mellin transform with a polar transformation of the input data. The magnitude of the polar Mellin transform, which is invariant to rotation and scale, has been used for object recognition in the past [7]. Our system is different from these past systems in several ways. We use space variant filtering to account for the aliasing caused by the nonlinear sampling. Instead of matching using direct correlation of the magnitude of the Mellin transform, we use the magnitude to index the appropriate memorized phase. This allows our system to classify, characterize, and reconstruct the previously memorized vector.

The phase of the Fourier transform holds the spatial layout of the image under examination. Oppenheim and Lim [8] examined the importance of the phase and showed that under fairly loose conditions the entire image could be reconstructed to within a constant multiple of the magnitude given only the phase. This implies that most of the information allowing discrimination between real images lie in the phase. However, Lane *et al.* [9] showed that the intrinsic form of a finite positive image is uniquely related to the magnitude of its Fourier transform, except under contrived conditions or trivial situations. This suggests that reasonable discrimination can still be obtained using the magnitude of the Fourier transform of an image.

C. Laplacian

The Laplacian that we use is a difference-of-Gaussians (DOG) approximation to the $\nabla^2 G$ function as given by Marr [2].

$$\nabla^2 G = \frac{1}{\pi\sigma^4}\left[1 - r^2/2\sigma^2\right] e^{(-r^2/2\sigma^2)} \quad (3)$$

The result of convolving the Laplacian with an image can be viewed as a two step process. The image is blurred by a Gaussian kernel of a specified width σ. Then the isotropic second derivative of the blurred image is computed. The width of the Gaussian kernel is chosen such that the conformally mapped image is visible—approximately 2 pixels in our experiments. The Laplacian sharpens the edges of the object in the image and sets any region that did not change much to zero. Below we describe the benefits from using the Laplacian.

The Laplacian eliminates the stretching problem encountered by the complex-log mapping due to changes in object size. When an object is expanded the complex-log mapped image will translate. The pixels vacated by this translation will be filled with more pixels sampled from the center of the scaled object. These new pixels will not be significantly different than the displaced pixels so the result looks like a stretching in the complex-log mapped image. The Laplacian of the complex-log mapped image will set the new pixels to zero because they do not significantly change from their surrounding pixels. After the complex-log mapped image is processed through the Laplacian, scale changes in the image will correspond directly to translation in the complex-log mapped image.

The second benefit of using the Laplacian is that it is not necessary to window the Fourier transform. The images that we work with are discrete and of finite dimension. The Fourier transform is obtained using an FFT routine which assumes the input is periodic. If, for example, the image being transformed has a left edge which is dark and a right edge which is light, an artifact in the form of an abrupt jump in contrast between the edges will be observed. This will cause a high frequency spreading in the Fourier transform. The complex-log mapped images have this kind of abrupt jump along the radial dimension. Since the Laplacian sets the edges of the complex-log mapped images to zero such frequency spreading is avoided.

Another benefit of using the Laplacian is to enhance the differences between memorized objects. The Laplacian accentuates edges and deemphasizes areas of little change. Since the objects that are being memorized differ mostly in shape, this processing emphasizes these differences.

D. Summary

The end result of applying the different transformations outlined in this section is to produce two vectors from an image: $|\bullet|_1$ which is invariant to geometric changes and Φ_2 which contains information about the position of the object in the image. Access to both of these vectors allows the image to be reconstructed. The magnitude vector $|\bullet|_2$ is used to reconstruct the memorized object. Most of the transforms are completely invertible so little of the useful information has been removed.

III. Distributed Associative Memory (DAM)

The particular form of distributed associative memory that we deal with in this paper is a memory matrix which, like a filter, can modify the flow of information. Stimulus vectors are associated with response vectors and the result of this association is spread over the entire memory space. Distributing in this manner means that information about a small portion of the association can be found in a large area of the memory. New associations are placed over the older ones and are allowed to interact. This means that the size of the memory matrix stays the same regardless of the number of associations that have been memorized.

The above discussion illuminates several properties of distributed associative memories which are different from the more traditional ones about memory. Because the associations are allowed to interact with each other an implicit representation of structural relationships and contextual information can develop, and as a consequence a very rich level of interactions can be captured. Since there are few restrictions on what vectors can be associated there can exist extensive indexing and cross-referencing in the memory. Since the information is distributed, the overall function of the system is resistant to faults in the memory and degraded stimulus vectors. Distributed associative memory captures a distributed representation which is context dependent. This is quite different from the simplistic behavioral model [10].

A. Construction and Recall

The *construction* stage assumes that there are n pairs of m-dimensional vectors that are to be associated by the distributed associative memory. This can be written as

$$M\bar{s}_i = \bar{r}_i \qquad \text{for } i = 1, \cdots, n \tag{4}$$

where $\bar{s}_i$ denotes the ith stimulus vector and $\bar{r}_i$ denotes the ith corresponding response vector. We want to construct a memory matrix M such that when the kth stimulus vector $\bar{s}_k$ is projected onto the space defined by M the resulting projection will be the corresponding response vector $\bar{r}_k$. More specifically we want to solve the following equation:

$$MS = R \tag{5}$$

where $S = [\bar{s}_1 | \bar{s}_2 | \cdots | \bar{s}_n]$ and $R = [\bar{r}_1 | \bar{r}_2 | \cdots | \bar{r}_n]$. A unique solution for this equation does not necessarily exist for any arbitrary group of associations that might be chosen. Usually, the number of associations n is smaller than m, the length of the vector to be associated, so the system of equations is underconstrained. The constraint used to solve for a unique matrix M is that of minimizing the square error, $\| MS - R \|^2$, which results in the solution

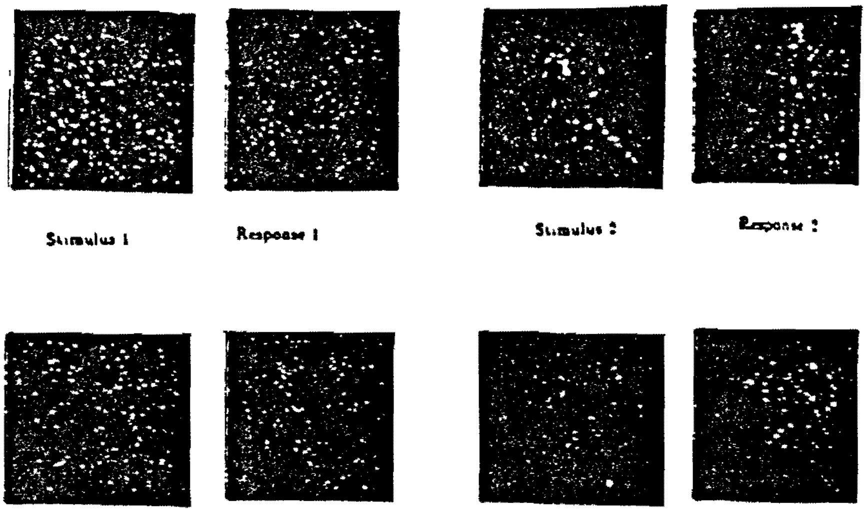
$$M = RS^+ \tag{6}$$

where S^+ is known as the Moore–Penrose generalized inverse of S [4].

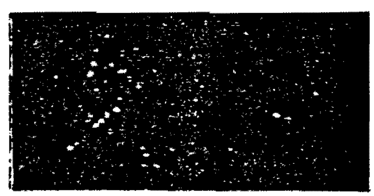
The *recall* operation projects an unknown stimulus vector $\bar{s}$ onto the memory space M. The resulting projection yields the response vector $\bar{r}$

$$\bar{r} = M\bar{s}. \tag{7}$$

If the memorized stimulus vectors are independent and the unknown stimulus vector $\bar{s}$ is one of the memorized vectors $\bar{s}_k$, then the recalled vector will be the associated response vector $\bar{r}_k$. If the memorized stimulus vectors are dependent, then the vector recalled by one of the memorized stimulus vectors will contain the associated response vector and some *crosstalk* from the other stored response vectors. Fig. 7 shows the result of a recall from a memory. The vector associations for this example are shown in Fig. 7(a). Notice that the first two stimulus vectors are combined to make up the last stimulus vector. If there is no crosstalk between the vectors in the memory we would expect the recall to be similar to the last response vector. The actual recall in this case is shown in Fig. 7(b). The recall is a combination of the first two response vectors



(a)



(b)

Fig. 7. Crosstalk in the recalled vector from an autoassociative memory. (a) Database. (b) Recall.

instead of the last response vector. The resulting noise or crosstalk in the output is due to the similarity of the memorized vectors.

The recall can be viewed as the weighted sum of the response vectors. The recall begins by assigning weights according to how well the unknown stimulus vector matches with the memorized stimulus vector using a linear squares classifier. The response vectors are multiplied by the weights and summed together to build the recalled response vector. The recalled response vector is usually dominated by the memorized response vector that is closest to the unknown stimulus vector. The distributed associative memory will have interactions between the different associations and this allows some generalization of responses to previously unknown stimulus.

Assume that there are n associations in the memory and each of the associated stimulus and response vectors have m elements. This means that the memory matrix has m^2 elements. Also assume that the noise that is added to each element of a memorized stimulus vector is independent, zero mean, with a variance of σ_i^2. The recall from the memory is then

$$\vec{r} = \vec{r}_k + \vec{v}_0 = M(\vec{s}_k + \vec{v}_i) = \vec{r}_k + M\vec{v}_i \quad (8)$$

where $\vec{v}_i$ is the input noise vector and $\vec{v}_0$ is the output noise vector. The ratio of the average output noise variance to the average input noise variance is

$$\sigma_0^2/\sigma_i^2 = \frac{1}{m} Tr[MM^T]. \quad (9)$$

For the autoassociative case this simplifies to

$$\sigma_0^2/\sigma_i^2 = \frac{n}{m}. \quad (10)$$

This says that when a noisy version of a memorized input vector is applied to the memory the recall is improved by a factor corresponding to the ratio of the number of memorized vectors to the number of elements in the vectors. For the heteroassociative memory matrix a similar formula holds as long as n is less than m [11].

$$\sigma_0^2/\sigma_i^2 = \frac{1}{m} Tr[RR^T]\, Tr[(S^TS)^{-1}] \quad (11)$$

Another way of viewing this error correcting process is to notice that the memory matrix is the orthogonal projection matrix for the set of stimulus vectors. The noise vector in this m-dimensional space will be projected onto the space spanned by the n memorized vectors. The parts of the noise vector that are orthogonal to the n memorized stimulus vectors will be lost and this accounts for the noise reduction in the output recall vector.

Fault tolerance is a byproduct of the distributed nature and error correcting capabilities of the distributed associative memory. By distributing the information, no single memory cell carries a significant portion of the information critical to the overall performance of the memory.

IV. Experiments

In this section we discuss the result of computer simulations of our system. The computer simulations occur in three phases: construction, recall, and recognition. In the construction phase, associations to be memorized are used to construct the memory matrix. In the recall phase, an unknown image is processed and then projected onto the memory matrix to produce a recalled vector. In the recognition phase, the recalled vector is used to reconstruct, classify, and characterize the unknown object.

Images of objects are first preprocessed through the subsystem outlined in Section II. The output of such a subsystem is four vectors: $|\bullet|_1$, Φ_1, $|\bullet|_2$, and Φ_2. We construct the memory by associating the stimulus vector $|\bullet|_1$ with the response vector Φ_2 for each object in the database. To perform a recall from the memory the unknown image is preprocessed by the same subsystem to produce the vectors $|\tilde{\bullet}|_1$, $\tilde{\Phi}_1$, $|\tilde{\bullet}|_2$, and $\tilde{\Phi}_2$. The resulting stimulus vector $|\tilde{\bullet}|_1$ is projected onto the memory matrix to produce a response vector which is an estimate of the memorized phase $\hat{\Phi}_2$. The estimated phase vector $\hat{\Phi}_2$ and the magnitude $|\tilde{\bullet}|_1$ are used to reconstruct the memorized object. The difference between the estimated phase $\hat{\Phi}_2$ and the unknown phase $\tilde{\Phi}_2$ is used to estimate the amount of rotation and scale experienced by the object.

The database of images consists of twelve objects: four keys, four mechanical parts, and four leaves. The objects were chosen for their essentially two-dimensional structure. Each object was photographed using a digitizing video camera against a black background. We emphasize that all of the images used in creating and testing the recognition system were taken at different times using various camera rotations and distances. The images are digitized to 256 × 256, eight bit quantized pixels, and each object covers an area of about 40 × 40 pixels. This small object size relative to the background is necessary due to the nonlinear sampling of the complex-log mapping discussed in Section II. The objects were centered within the frame by hand. This is the source of much of the noise and could have been done automatically using the object's center of mass or some other criteria determined by the task. The orientation of each memorized object was arbitrarily chosen such that their major axis was vertical. The two-dimensional images that are the output from the invariant representation subsystem are scanned horizontally to form the vectors for memorization. The database used for these experiments is shown in Fig. 8.

The first example of the operation of our system is shown in Fig. 9. In the upper left quadrant is the image of one of the leaves as it was memorized. In the upper right quadrant is the unknown object presented to our system. The unknown object in this case is the same leaf that has been rotated by 90 degrees. In the lower left quadrant is the recalled, reconstructed image. The rounded edges of the recalled image are artifacts of the complex-log mapping. Notice that the reconstructed recall is the unrotated memorized leaf with some noise caused by errors in the recalled phase. The lower right quadrant is a histogram which graphically displays the classification vector which corresponds to $S^{-}s$. The histogram shows the interplay between the memorized images and the unknown image. The "11" on the bargraph indicates which of the twelve classes the unknown object belongs.

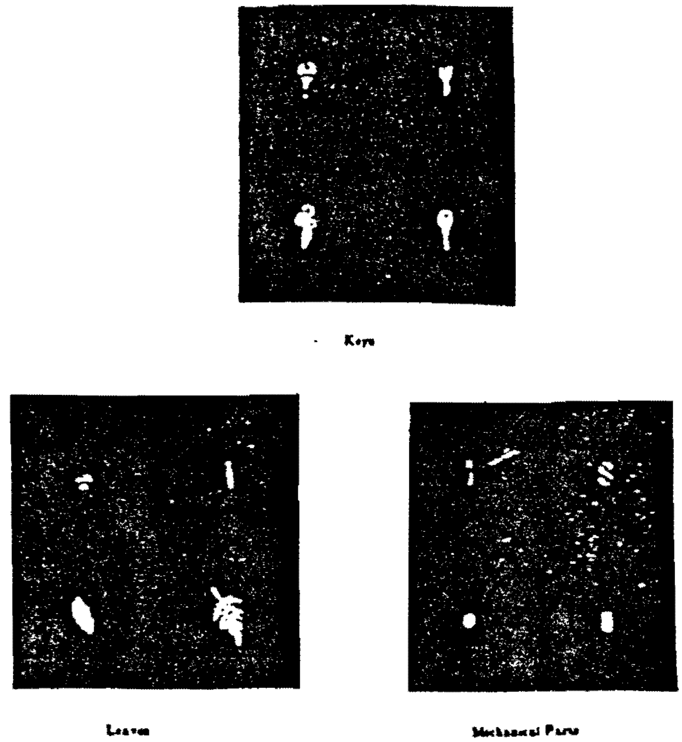


Fig. 8. The database of objects used in the experiments.

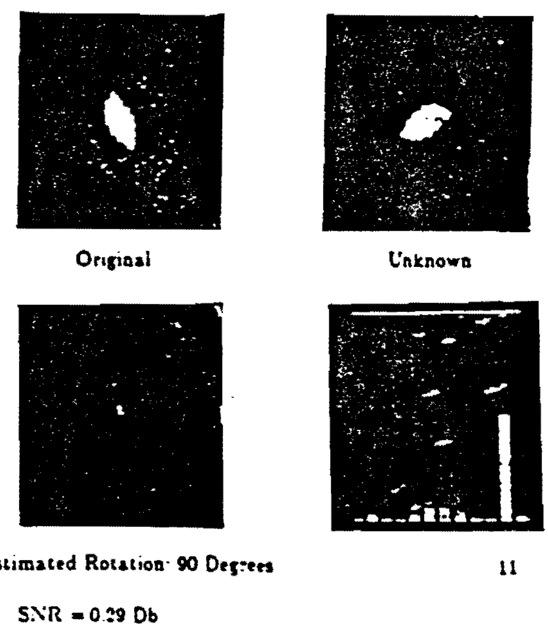


Fig. 9. Recall using a rotated leaf.

The histogram gives a value which is the best linear estimate of the image relative to the memorized objects. Another measure, the signal-to-noise ratio (SNR), is given at the bottom of the recalled image. SNR compares the variance of the ideal recall after processing with the variance of the difference between the ideal and actual recall. This is a measure of the amount of noise in the recall. The SNR does not carry much information about the quality of the recall image because the noise measured by the SNR is due to many factors such as misalignment of the center, changing reflections, and dependence between other memorized objects—each affecting quality in a variety of ways.

Rotation and scale estimates are made using a vector $\bar{D}$ corresponding to the difference between the unknown vector Φ_2 and the recalled vector $\hat{\Phi}_2$. In an ideal situation $\bar{D}$ will be a plane whose gradient indicates the exact amount of rotation and scale the recalled object has experienced. In our system the recalled vector $\hat{\Phi}_2$ is corrupted with noise which means rotation and scale have to be estimated. The estimate is made by letting the first order difference $\bar{D}$ at each point in the plane vote for a specified range of rotation or scale. The estimate is the range which receives the most votes. For example, rotation will have a first order difference of $\bar{D}$ in the horizontal direction that lies between -180 and 180 degrees. If the first order difference is between -22.5 and 22.5 degrees then a vote is added to the no shift range. If it lies between 22.5 and 67.5 degrees then a vote is added to the 45 degree range, and so on.

We show only the estimate of the rotation of the object and not an estimate of the scale because of the coarseness of the method. It works well for estimation of the amount of rotation because rotation in the image corresponds to relatively large translations in the complex-log mapped image. This is not the case for scale. The images used in our simulation can be perceptively larger in the image domain but the differences in the complex-log domain are not very great. The unknown object in Fig. 10 is a memorized key that has been expanded. The reconstructed recall is a key which is the same size and shape as the memorized key. At the bottom of Fig. 10 is the complex-log mapped version of the memorized key and the scaled key. Notice that the difference along the scale axis is not very great which makes estimating the size change very difficult.

Fig. 11 shows the recall when the unknown is a key which is both rotated and scaled. The reconstructed image is not rotated or scaled relative to the way it was memorized. There is an error in the estimate of rotation on this example. The unknown key is rotated 180 degrees and the estimate is -135 degrees. This error is due to noise in the $\bar{D}$ vector. The estimate is actually off only by one adjoining bin and the difference between the number of votes between the real rotation and the estimate is 12 out of 600.

Fig. 12 is an example of occlusion. The unknown object in this case is an "S" curve which is larger and

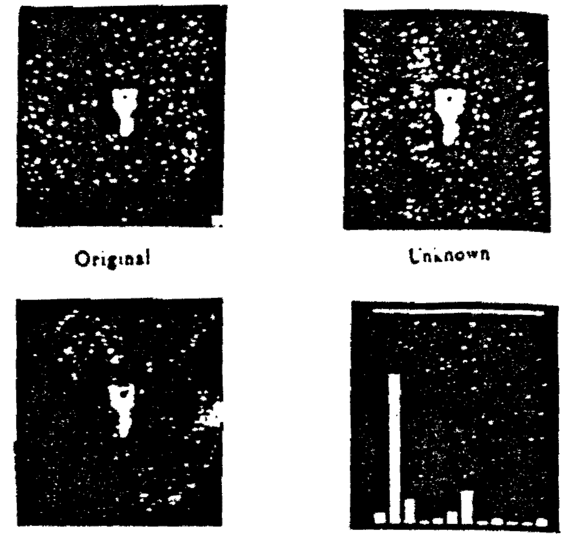


Fig. 10. Recall using scaled key.

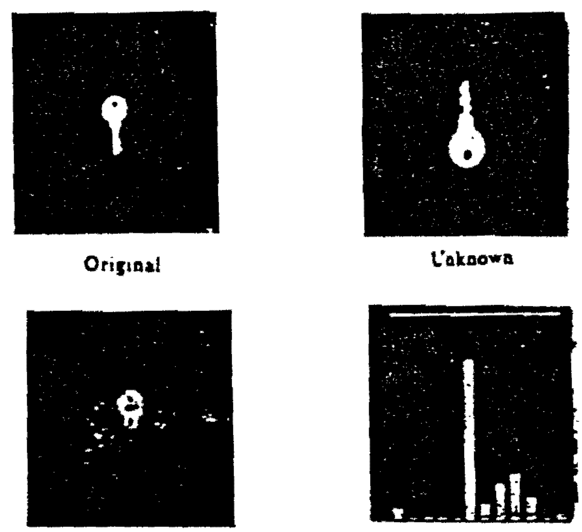


Fig. 11. Recall using scaled and rotated key.

of the bottom curve was occluded. The resulting reconstruction is very noisy but has filled in the missing part of the bottom curve. The noisy recall is reflected in both the SNR and the interplay between the memories shown by the histogram.

Fig. 13 displays the result of *locally* setting a fraction of the memory matrix elements to zero. The damage done locally in the memory matrix is present in a local sense in the recall. In the upper left quadrant is the ideal reconstructed recall with no damage to the memory matrix. In the upper right quadrant is the recall when 30 percent of the memory matrix is set to zero. In the lower left quadrant is the recall for 50 percent and in the lower right quadrant is the recall for 75 percent. When 75 percent of the memory matrix is locally set to zero the

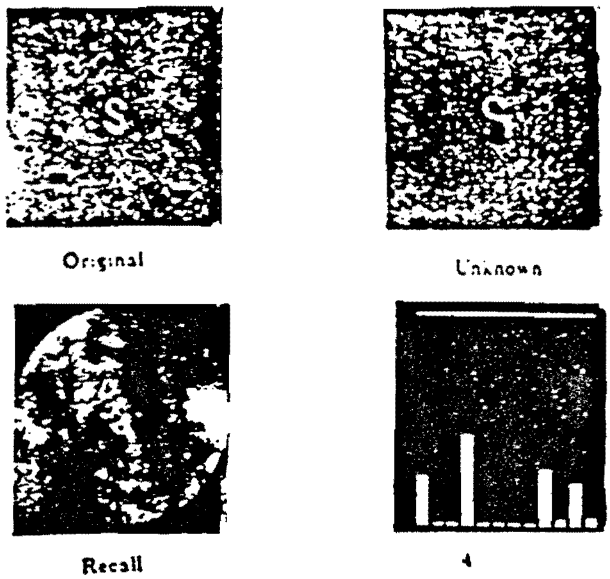


Fig. 12. Recall using scaled and rotated "S" with occlusion.

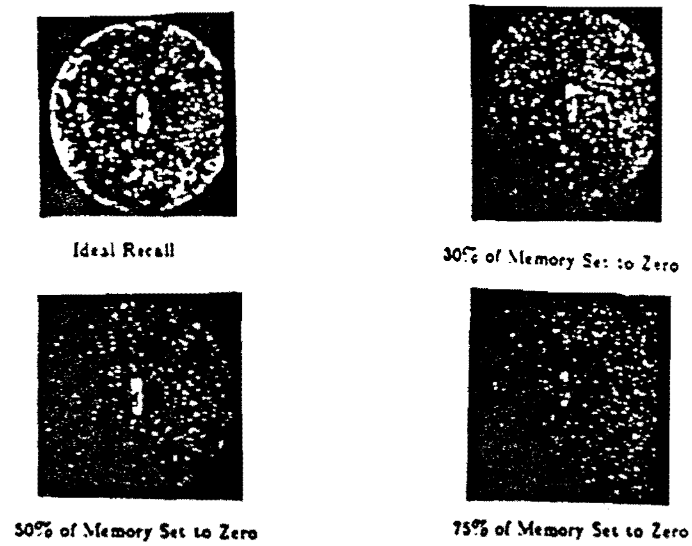


Fig. 14. Recall for memory matrix randomly set to zero.

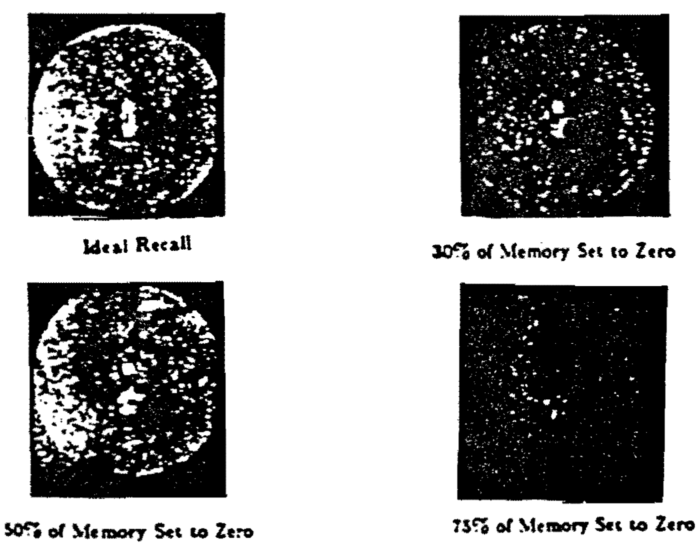


Fig. 13. Recall for memory matrix locally set to zero.

Fig. 14 is the result of *randomly* setting the elements of the memory matrix to zero. The effect of this kind of damage is not nearly as critical as in the case of the local damage. The upper left quadrant shows is the ideal recall. In the upper right quadrant is the recall after 30 percent of the memory matrix has been set to zero. In the lower left quadrant is the recall for 50 percent and in the lower right is the recall for 75 percent. Even when 90 percent of the memory matrix has been set to zero a faint outline of the pin could still be seen in the recall. This result is important in two ways. First, it shows that the distributed associative memory is robust in the presence of noise. Second, it shows that a completely connected network is not necessary and as a consequence a scheme for data compression of the memory matrix could be found.

V. Conclusion

In this paper we demonstrate a computer vision system which recognizes two-dimensional objects invariant to rotation or scale. The system combines an invariant representation of the input images with a distributed associative memory such that objects can be classified, reconstructed, and characterized. The distributed associative memory is resistant to moderate amounts of noise and occlusion. Several experiments, demonstrating the ability of our computer vision system to operate on real, gray scale images, were presented.

There are some similarities between the computer vision system that we present and the transformations that may take place in early biological vision. We do not suggest that our computer vision system is anything more than a very rough first-order approximation to the diverse and complex biological system but we feel it is important to understand the strengths and weaknesses of the system within this context.

One of the fundamental assumptions of our system is that the object or a feature of the object can be centered in the frame. We do not take translation of the object into account. Instead we suggest this centering can be done by a change in the viewpoint, which is not completely uncharacteristic of the biological vision system's approach to translation. Although there are many studies which show that humans have the ability to recognize patterns which are not centered on the fovea, for normal recognition tasks such as reading, humans do bring the object under examination to the center of their view [12].

The complex-log mapping has been proposed previously as a model of the projection of the retina onto visual area 17 of the cat [13]. Other evidence such as size constancy and the cortical magnification factor strongly suggest that scale invariant recognition is at work in biological vision. Our system required a space variant filter to reduce the effects of aliasing caused by sampling the image nonlinearly. This process is similar to the kind of summation found across the retina. Another requirement of our system from the standpoint of signal processing is the need to use a Laplacian after the complex-log mapping. The application of the isotropic Laplacian of a single channel size to the complex-log mapped image is the same as convolving the image with a Laplacian whose channel size increases with eccentricity—similar to the spatial frequency channel mechanisms in human vision proposed by Wilson and Bergen [14]. Complex-log mapping is only a first order approximation to the early visual processes and is not intended to account for a multimode of phenomena such as orientation selective cells present in the cortex [15].

Although the existence of spatial frequency channels in the biological vision system is well established [16], there is no evidence that the global Fourier transform is performed anywhere in the cortex. The magnitude of the Fourier transform is used in our computer vision system to index the distributed associative memory primarily because it is invariant to translation of the input signal. There are other classes of shift invariant transforms, such as C-transforms [17], which can be executed by networks of simple threshold logic units—more consistent with the type of processing of which neurons are capable. The phase of the Fourier transform is used to reconstruct the memorized object and estimate corresponding scale and rotation changes. The reconstruction and estimation can be used by other systems to accomplish a desired task. If scale or rotation are necessary for the task, then the concept of indexing with an invariant pattern to gain relative information about change in the input is not an altogether unlikely model for what might occur in early biological vision.

Neural network models, of which the distributed associative memory is one example, were originally developed to simulate biological memory. They are characterized by a large number of highly interconnected simple processors which operate in parallel. An excellent review of the many neural network models is given in [18]. The distributed associative memory we use is linear, and as a result there are certain desirable properties which will not be exhibited by our computer vision system. For example, feedback through our system will not improve recall from the memory. Recall could be improved if a non-linear element, such as a sigmoid function, is introduced into the feedback loop. Nonlinear neural networks, such as those proposed by Hopfield [19] or Anderson *et al.* [20], can achieve this type of improvement because each memorized pattern is associated with stable points in an energy space. The price to be paid for the introduction of nonlinearities into a memory system is that the system will be difficult to analyze and can be unstable. Implementing our computer vision system using nonlinear distributed associative memory is a goal of our future research.

Each component of our computer vision system can be implemented in parallel. Messner and Szu [21] described a parallel architecture which can produce the complex-log mapping of an image. There exist many parallel algorithms for implementing discrete Fourier transforms and matrix multiplications. Another approach is to implement the different functions of the system optically. Case *et al.* [22] designed holographic lenses to perform mathematical transformations such as the complex-log mapping of an image. The Fourier transform of an image is easily accomplished using a lens. The distributed associative memory and holograms have many similarities but it is not immediately apparent how this part of our system could be implemented optically.

We are presently extending our work toward three-dimensional object recognition. Much of the present research in three-dimensional object recognition is limited to polyhedral, nonoccluded objects in a clean, highly controlled environment. Most systems are edge based and use a generate-and-test paradigm to estimate the position and orientation of recognized objects. We propose to use an approach based on characteristic views [23] or aspects [24] which suggests that the infinite two-dimensional projections of a three-dimensional object can be grouped into a finite number of topological equivalence classes. An efficient three-dimensional recognition system would require a parallel indexing method to search for object models in the presence of geometric distortions, noise, and occlusion. Our object recognition system using distributed associative memory can fulfill those requirements with respect to characteristic views.

References

[1] H. A. Simon, *The Science of the Artificial*, 2nd ed. Cambridge, MA: MIT Press, 1984.
[2] D. Marr, *Vision*. San Francisco, CA: Freeman, 1982.
[3] T. O. Binford, "Visual perception by computer," in *Proc. IEEE Conf. Systems and Control*, Miami, FL, 1971.
[4] T. Kohonen, *Self-Organization and Associative-Memories*. New York: Springer-Verlag, 1984.
[5] L. Massone, G. Sandini, and V. Tagliasco, "Form-invariant topological mapping strategy for 2D shape recognition," *Comput. Vision, Graphics, Image Processing*, vol. 30, pp. 169–188, 1985.
[6] C. H. Anderson, P. J. Burt, and G. S. Van Der Wal, "Change detection and tracking using pyramid transform techniques," in *Proc. SPIE Conf. Intelligence, Robots, and Computer Vision*, vol. 579, 1985, pp. 72–78.
[7] D. Casasent and D. Psaltis, "New optical transforms for pattern recognition," *Proc. IEEE*, vol. 65, no. 1, pp. 77–84, 1977.
[8] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals," *Proc. IEEE*, vol. 69, no. 5, pp. 529–541, 1981.
[9] R. G. Lane, W. R. Fright, and R. H. Bates, "Direct phase retrieval," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 520–526, 1987.
[10] O. D. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.
[11] G. S. Stiles, and D. L. Denq, "On the effect of noise on the Moore-Penrose generalized inverse associative memory," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 3, pp. 358–360, 1985.
[12] H. Bouma, "Visual search and reading: eye movements and functional visual field," in *Attention and Performance VII*, J. Requin, Ed. Hillsdale, NJ: Erlbaum, 1978.
[13] E. L. Schwartz, "Spatial-mapping in the Primate sensory projection: Analytical structure and relevance to perception," *Biol. Cybern.*, vol. 25, 1977.
[14] H. R. Wilson and J. R. Bergen, "A four mechanism model for threshold spatial vision," *Vision Res.*, vol. 19, pp. 19–32, 1979.
[15] D. Hubel and T. Wiesel, "Brain mechanisms of vision," *Sci. Amer.*, Oct. 1979.
[16] F. W. Cambell, J. Nachmias, J. Jukes, "Spatial frequency discrimination in human vision," *J. Opt. Soc. Amer.*, vol. 60, pp. 555–559, 1970.
[17] H. J. Reitboek and J. Altmann, "A model for size-and-rotation-invariant pattern processing in the visual system," *Biol. Cybern.*, vol. 51, pp. 113–121, 1984.
[18] J. L. McClelland, D. E. Rumelhart, and the PDP Research Group, Eds., *Parallel Distributed Processing*, vol. 1, no. 2. Cambridge, MA: MIT Press, 1986.

[19] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc. Nat. Acad. Sci.*, vol. 79, Apr. 1982.
[20] J. A. Anderson, J. W. Silverstein, S. A. Ritz, and R. S. Jones, "Distinctive features, categorical perception, and probability learning: some applications of a neural model," *Psychol. Rev.*, vol. 84, pp. 413–451, 1977.
[21] R. A. Messner and H. H Szu, "An image processing architecture for real time generation of scale and rotation invariant patterns," *Comput. Vision, Graphics, Image Processing*, vol. 31, pp. 50–66, 1985.
[22] S. Case, P. R. Haugen, and O. J. Lokberg, "Multi-facet holographic optical elements for wavefront transformations," *Appl. Opt.*, vol. 20, pp. 2670–2673, 1981.
[23] I. Chakravarty and H. Freeman, "Characteristic views as a basis for 3-D object recognition," in *Proc. SPIE Robot Vision*, vol. 336, pp. 37–45, 1982.
[24] J. J. Koenderink and A. J. Van Doorn, "Internal representation of solid shape with respect to vision," *Biol. Cybern.*, vol. 32, no. 4, pp. 211–216, 1979.

Harry Wechsler (SM'86) received the Ph.D. degree in computer science from the University of California, Irvine, in 1975.

From 1976 to 1978, he was an Assistant Professor and Systems Manager for the Advanced Automation Research Laboratory (AARL) in the Department of Electrical Engineering at Purdue University. In 1978, he joined the University of Wisconsin as an Assistant Professor of Computer Science and Electrical Engineering. In 1980 he joined the University of Minnesota as an Associate Professor of Electrical Engineering. Since 1987 he has been a Professor of Computer Science with George Mason University. His academic career includes also visiting Professorships at INRIA, Kyoto University, the Technion, and the Center for Automation Research at the University of Maryland. His research activities focus on computer vision, neural networks, artificial intelligence, and medical image processing (in cooperation with the Mayo Clinic). The main emphasis of his recent research has been on invariant recognition and the use of joint space/spectral image representations for visual tasks like texture segmentation and optical flow derivation. He has authored more than 60 publications and is presently writing a book on computational vision scheduled for publication by Academic Press in 1989.

George Lee Zimmerman (S'82) was born in Boise, ID, in 1960. He received the B.S. degree in electrical engineering from the University of Utah in 1984.

Since then he has been pursuing the Ph.D. degree in electrical engineering at the University of Minnesota, Minneapolis. His research interests include computational vision, biological vision, and signal processing.

Mr. Zimmerman is a member of Eta Kappa Nu.