To appear in *Robots With Redundancy*, A. Bejczy (Ed.),
Springer-Verlag, 1989

REDUNDANT VISION FOR ROBOTICS

Harry Wechsler
Department of Computer Science
George Mason University
Fairfax, VA 22030

Lee Zimmerman
Department of Electrical Engineering
University of Minnesota
Minneapolis, MN 55455

Abstract: We suggest the distributed associative memory (DAM) model for distributed and fault-tolerant computation as it relates to object recognition tasks. The fault-tolerance is with respect to geometrical distortions (scale and rotation), noisy inputs, occlusion/overlap, and memory faults. We have developed an experimental system for fault-tolerant structure recognition which shows the feasibility of such an approach. We further extended our approach to the problem of multisensory data integration and applied it successfully to the recognition of colored polyhedral objects.

1) 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 and occlusion, and non-uniform 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.

Our goal is to derive the functional characteristics of image representations suitable for invariant recognition using distributed processing methods. One has to seek appropriate transformations such that interactions between the internal structure of the resulting representations yield invariant recognition. As Simon (1982) points out, all mathematical derivation can be viewed simple as a change of representation, making evident what was previously true but obscure. Solving a problem then means transforming it so as to make the solution transparent.

The recognition process should match a derived input representation of the short term memory (STM) against long term memory (LTM). We suggest the LTM organization in terms of distributed associative memory (DAM). The DAM is related to generalized match filters (GMF) (Caulfield and Weinberg, 1982) and synthetic discriminant functions (SDF) (Hester and Casasent, 1981). Like them, DAMs attempt to capture a distributed representation which averages over the variations belonging to the same class. DAMs allow for the implicit representation of structural relationships and contextual information, helpful constraints for choosing among different interpretations. Finally, because information is distributed in the memory, the overall function of the memory becomes resistant to noise, faults in memory, and degraded stimulus key vectors.

2) System Outline

This section describes our invariant object recognition system in detail. We begin by examining the preprocessing system which produces the vectors associated by the distributed associative memory. Then the distributed associative memory and our method of fusing multiple sources of information are described and analyzed. A block diagram of the system is shown in Figure 1.

2.1) Invariant Representation

The goal of this subsection is to examine the various components used to produce 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 Figure 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 (See 2.1.3). 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.

2.1.1) Complex-Log Mapping and Space Variant Filtering

The first box of the block diagram given in Figure 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. If the image is mapped onto 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

$$w = \ln(z) = \ln(|z|) + j\theta_z \qquad (1)$$

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

Our system sampled 256x256 pixel images to construct 64x64 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 in selecting the appropriate number of samples for a complex-log mapped image is given in (Massone et. al., 1985). The non-linear 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 anti-aliasing filter is needed. Samples taken at the edge of the image are large and an anti-aliasing 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 a number of angular samples and radial samples. 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.

A problem associated with the complex-log mapping is sensitivity to center misalignment of the sampled image. Small shifts from the center causes distortions in the complex-log mapped image. 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 (see 3.2). 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 (1985). 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.

2.1.2) Fourier Transform

The second box in the block diagram of Figure 1 is the Fourier transform. The Fourier transform of a 2-dimensional image $f(x,y)$ is given by

$$F(u,v) = \int_{-\infty}^{\infty}\int f(x,y)\exp[-j(ux+vy)]dxdy \qquad (2)$$

and can be described by two 2-dimensional functions corresponding to the magnitude $|F(u,v)|$ and phase $_F(u,v)$. 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 $|\cdot|_1$ will not change significantly with rotation and scale of the object in the image.

2.1.3) Laplacian

The Laplacian that we use is a difference-of-Gaussians (DOG) approximation to the $\nabla^2 G$ function as given by Marr (1982).

$$\nabla^2 G = \frac{1}{\pi\sigma^4}\,[1 - r^2/2\sigma^2]\exp\{-r^2/2\sigma^2\} \qquad (3)$$

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. The Laplacian eliminates high frequency spreading due to the finite structure of the discrete Fourier transform and enhances the differences between memorized objects by accentuating edges and de-emphasizing areas of little change.

2.2) Distributed Associative Memory (DAM)

The particular form of distributed associative memory that we deal with in this paper is a memory matrix which modifies 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 (Hebb, 1949).

2.2.1) 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

$$M\vec{s}_i = \vec{r}_i \quad \text{for } i = 1,\ldots,n \tag{4}$$

where $\vec{s}_i$ denotes the i^{th} stimulus vector and $\vec{r}_i$ denotes the i^{th} corresponding response vector. We want to construct a memory matrix M such that when the k^{th} stimulus vector $\vec{s}_k$ is projected onto the space defined by M the resulting projection will be the corresponding response vector $\vec{r}_k$. Specifically we want to solve the following equation:

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

where $S = [\vec{s}_1|\vec{s}_2| \ldots|\vec{s}_n]$ and $R = [\vec{r}_1|\vec{r}_2| \ldots|\vec{r}_n]$. 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 (Kohonen, 1984)

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

$$\tilde{r} = M\tilde{s} \tag{7}$$

If the memorized stimulus vectors are independent and the unknown stimulus vector $\tilde{s}$ is one of the memorized vectors $\vec{s}_k$, then the recalled vector will be the associated response vector $\vec{r}_k$. 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. 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 least 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.

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. 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

$$\vec{r} = \vec{r}_k + \vec{y}_o = M(\vec{s}_k + \vec{y}_i) = \vec{r}_k + M\vec{y}_i \qquad (8)$$

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

$$\frac{\sigma_o^2}{\sigma_i^2} = \frac{1}{m} Tr[MM^T] \qquad (9)$$

For the autoassociative case this simplifies to

$$\frac{\sigma_o^2}{\sigma_i^2} = \frac{n}{m} \qquad (10)$$

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 (Stiles and Denz, 1985).

$$\frac{\sigma_o^2}{\sigma_i^2} = \frac{1}{m} Tr[RR^T]Tr[(S^TS)^{-1}] \qquad (11)$$

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.

2.3) Data Fusion

We suggest next using distributed associative memory to integrate visual information from multiple cues or sources for performing image interpretation tasks. A source amounts to a 2-dimensional function extracted from an image which carries some distinct information about the 3-dimensional scene. In our particular case the sources of information can take on two forms. direct or derived. A direct source is one where the information is directly in registration with the iconic image. Examples would be infra-red images, range images, spectral band images, etc.. A

derived source is one which requires some type of preprocessing before the spatial information is made explicit. Preprocessing examples are stereopsis and optical flow derivation. Integration means combining information from several distinct sources to produce a response which is possibly quite different from the input information. An example of integration would be the ability of human vision to extract reliable viewer-centered depth information from the binocular and motion information contained in a set of images. In this case the 'sensors', stereopsis and optical flow, are distinct but both carry geometric information about the scene. In order for information from distinct sources to be integrated at some point they must speak the same language.

Looking for a general method for combining multiple distinct sources into a unified interpretation is a problem which has been examined from many angles. Statistical approaches such as Garvey and Lowrance's (1983) threat assessment for battle management which uses Dempster/Shaffer model for source integration, need much a-priori information about the statistical variations between the different source measurements and are computationally very expensive. Hybrid approaches such as Waxman and Duncan's (1986) stereomotion, use extensive knowledge about how the sources vary with each other in order to develop their algorithm, and as a result are not general or easily extensible to the addition of other sources. Both of these methods need higher level processing to handle conflicts in source interpretation.

Our approach attempts to overcome the above problems. We assume that the output of the sources vary consistently for a given input. Let $S = [\vec{s}_1|\vec{s}_2| \ldots|\vec{s}_n]$ and $R = [\vec{r}_1|\vec{r}_2| \ldots|\vec{r}_n]$ be the stimulus and response matrices respectively. Each stimulus vector is made up of elements coming from the different sources $\vec{s}_i^T = [\vec{s}_{i_1}^T|\vec{s}_{i_2}^T| \ldots|\vec{s}_{i_\ell}^T]$ where $\vec{s}_{i_j}^T$ consists of elements from the j^{th} source. The memory matrix is constructed in the same fashion as before, $M = RS^+$.

Integration of sources in this manner has several positive properties. First, it is quite general and as such it is easily extensible to multiple sources. If new sources need to be added they simply augment the previous stimulus vector structure. Second, prioritizing the influence that a single source can have on the output can be done by adjusting the size and/or the quantization of elements that source donates to the stimulus vectors.

3) Experiments

In this section we discuss the result of computer simulations of our system. Images of objects are first preprocessed through the subsystem outlined in subsection 2.2. The output of such a subsystem is four vectors: $|\cdot|_1$, Φ_1, $|\cdot|_2$, and Φ_2. We construct the memory by associating the stimulus vector $|\cdot|_1$ 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 $\widetilde{|\cdot|}_1$, $\widetilde{\Phi}_1$, $\widetilde{|\cdot|}_2$, and $\widetilde{\Phi}_2$. The resulting stimulus vector $\widetilde{|\cdot|}_1$ 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 $|\tilde{\cdot}|_1$ are used to reconstruct the memorized object. The difference between the estimated phase $\hat{\phi}_2$ and the phase $\tilde{\phi}_2$ is used to estimate the amount of rotation and scale experienced by the object.

3.1) Invariant Recognition

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 256x256, eight bit quantized pixels, and each object covers an area of about 40x40 pixels. This small object size relative to the background is necessary due to the non-linear sampling of the complex-log mapping. 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 2-dimensional images that are the output from the invariant representation subsystem are scanned horizontally to form the vectors for memorization.

The first example of the operation of our system is shown in Figure 2. Figure 2a) is the image of one of the key as it was memorized. Figure 2b) is the unknown object presented to our system. The unknown object in this case is the same key that has been scaled. Figure 2c) 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 key with some noise caused by errors in the recalled phase. Figure 2d) is a histogram which graphically displays the classification vector which corresponds to $\tilde{S}^T\hat{S}$. The histogram shows the interplay between the memorized images and unknown image. The "2" on the bargraph indicates which of the twelve classes the unknown object belongs. 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 vector $\bar{D}$ corresponding to the difference between the unknown vector $\tilde{\phi}_2$ 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.

Figure 3 is the result of randomly setting the elements of the memory

matrix to zero. Figure 3a) shows the ideal recall. Figure 3b) is the recall after 30 percent of the memory matrix has been set to zero. Figure 3c) is the recall for 50 percent and Figure 3d) 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. Wechsler and Zimmerman (1989) provide additional details on the invariant recognition outlined in this section.

3.2) Sensory Integration

This section describes a series of experiments whose goal is to assess our approach to the data fusion problem. The databased is made up of 15 classes of colored polyhedral objects. The objects were video taped from directly above while sitting on a flat surface. Each object class has at least one other class which presents the same shape in the image. Also several of the classes consistent of objects of the same color. As discussed in section 3.3 the stimulus vectors are of the form $\vec{s}_i^T = [\vec{s}_{i_R}^T | \vec{s}_{i_G}^T | \vec{s}_{i_B}^T]$ where R, G, B corresponds to the red, green, and blue channels of the video recorder. The geometrical transformations were obtained by rotating or translating the video camera. The images used to test the system were taken at a different time than the ones used to construct this memory.

Table 1 specifies the database (DB) and the transformations the objects were subject to before the memory recall operation is performed. Each object, subject to the linear transformation specified in Table 1 is correctly recognized. The actual recall histograms are shown in Table 2. Table 3 shows the resulting histograms obtained in the presence of occlusion or noise. Approximately 1/3 of the yellow 12 sided object was erased from the image. A comparison can be made between the results of the recall with and without occlusion. A uniformly distributed noise of variance and mean equal to that of the image of the green six sided object was added. The object recall, although diminished, was still correct.

The next series of experiments was aimed at exploring the capability of our system when exposed to overlap situations, like those encountered during bin-picking. The database for this memory consists of six objects. Three of the objects have a side view learned state along with the standard top-down view used in the previous example. This memory is tested using three objects which overlap (the central object is supported by the other two). The camera moves, similar to a conveyor belt, and samples the image in five distinct locations. Figure 4 shows several graphs which indicate how the recall histogram varies as the camera is moved. The top three graphs show the response of the three objects which were used in the test. Each of these objects were memorized in two views - one from the top looking down and a second view from the side. The dotted line on all of the graphs shows the maximum response given by any object in the memory. The results show that for points close to the central locations of the

objects being viewed the response is correct. For viewpoints between two objects the response was dominated by one of the objects present in the input. The drawing 4e) shows the placement of the sampled center points.

4) Conclusion

Our experiments demonstrate the feasibility of our system for 2-dimensional invariant recognition and data fusion. Information, implicit in the input patterns, is made explicit through the preprocessing subsystem. This is followed by the distributed associative memory which stores and retrieves information. The DAM, because of its distributed nature, lends robustness to the overall system function in the presence of noise, occlusion, and other factors which degrade performance. There are some properties of the DAM which are crucial for data integration. In the DAM, the parameters necessary for combining multiple sources are extracted from the environment and recall is context sensitive. We feel an important aspect of our system is the natural way it processes iconic information which is in contrast to AI symbolic approaches.

The DAM, like most other NN models, does not exploit the topographical power naturally present in visual information. Examinations of the visual context have shown that visual information is processed through multiple mappings of the visual fields. Incorporation of this type of information into a NN model could lead to dramatic compression of the necessary number of connections and more processing power for visual tasks. All pattern recognition techniques, including NN, are sensitive to the nature of the training set. If the training set is not representative of the classes which will be encountered in the environment then the abilities of the system to classify and generalize will be hampered. Research in the area of NN is in its infancy. We expect the future to bring better understanding of these characteristics along with new methods of learning.

The computer vision system presented in this paper was designed with only 2-dimensional metric distortions in mind. We have shown some ability to deal with clustered 3-dimensional polyhedra and are presently extending our work toward 3-dimensional object recognition. We propose to use an approach based on characteristic views (Chakravarty and Freeman, 1982) or aspects (Koenderink and Van Doorn, 1979) which suggests that the infinite 2-dimensional projections of a 3-dimensional object can be grouped into a finite number of topological equivalence classes. An efficient 3-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. The strength of biological vision and the weaknesses in computational vision when it comes to analyzing, sensing, and integrating information from the environment, suggest that certain aspects need to be incorporated into future systems. First, the interpretation of the environment should be done in 3-dimensions because the world is 3-dimensional. Second, maintaining spatial integrity of the scene and its projection is extremely important for recognition and integration. Third, the integration of information should take place in an active manner. Finally, and most importantly, methods of integration need to include methods for verifying assumptions and learning from the environment.

References

1. Anderson, C. H., P. J. Burt, and G. S. Van Der Wal: Change detection and tracking using pyramid transform techniques, Proc. of the SPIE Conf. on Intelligent Robots, and Computer Vision, Vol. 579, pp. 72-78, 1985.

2. Caulfield, H. J. and M. H. Weinberg: Computer recognition of 2-D pattern using generalized matched filters, Applied Optics, Vol. 21, no. 9, 1982.

3. Chakravarty, I., and H. Freeman: Characteristic views as a basis for 3-D object recognition, Proc. SPIE of Robot Vision, Vol. 336, pp. 37-45, 1982.

4. Faugeras, O. D., and M. Hebert: The representation, recognition, and positioning of 3-D shapes from range data, Techniques for 3-D Machine Perception, A. Rosenfeld (Ed.), North-Holland, pp. 13-52, 1986.

5. Hartline, P. L. Kass, and M. Loop: Merging of modalities in the optic tectum: infrared and visual integration in the rattlesnake, Science, Vol. 199, pp. 545-548, 1978.

6. Hebb, D. O.: The Organization of Behavior, New York: Wiley.

7. Hester, C., and D. Casasent: Interclass discrimination using synthetic discriminant functions (SDF), Proc. SPIE on Infrared Technology for Detection and Classification, Vol. 302, 1981.

8. Koenderink, J. J., and A. J. Van Doorn: Internal representation of solid shape with respect to vision, Biological Cybernetics, Vol. 32, no. 4, pp. 211-216, 1979.

9. Kohonen, T.: Self-Organization and Associative-Memories, Springer-Verlag, 1984.

10. Massone, L., G. Sandini, and V. Tagliasco: Form-invariant topological mapping strategy for 2D shape recognition, Computer Vision Graphics and Image Processing, Vol. 30, 169-188, 1985.

11. McClelland, J. L., D. E. Rumelhart, and the PDP Research Group (Eds.): Parallel Distributed Processing, Vol. 1, 2, MIT Press, 1986.

12. Simon, H. A.: The Sciences of the Artificial (2nd ed.), MIT Press, 1982.

13. Stiles, G. S. and D. L. Deng: On the effect of noise on the Moore-Penrose generalized inverse associative memory, IEEE Trans. on PAMI, Vol. 7, no. 3, pp. 358-360, 1985.

14. Waxman, A. M., and J. H. Duncan: Binocular image flows: steps toward stereo motion fusion, IEEE Trans. on PAMI, Vol 8, no. 6, 715-729, 1986.

15. Wechsler, H. and G. L. Zimmerman: 2-D invariant object recognition using distributed associative memory, IEEE Trans. on PAMI (to appear), 1989.

Acknowledgement: This work was supported in part by National Science Foundation under grant NSF-EET8713563.

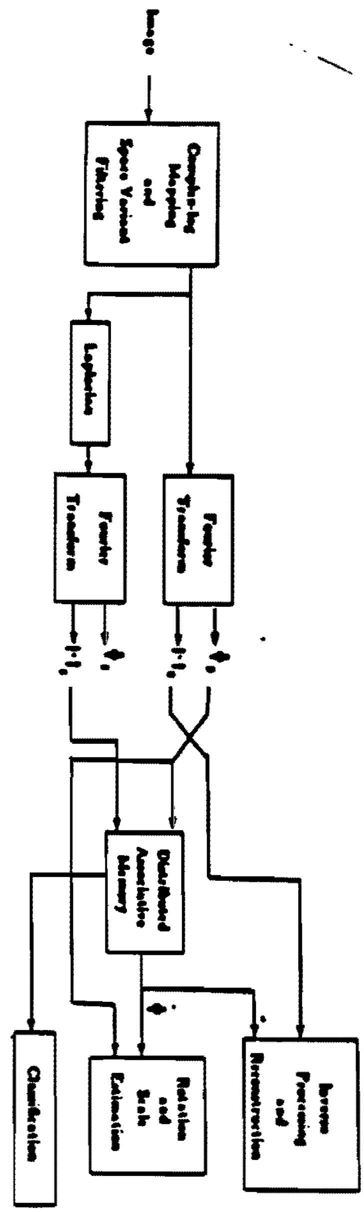

Figure 1: Block Diagram of the System.

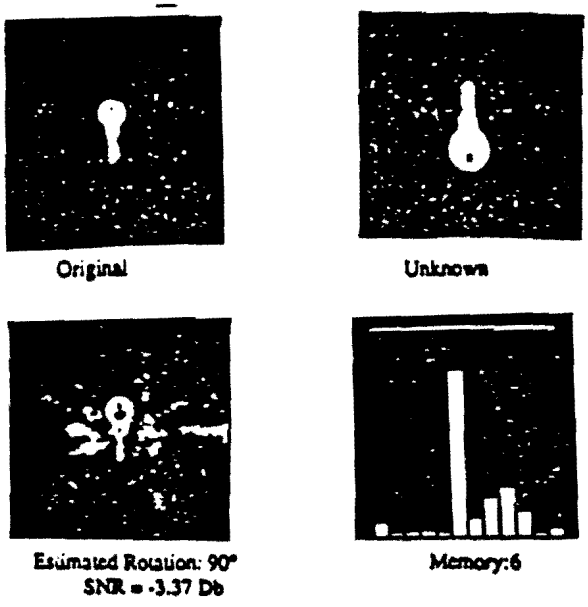

Figure 2. Recall Using Rotated and Scaled Key

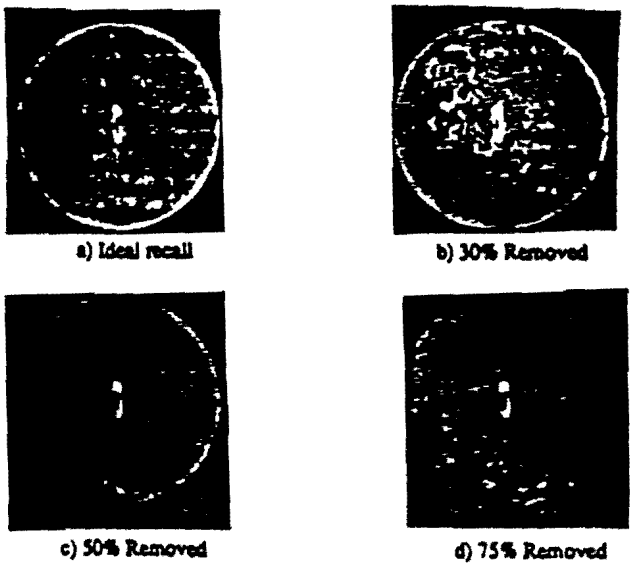

Figure 3. Recall for Memory Matrix Randomly Set to Zero

20b - 20 sided medium blue; rotation approx. 30 right
20g - 20 sided gray; rotation 15 left
12r - 12 sided red; smaller with rotation; very dim
12y - 12 sided yellow; smaller
10b - 10 sided light brown; smaller rotated 20
10o - 10 sided orange; rotation 40
8b - 8 sided light blue; rotation 30
8bl - 8 sided black; smaller; extremely dim
8o - 8 sided orange;larger
6b - 6 sided dark blue; larger
6g - 6 sided green; rotation 45
6y - 6 sided yellow; larger
4b - 4 sided dark blue; rotation 45
4g - 4 sided light blue; smaller rotation 10
4p - 4 sided lavender; rotation 30

Table 1. - Description of 15 Classes of Colored Polyhedral Objects and Their Corresponding Transformations.

10b = scaled and rotated
20b = -0.037122
20g = 0.112992
12r = 0.074970
12y = -0.002770
10b = 0.589933
10o = 0.389303
8b = -0.008726
8o = -0.038870
8bl = -0.260333
6b = -0.039565
6g = -0.011397
6y = -0.034024
4b = 0.010866
4g = 0.018852
4p = 0.083681
maximum: 10b = 0.589933
minimum: 8bl = 0.260333

20b = rotated
20b = 0.669728
20g = 0.085861
12r = -0.063536
12y = -0.021761
10b = 0.027957
10o = 0.051344
8b = 0.038651
8o = -0.027160
8bl = -0.206991
6b = 0.189514
6g = 0.076489
6y = -0.013071
4b = 0.001625
4g = -0.000555
4p = 0.080724
maximum: 20b = 0.669728
minimum: 8bl = -0.206991

4g - scaled and rotated
20b = 0.099462
20g = -0.005444
12r = 0.101725
12y = -0.092328
10b = 0.102195
10o = -0.020118
8b = -0.020353
8o = 0.156823
8bl = 0.239798
6b = 0.177184
6g = 0.035344
6y = 0.007695
4b = 0.112544
4g = 0.451811
4p = 0.023719
maximum: 4g = 0.451811
minimum: 12y = -0.092328

6b - scaled
20b = 0.121183
20g = 0.018289
12r = 0.078140
12y = -0.041012
10b = 0.082051
10o = -0.084216
8b = 0.009179
8o = -0.085697
8bl = 0.274767
6b = 0.330043
6g = 0.152556
6y = 0.040148
4b = 0.006260
4g = 0.005951
4p = 0.091934
maximum: 6b = 0.330043
minimum: 8o = -0.085697

Table 2. Responses of the Fusion Memory

8b - rotated
20b = 0.149574
20g = 0.226339
12r = 0.091095
12y = 0.033431
10b = 0.072598
10o = -0.084049
8b = 0.553082
8o = 0.277205
8bl = 0.332871
6b = -0.155819
6g = -0.043836
6y = -0.031867
4b = -0.051010
4g = 0.070805
4p = -0.201123
maximum: 8b = 0.553082
minimum: 4p = -0.201123

10o - rotated
20b = -0.107486
20g = 0.016014
12r = -0.019775
12y = 0.026568
10b = 0.213287
10o = 0.519179
8b = 0.014449
8o = -0.049224
8bl = -0.481672
6b = 0.266687
6g = 0.055487
6y = -0.006067
4b = 0.075286
4g = 0.061362
4p = 0.208430
maximum: 10o = 0.519179
minimum: 8bl = -0.481672

12y - scaled
20b = -0.016374
20g = -0.030442
12r = 0.007098
12y = 0.684478
10b = 0.068539
10o = 0.002150
8b = 0.072255
8o = -0.128156
8bl = -0.156196
6b = 0.258971
6g = -0.085213
6y = 0.036109
4b = 0.044082
4g = 0.116001
4p = -0.021450
maximum: 12y = 0.684478
minimum: 8bl = -0.156196

4b - rotated
20b = 0.033306
20g = -0.043632
12r = 0.102113
12y = -0.053745
10b = -0.021697
10o = 0.045749
8b = 0.077244
8o = 0.127757
8bl = 0.302109
6b = 0.038464
6g = -0.060469
6y = 0.057145
4b = 0.347842
4g = 0.089829
4p = 0.012316
maximum: 4b = 0.347842
minimum: 6g = -0.060469

6g - rotated	8b1 - scaled	12r - scaled and rotated	20g - rotated
20b = -0.049967	20b = -0.009174	20b = -0.101214	20b = 0.074942
20g = 0.068956	20g = -0.018269	20g = 0.061621	20g = 0.789082
12r = -0.166670	12r = 0.046817	12r = 0.333425	12r = 0.209688
12y = 0.003740	12y = 0.016891	12y = -0.011372	12y = -0.032170
10b = 0.118696	10b = -0.014401	10b = 0.090512	10b = 0.009353
10o = 0.046837	10o = 0.012176	10o = 0.107102	10o = 0.001790
8b = -0.007574	8b = -0.000099	8b = 0.042359	8b = 0.097038
8o = -0.077262	8o = -0.006983	8o = -0.090681	8o = 0.113085
8b1 = 0.009852	8b1 = 0.445833	8b1 = 0.138120	8b1 = 0.083690
6b = 0.012869	6b = 0.099586	6b = 0.119825	6b = 0.187161
6g = 0.461146	6g = 0.017941	6g = -0.071757	6g = -0.066753
6y = 0.210418	6y = -0.013268	6y = 0.077901	6y = 0.073837
4b = 0.080270	4b = 0.025530	4b = -0.016150	4b = -0.145125
4g = 0.052528	4g = 0.020772	4g = 0.032028	4g = -0.046533
4p = -0.078049	4p = 0.087969	4p = 0.053942	4p = -0.283062
maximum: 6g = 0.461146	maximum: 8b1 = 0.445833	maximum: 12r = 0.333425	maximum: 20g = 0.789082
minimum: 12r = -0.166670	minimum: 20g = -0.018269	minimum: 20b = -0.101214	minimum: 4p = -0.283062

Table 2. Responses of the Fusion Memory

4p - rotated	6y - scaled	8o - scaled
20b = 0.135660	20b = 0.026048	20b = 0.067367
20g = -0.034477	20g = -0.044931	20g = 0.056499
12r = -0.036622	12r = -0.226118	12r = 0.108363
12y = -0.020434	12y = -0.033081	12y = -0.003583
10b = 0.015881	10b = 0.107014	10b = -0.117320
10o = 0.087427	10o = 0.043923	10o = 0.009010
8b = -0.062108	8b = -0.078311	8b = -0.051742
8o = 0.044841	8o = 0.035680	8o = 0.986751
8b1 = 0.134441	8b1 = -0.225961	8b1 = -0.184686
6b = 0.063891	6b = 0.066302	6b = -0.202276
6g = 0.063415	6g = 0.645227	6g = 0.031471
6y = -0.022305	6y = 0.683283	6y = 0.017449
4b = 0.059183	4b = 0.088817	4b = 0.034253
4g = 0.042149	4g = 0.024370	4g = 0.124651
4p = 0.329339	4p = 0.032338	4p = 0.371463
maximum: 4p = 0.329339	maximum: 6y = 0.683283	maximum: 8o = 0.986751
minimum: 8b = -0.062108	minimum: 12r = -0.226118	minimum: 6b = -0.202276

20b = -0.016374	20b = -0.061317
20g = -0.030442	20g = 0.055189
12r = 0.007098	12r = 0.036854
12y = 0.684478	12y = 0.638085
10b = 0.068539	10b = 0.084086
10o = 0.002150	10o = 0.059582
8b = 0.072255	8b = 0.086944
8o = -0.128156	8o = -0.140270
8bl = -0.156196	8bl = -0.566859
6b = 0.258971	6b = 0.114629
6g = -0.085213	6g = -0.098479
6y = 0.036109	6y = 0.043607
4b = 0.044082	4b = 0.118768
4g = 0.116001	4g = 0.107182
4p = -0.021450	4p = 0.100319
maximum: 12y = 0.684478	maximum: 12y = 0.638085
minimum: 8bl = -0.156196	minimum: 8bl = -0.566859
without occlusion	with occlusion

6g - rotated

20b = -0.049967	20b = -0.039071
20g = 0.068956	20g = 0.098584
12r = -0.166670	12r = -0.141735
12y = 0.003740	12y = -0.008002
10b = 0.118696	10b = 0.110771
10o = 0.046837	10o = 0.091265
8b = -0.007574	8b = -0.051865
8o = -0.077262	8o = -0.045028
8bl = 0.009852	8bl = 0.070492
6b = 0.012869	6b = -0.042576
6g = 0.461146	6g = 0.413561
6y = 0.210418	6y = 0.204100
4b = 0.080270	4b = 0.077749
4g = 0.052528	4g = 0.093187
4p = -0.078049	4p = -0.097903
maximum: 6g = 0.461146	maximum: 6g = 0.413561
minimum: 12r = -0.166670	minimum: 12r = -0.141735
without noise	noise (OOb,mean=5)

Table 3. Invariant Recognition to Occlusion and Noise

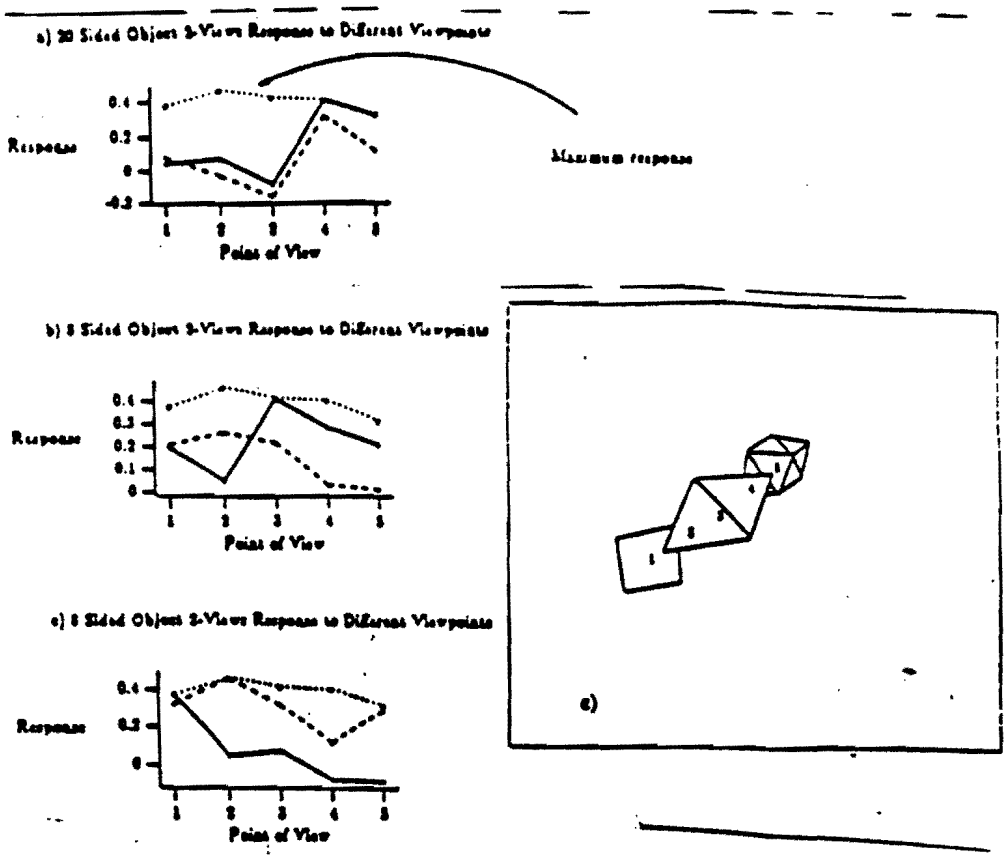

Figure 4 Response of memory at multiple viewpoints for the overlapped configuration