89-12

Distributed Associative Memory (DAM) for Bin-Picking

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

Abstract—We show the feasibility of using a distributed associative memory as the recognition component for a bin-picking system. The system displays invariance to geometric distortions and a robust response in the presence of noise, occlusions, and memory faults. Although our system is primarily concerned with two-dimensional object recognition we suggest how our method can be extended to handle the more general three-dimensional case.

Index Terms—Bin-picking, distributed associative memory (DAM), fault-tolerance, invariance, object recognition, parallel computation.

I. Introduction

OUR research is directed at solving the problem of recognizing objects subject to changes in the geometry of image formation and illumination. *Recognize* is used here as a generic term for performing classification, reconstruction (i.e., evoking a good quality image from memory), and semantic interpretation. The approach that we use derives a complex-log invariant image representation which along with a distributed associative memory (DAM) yields a system able to recognize a memorized object regardless of the scale and rotation it has been subject to. The system allows for noise and overlap, is fault-tolerant, and could be implemented using a parallel distributed architecture.

The challenge of the visual recognition problem stems from the fact that the interpretation of an image is confounded by several dimensions of variability. Such dimensions include uncertain perspective, changing orientation and scale, sensor noise, object occlusion and overlap, and nonuniform illumination. Machine vision systems must not only be able to sense the identity of objects despite this variability, but they must also be able to characterize such variability. This is so because the variability in the image formation process inherently carries much of the valuable information about the world surrounding us. It is interesting to note that humans can recognize objects when viewed from novel orientations, moderate levels of noise, and overlap/occlusion, while new instances of a category can be rapidly classified. Available machine vision systems lack such characteristics. They assume limited and known noise, even illumination, isolated objects, and the object classes are learned for a limited number of stable states given by their attitude in space. Both industrial automation, including the bin-picking problem, and target tracking are severely hampered by such restrictions.

We define the generic bin-picking problem as the search for the identity of objects O_i within a mix given by $\Sigma\ \alpha_i$ where α_i can be one of the following:

1) $T_i(O_i)$ for some geometric change T in object scale and orientation;

2) N for noise in both input and the storage memory; and

3) A for ambient illumination.

Solving the bin-picking problem is equivalent to unscrambling the mix of degraded objects and picking-up the identity of the unknown objects one at a time. One of the crucial concepts for our approach is the availability of transformations [23]. All mathematical derivation can be viewed simply as a change of representation, making evident what was previously true but obscure. This view can be then extended to all of problem solving—solving a problem means representing (transforming) it so as to make the solution transparent. Vision is a sequence of transformations, some of them carried out in parallel, whose goal is to capture some invariant aspects of the world surrounding us.

We begin with a review of solutions to the bin-picking problem. This review is concerned with both 2-D and 3-D solutions and drives the development of our own system. The third section is a brief description of our invariant recognition system which is outlined more thoroughly in [24]. The fourth section describes a series of experiments we performed. We initially address only the 2-D bin-picking problem and then show a promising extension to the system allowing a partial solution to the 3-D problem. We conclude with our goals for future research.

II. Review of Bin-Picking

Industrial automation, both for assembly and quality inspection reasons, requires the capability to recognize both the identity and the attitude in space of objects despite image variability (noise, uneven illumination, geometric distortions, and occlusion). Invariant recognition despite

Manuscript received March 9, 1987; revised July 6, 1988. Recommended for acceptance by J. L. Mundy. 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 School of Information Technology and Engineering, 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 8928495.

such variability is generically known as the bin-picking problem. A survey of solutions suggested to solve the problem for flat 2-D patterns and volumetric 3-D patterns is presented. Each method is assessed in terms of computational requirements (memory and speed of computation), performance, invariance, and system approach.

The case can be made that once a computational task is defined, one has to choose appropriate image representations and the algorithms suitable to operate on such representations. We detail next some of the main approaches for bin-picking and discuss their limitations. All methods are based on matching between derived image representations and "learned" models prestored in memory.

A. 2-D Approaches

1) The Feature-Based Approach: Most of the commercially available machine vision systems [9] are restricted to binary vision and/or silhouette defined objects. Connectivity analysis for defining blobs and/or edge detection to determine outlines are the main methods used to locate potential objects. Information about the surface of the object, like texture, is discarded and the environment is assumed to be noise free. Specifically, the illumination is even, no shadows are allowed and objects do not touch each other. Feature extraction yields properties like perimeter (P), area (A), major and minor axes, minimum and maximum radii (from the center of gravity to object's boundary), compactness ($C = P^2/4\pi A$). Such properties are invariant to rotation and translation. Scale invariance is explicitly achieved through system calibration. Statistical classifiers then use such features for object recognition. The silhouette based systems use a polygonal approximation for the object's boundary. The segments used in approximation are like letters in some alphabet and their characteristics are in terms of length and orientation. The joins (angle) between such segments approximate grammatical constraints. Syntactical pattern recognition and/or graph search algorithms are then used to identify the objects.

2) Intrinsic Functions and Correlation: Once occlusion is taken into consideration there is the need for subtemplate matching. One particular object representation which facilitates such a task is the intrinsic ($\theta - s$) (parametric) curve [18] where the angle of the slope (θ) and the corresponding arc length (s) along the boundary are used as a new system of coordinates. The $\theta - s$ representation allows descriptions of the form $\theta(s)$ where a change in orientation (α) in the $x - y$ space corresponds to simply adding (α) to $\theta(s)$ in the $\theta - s$ space. The method is sensitive to the quantization noise introduced when determining the boundary curvature $\theta(s)$. Furthermore, one has yet to determine what subtemplates are salient.

3) The Hough Transform: The Hough transform is implemented through the use of an accumulator array, and subtemplates ($\theta - s$) matches for each of the (mechanical) parts to be recognized point to possible objects and their centroids. The goodness of the fit is used to increment the accumulator cells selectively. Specifically, if B and T stand for the object boundary and a particular template, respectively, then by defining $\tilde{T}$ as T rotated by 180°, it can be shown [21] that convolution between B and $\tilde{T}$ is equivalent to cross-correlation between B and T. In other words, the attempt to implement the Hough transform by looking for potential matches between the boundary B and the rotated template $\tilde{T}$ is equivalent to match-filter detection. One can hardly overemphasize the fact that most of object recognition and therefore, bin-picking, is match filter in one form or another. Occlusion is dealt with by determining the saliency of the subtemplates in terms of local properties. Orthogonality of the subtemplates-coordinates systems is achieved through factor analysis/minimization over the whole set of parts to be recognized.

Saliency of subtemplates through appropriate weighting factors can improve the unrestricted use of the Hough transform. Drawbacks in using the Hough concept are mainly related to the memory requirements needed to implement big accumulator arrays, and the computation itself, which can be quite wasteful due to the exhaustive nature of the algorithm.

4) Match Filters: The solutions suggested for solving the bin-picking problem vary in their generality. Partial solutions can be considered as well, whereby one merely attempts to locate potential holdsites for enabling a robot end-effector to grasp and to hold to a mechanical part. Recognition of the part, if and when needed, could follow. Dessimoz, Birk, Kelley, and Martins [11] suggest the use of match filters for a task as described above. Assume that the holdsites are described by local masks $p(i, j)$ and that the original image (D) is given as $f(i, j)$. Potential holdsites are then located when the similarity between f and p is high, or equivalently for those locations where the error E^2 is minimized. E^2 is minimized if the cross-correlation of f and p as given by $f(i + m, j + n)p(m, n)$ is maximized. If the energy (illumination) of the image is not uniform then normalized cross-correlation should be used.

5) Probabilistic (Labeling) Relaxation: Bhanu and Faugeras [3] suggest the use of probabilistic labeling. The approach is based on relaxation [10] and is similar to the Viterbi algorithm. The object is given as a polygonal approximation and one attempts to label each segment as one of the predefined subtemplates. Optimality is achieved by seeking for the best solution in a global sense. Computational requirements are high as noted by Ayache *et al.* [1] because all the potential solutions are calculated and stored.

6) Graph Theory: Given one image representation and a corresponding memory model one has to perform some kind of (sub)graph isomorphism, which is known to be NP-complete. Different strategies can be used to speed-up the computation. One can look for maximum cliques [5], use some heuristic evaluation function (to prune out unlikely matches) within the framework of the A^* algorithm [2], and/or do clustering analysis in some feature space, maybe helped by the MST (minimum spanning tree) approach.

B. 3-D Methods

1) Geometry: An idea whose origin comes from differential geometry is that of the Gaussian sphere and EGI (extended Gaussian sphere) [13]. The EGI is equivalent to an orientation histogram corresponding to the normals ("needles") as estimated across a 3-D structure.

The EGI have many potential problems. The "needle" map is obtained through the use of photometric stereo and is highly dependent on the illumination. Fluctuating illumination and shadows can easily distort the results. As Yang and Kak [26] point out, the EGI method could benefit from the use of range instead of brightness as an intrinsic characteristic. High computational requirements and the difficulty to deal with occlusion are additional drawbacks of the EGI. The generation of models in terms of their EGI is also quite difficult.

2) Deprojection and Recursive Estimation: Ayache and Faugeras [1] suggest an approach called HYPER (hypotheses predicted and evaluated recursively). The method has an AI flavor by considering (suggesting) different hypotheses, making predictions and then verifying them. The loop is traversed until enough confidence is gained. Estimation about the attitude of the object, assuming rigidity, is made recursively through the application of a Kalman filter. Specifically, the objects are given as polygonal approximations, including salient segments with which the match starts. The affine transformation which maps some model into the image is given as $T(\theta, k, t_x, t_y)$, where θ, k, t_x, t_y stand for rotation, scale, and translation. Estimating T allows one to locate the attitude of the object in space. HYPER operates in an iterative fashion using a Kalman filter.

3) Graphics Primitives or CAD Approach: Computer vision has been characterized by an attempt to find representations which are complete enough to capture phenomena in the three-dimensional environment and limited enough for easy, fast solutions to the tasks at hand. Computer graphics, which seems to be the inverse of the computer vision problem, has well understood the need for primitive representations. As a result much of the work beginning with Roberts [20] in 1967 has been directed at using these graphic primitives for computer vision tasks including the bin-picking problem. Many of these models are edge-based and disregard surface information [12], [16]. Being edge-based they are limited in their ability to extract edges from images in the presence of noise, spurious surface markings, and illumination changes. These methods are difficult to evaluate. There is some evidence for these volumetric primitives being used in biological vision [4].

C. Reevaluating the Problem

A recent survey by Chin and Dyer [8] summarizes and compares different recognition methods. They also conclude that the available methods lack in generality. The 2-D techniques, based on edge (silhouette) detection, fall within three major classes: 1) global feature methods (like the SRI Machine Vision System); 2) structural feature methods [22]; and 3) relational graph methods [5]. The global feature methods are faulted for not being able to handle noisy images and/or occlusion. The structural feature method is expensive both in terms of feature extraction and matching and lacks the invariance property. The relational graph method also fails with respect to invariance and slow matching, and can handle occlusion only if key features are apparent. Finally, 3-D object-centered representations are only starting to appear and there are no conclusive results regarding their applicability.

We describe an alternative, to the methods outlined previously, based on DAM's, which could eventually solve the bin-picking problem by directly addressing the problems of unknown orientation, sensitivity to noise, illumination, and occlusion.

III. Methodology

The machine vision system that we have constructed performs classification (recognition) and reconstruction. The system is invariant to image rotation and scale, and is fault-tolerant. New instances of objects can be classified and reconstructed according to the previously memorized objects. The system responds robustly in the presence of noise, occlusion, and overlap, and random faults in the memory. Such performance is achieved through the integration of two subsystems, as shown in Fig. 1. The first component derives an invariant representation, while the second component builds and interprets the distributed associative memory (DAM). The recall or matched recognition compares a derived invariant image representation with the DAM and yields invariant object recognition. The next sections detail both subsystems.

B. Preprocessing-Invariant Representational Subsystem

The block diagram which describes the various functional units involved in obtaining an invariant image representation is shown in Fig. 1. These units are described more completely in [24]. The filtered 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 [24]. 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 metric 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. The rest of this subsection describes the various components in more detail.

The complex-log mapping transforms radial lines into vertical lines and concentric circles into horizontal lines. Each point (x, y) on the plane can be described mathe-

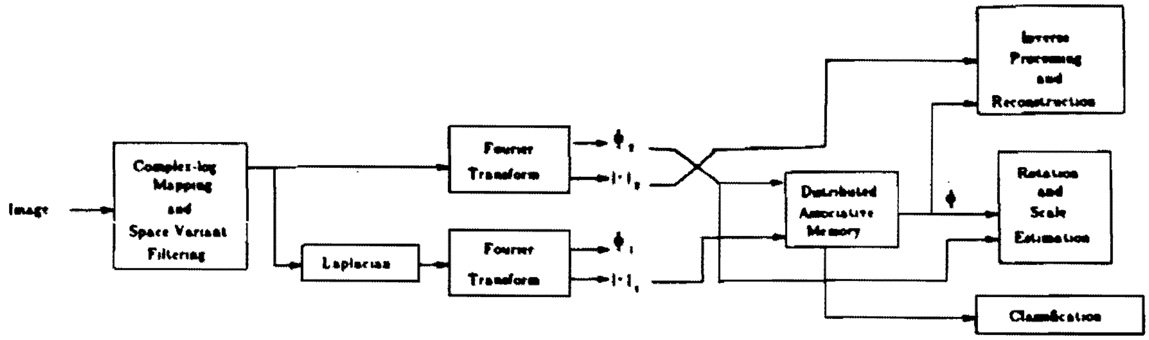

Fig. 1. Block diagram of the system.

matically by $z = x + jy$. The corresponding complex-log mapped points w are given 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 in selecting the appropriate number of samples for a complex-log mapped image is given in [17]. 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 needed. 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. 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. More information on this and similar transformations can be found in [6], [17], [19], [25].

The next 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

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

and can be described by two two-dimensional functions corresponding to the magnitude • and phase ϕ. The magnitude component of the Fourier transform carries much of the contrast information of the image, while the phase component of the Fourier transform carries information about how things are placed in an image. The Fourier transform has many properties which make it a useful tool for pattern recognition. A property, particularly useful for our purpose, is that the magnitude of the Fourier transform is invariant to translation and that information about the translation of an object in an image is carried in the phase of the Fourier transform. Translation corresponds to a linear phase component. In the block diagram of Fig. 1, the complex-log mapping transforms rotation and scaling into translation. The magnitude of the Fourier transform of the complex-log mapped image is invariant to these translations.

The Laplacian is used in our system to handle the problem related to the size invariant aspect of the complex-log mapped image. 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 and not simple translation in the complex-log mapped image. The Laplacian eliminates this problem by sharpening the edges and setting to zero regions that do not change much. It also eliminates the need for windowing the Fourier transform, so that artificial frequencies will not interfere with discrimination. Finally, it extracts edges which can enhance the differences between the objects that are memorized thus improving the memory recall.

B. *Distributed Associative Memory (DAM)*

The key to the overall system is the distributed associative memory [15]. 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 they 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 such 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, cross-referencing, or intramodal data fusion in the memory. As a direct result of distributing the associations, this type of memory lends itself easily to parallel processing. Because information is distributed in the memory, the overall function of the memory becomes resistant to noise, faults in the memory, and degraded stimulus vectors.

The *construction* of the memory begins with n pairs of m-dimensional vectors that are to be associated. This can be written as

$$M\bar{s}_i = \bar{r}_i \qquad \text{for } i = 1, \cdots, n \tag{3}$$

where $\bar{s}_i$ denotes the ith stimulus vector and $\bar{r}_i$ denotes the ith corresponding response vector. (A top-down, left-to-right raster scan of an image yields such vectors.) 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 result will be the corresponding response vector $\bar{r}_k$. Specifically, we want to solve the following equation:

$$MS = R \tag{4}$$

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 that the system of equations is underconstrained. The constraint used to solve for a unique matrix M is that of minimizing the square error (i.e., min $\|MS - R\|^2$). The solution to the optimization (4) is given by

$$M = RS^+ \tag{5}$$

where S^+ is known as the Moore–Penrose generalized inverse of S [15]. The stimulus and response vector interaction should not be confused with simplistic stimulus-response associations because they interact with each other through M in a more complex way.

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

$$\hat{r} = M\hat{s}. \tag{6}$$

If the memorized stimulus vectors are independent and the unknown stimulus vector $\hat{s}$ 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 will contain the associated response vector and some *crosstalk* from the other stored response 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 vectors 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 thus allow some (learning) generalization of responses to previously unknown stimuli.

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

IV. Experiments

In this section we discuss the results of several computer simulations of our system. The computer simulation occurs in three phases: *construction*, *recall*, and *recognition*. In the construction phase associations that have to be memorized are used to construct the memory matrix. In the recall phase an unknown object 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 estimate the scale and rotation the object has gone through.

Images of objects are first preprocessed through the subsystem outlined in Section III and shown in Fig. 1. 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 as described in Section III-B. To perform a recall from the memory the unknown image is preprocessed by the same subsystem as the memorized images to produce the vectors $|\hat{\bullet}|_1$, $\hat{\Phi}_1$, $|\hat{\bullet}|_2$, and $\hat{\Phi}_2$. The resulting stimulus vector $|\hat{\bullet}|_1$ is projected onto the memory matrix to produce a response vector which is an estimate of the memorized phase vector Φ_2. The estimated phase vector $\hat{\Phi}_2$ and the magnitude vector $|\hat{\bullet}|_1$ are used to reconstruct the memorized object. The difference between the estimated phase $\hat{\Phi}_2$ and the phase $\bar{\Phi}_2$ is used to make an estimate of the rotation and scale the object in the image has undergone.

A. 2-D Bin-Picking Experiments

The database of images consists of twelve different 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. The f-stop of the camera was adjusted such that the contrast of the object did not saturate. It should be emphasized that all of the images used in creating and testing the memory system were taken at different times using various camera rotations and distances. The images are digitized to 256

× 256, eight bit quantized pixels, where the 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. The objects are centered within the frame by hand. This is probably the source of much of the noise and could have been done automatically using the object's center of mass. Also, each object that is used to construct the memory has its major axis vertical; this choice is made arbitrarily. The two-dimensional images (magnitude and phase) that are the output from the invariant representation subsystem are scanned horizontally to form the vectors stimulus $\bar{s} = |\bullet|$, $\bar{r} = \Phi_2$ for memorization. The image database used for these experiments is shown in Fig. 2, and we are concerned with recovering the identity of that object O_i which sits in the bin, possibly on top of some pile of other objects, and evoking a good quality image.

Fig. 3 displays the significant improvement observed when the input is highly corrupted by noise. Specifically, the recall operation, attempts to recover a key (α_1 = key) embedded in heavy noise (α_2 = Noise of SNR = −3 dB). The key is recognized and the recall's SNR (compared to the ideal recall) is equal to −0.9 dB.

Fig. 4 is an example of both occlusion and some geometric transformation of one of the objects in the database. The unknown object in this case is α_1 = leaf, which is rotated from the memorized leaf as given by the transformation T_1. A portion of the bottom was occluded by α_2. The problem is equivalent to recovering α_1 from the mix given by $T_1(\alpha_1) + \alpha_2$. The resulting reconstruction is very noisy but has filled in the missing part of the bottom of the leaf. The noisy recall is reflected in both the SNR and the interplay between memories shown by the histogram.

We consider in the next three examples the case of overlapping. Specifically, the generic case is given by $\Sigma\, T_i(O_i) + N$, i.e., objects O_i subject to some geometric transformation as given by T_i overlap, and the resulting image might be corrupted by some noise N due to the image formation process.

Fig. 5 is an example of overlapping a rotated key over a rotated leaf. The recall histogram vector picks up the key (item 7) and the leaf (item 11) as the top two best estimates. Fig. 6 is another example of overlapping two parts. Specifically, a rotated pin (item 1) sits on top of a rotated and scaled "S" curve (item 4). Finally, we considered the case where three parts overlap. Fig. 7 has a pin on top, an "S" in the middle and the key at the bottom. The parts were subject to transformations similar to the ones described above. The recall histogram again succeeds in identifying the correct objects as the pin (item 1), the "S" curve (item 4) and the key (item 7). The relative magnitude of the recall histogram elements is not necessarily a clear indication regarding the order of the elements in the pile. Additional factors such as relative size and centering should be considered as well. Further processing, in the form of some model-based production

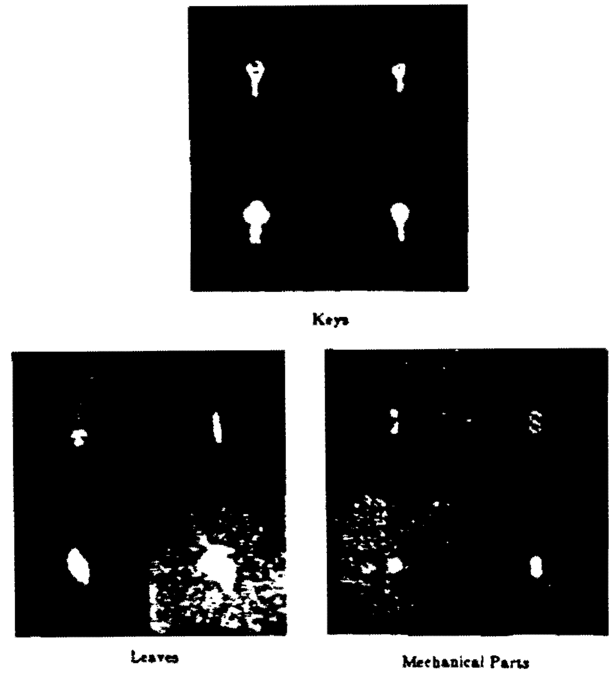

Fig. 2. Database of objects used in the experiments.

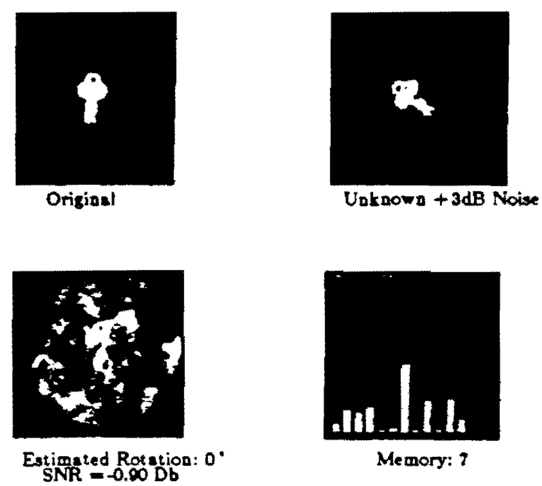

Fig. 3. Recall improves the SNR.

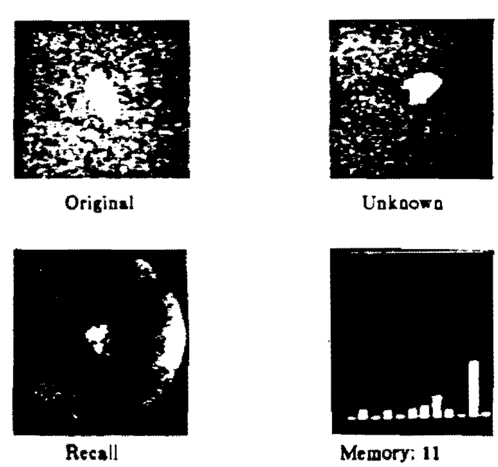

Fig. 4. Recall using a rotated leaf with occlusion.

Fig. 5. Overlapped key and leaf.

Fig. 6. Overlapped pin and "S".

Fig. 7. Overlapped Pin, "S", and leaf.

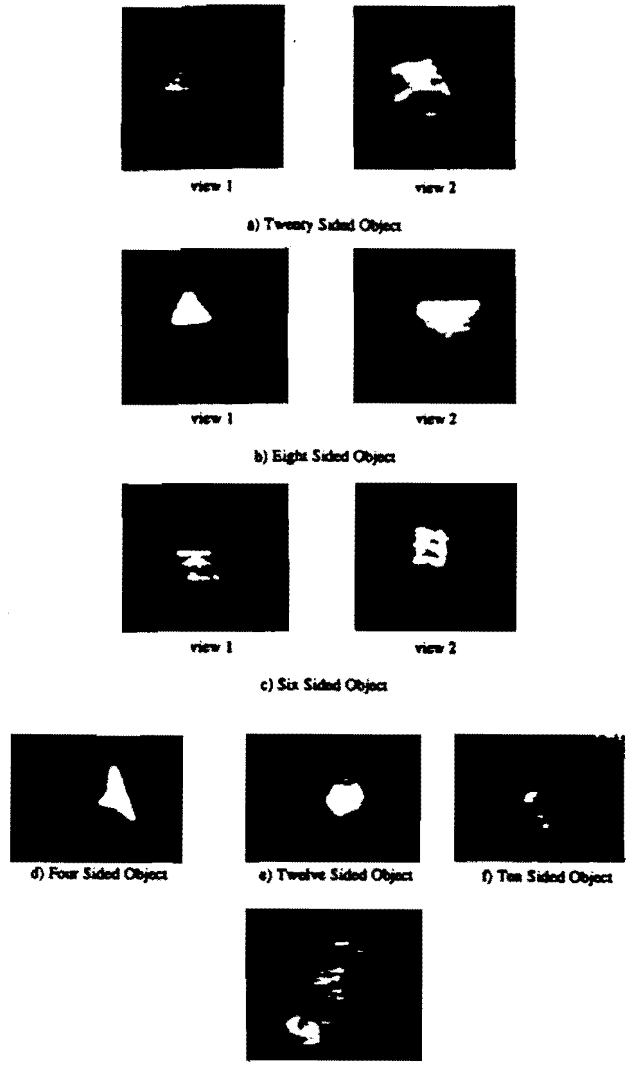

Fig. 8. Database for overlapped configuration.

rules based on considerations as given above could possibly identify the exact order of the elements.

B. Extensions for 3-D Pin-Picking

The next series of experiments explore the capability of our system when exposed to situations where the object's projected shape can change dramatically with viewpoint. The memory consists of six polyhedral objects. The (video) database used to construct this memory is shown in Fig. 8. Three of the objects [Fig. 8(a), (b), and (c)] are learned in both a side view and the standard top view. In this paradigm, recognition of the objects in three dimensions depends on matching the given projection of the object with one of the memorized characteristic views. The memory is tested using an image of three objects, which overlap as shown in Fig. 8(g), and diagramed in Fig. 9(e). In the test image the eight-sided object [Fig. 8(b)] is supported by the six-sided object [Fig. 8(c)] and the twenty-sided object [Fig. 8(a)]. The camera moves, similar to a conveyor belt, and fixates the overlapping objects in the test image at three distinct points shown in Fig. 9(e). Fig. 9 depicts several graphs which indicate how the recall histogram varies for a given object as the camera moves. The first three graphs show the response of the system to the objects used in the test image. For example, Fig. 9(a) displays three curves. The solid line corresponds to the response of the first view and the dashed line corresponds to the response of the second view of the twenty-sided object. The dotted line indicates the maximum response of the system to each object in the database. Note that the twenty-sided object had a maximal response for view one at sampling points 4 and 5. These points are near the center of the twenty-sided object in the test image. The results for all of the objects show that for sampling points close to the central locations of the objects the recognition is correct. For sampling points between objects the recognition response is dominated by one of the objects present in the input and close to the fixation point.

The previous example emphasizes two results. First, the experiment demonstrates the ability of the system to operate on clustered objects. The object views are memorized in isolation, so it is not immediately apparent that the system can recognize individual objects within a group of objects. Second, changing viewpoints leads to dramatic changes in the projected image of a three-dimensional ob-

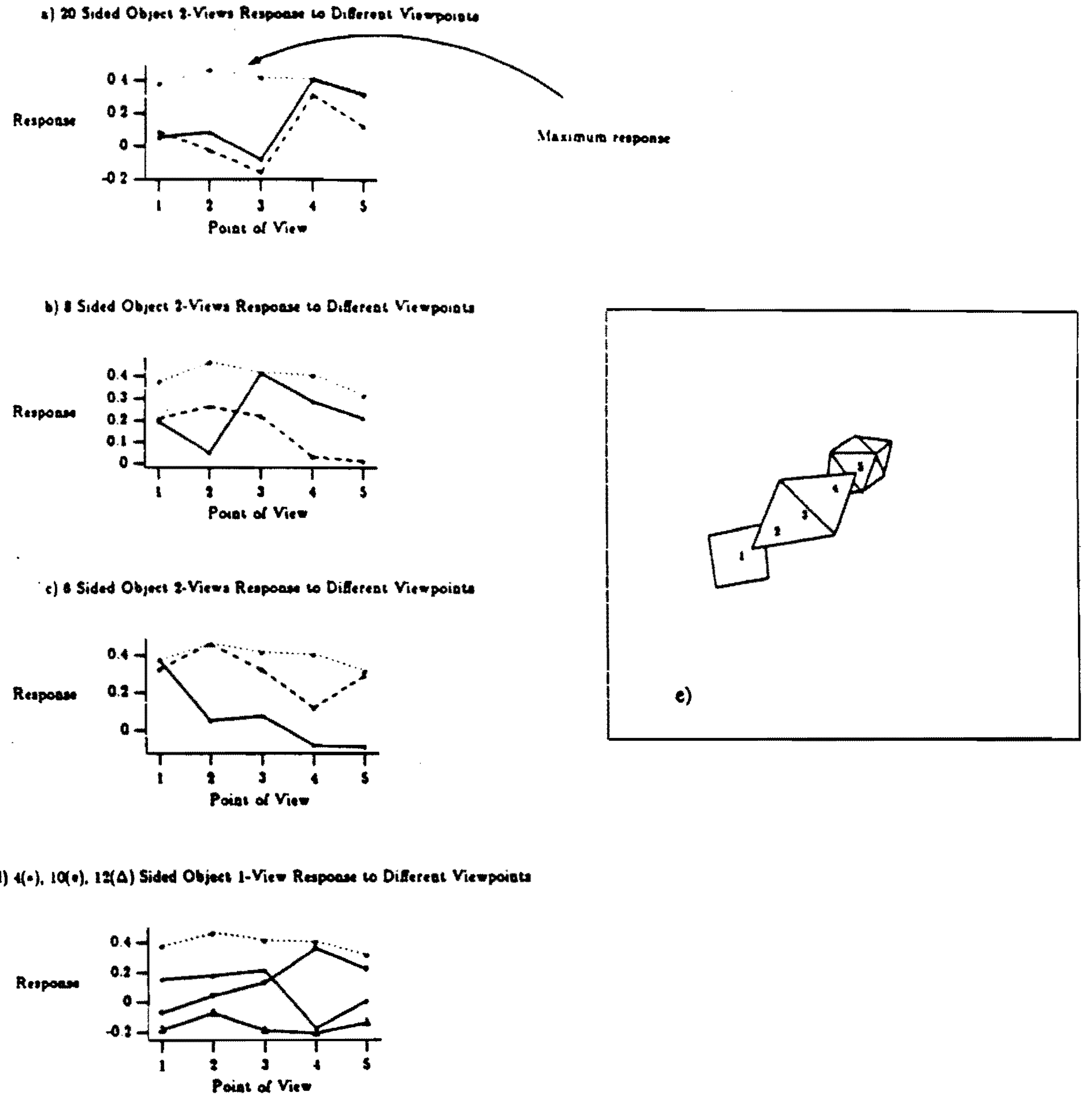

Fig. 9. Recognition results for overlapped configuration.

ject. The experiment shows that at least for simple polyhedral objects, recognition can be carried out using multiple views.

V. Conclusions

We have suggested in this paper the DAM for bin-picking and showed the feasibility of such an approach. The system developed displays invariance to sensor noise, geometric distortions (scale, rotation, misalignments) and occlusion/overlap. Furthermore, extensions to the system allow the 3-D bin-picking problem to be addressed. There are some weaknesses in the Neural Network (NN) model we have chosen for the heart of the recognition system. 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 nonlinear element, such as a sigmoid function, is introduced into the feedback loop. Nonlinear neural networks 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 for future research.

The DAM, like most other NN models, does not exploit the topographical power naturally present in input visual information. Examinations of the visual cortex have shown that visual information is processed through multiple mappings of the visual field. 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 than the ability 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 learning methods.

The computer vision system presented in this paper was designed with only two-dimensional geometric distortions in mind. We have shown some ability to deal with clustered three-dimensional polyhedra as presented in the previous experiments. We are presently extending our work toward three-dimensional object recognition. We propose to use an approach based on characteristic views [7] or aspects [14] 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] N. Ayache and O. D. Faugeras, "HYPER: A new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. PAMI-8, no. 1, pp. 44-54, 1986.

[2] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[3] B. Bhanu and O. D. Faugeras, "Shape matching of two-dimensional objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 2, pp. 137-155, 1984.

[4] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psych. Rev.*, vol. 94, no. 2, pp. 115-147, 1987.

[5] R. C. Bolles and R. A. Cain, "Recognizing and locating partially visible workpieces," in *Proc. IEEE Comput. Soc. Conf. Pattern Recognition and Image Processing*, Las Vegas, NV, 1982, pp. 498-503.

[6] D. Casasent and D. Psaltis, "New optical transforms for pattern recognition," *Proc. IEEE*, vol. 65, no. 1, pp. 77-84, 1977.

[7] I. Chakravarty and H. Freeman, "Characteristic views as a basis for 3-D object recognition," in *Proc. SPIE Robot Vision*, vol. 336, 1982, pp. 37-45.

[8] R. T. Chin and C. R. Dyer, "Model-based recognition in robot vision," *Comput. Surveys*, vol. 18, no. 1, pp. 67-108, 1986.

[9] J. L. Crowley, "Machine vision: Three generations of commercial systems," Robotics Inst., Carnegie-Mellon Univ., Rep. CMU-RI-TR-84-1, 1984.

[10] L. S. Davis and A. Rosenfeld, "Cooperating processes for low-level vision: A survey," *Artificial Intell.*, vol. 17, no. 1-3, pp. 245-263, 1981.

[11] J. D. Dessimoz, J. R. Birk, R. B. Kelley, H. A. S. Martins, and C. L. I, "Matched filters for bin-picking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 6, pp. 686-697, 1984.

[12] W. E. L. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range data," in *Techniques for 3-D Machine Perception*, A. Rosenfeld, Ed. Amsterdam, The Netherlands: North-Holland, 1986, pp. 113-148.

[13] B. K. P. Horn and K. Ikeuchi, "The mechanical manipulation of randomly oriented parts," *Sci. Amer.*, pp. 100-111, Aug. 1984.

[14] 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.

[15] T. Kohonen, *Self-Organization and Associative-Memories*. New York: Springer-Verlag, 1984.

[16] D. G. Lowe, *Perceptual Organization and Visual Recognition*. Boston, MA: Kluwer Academic, 1985.

[17] L. Massone, G. Sandini, and V. Tagliasco, "'Form-invariant' topological mapping strategy for 2D shape recognition," *CVGIP*, vol 30, pp. 169-188, 1985.

[18] W. A. Perkins, "A model based vision system for industrial parts," *IEEE Trans. Comput.*, vol. C-27, no. 2, pp. 126-143, 1978.

[19] 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.

[20] L. G. Roberts, "Machine perception of three-dimensional solids," in *Optical and Electro Optical Information Processing*, J. T. Tippet, Ed. Cambridge, MA: MIT Press, 1965.

[21] J. Sklansky, "On the Hough technique for curve detection," *IEEE Trans. Comput.*, vol. C-27, no. 10, pp. 923-926, 1978.

[22] J. L. Turney, T. N. Mudge, and R. A. Voltz, "Recognizing partially occluded parts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 4, pp. 410-421, 1985.

[23] S. Ullman, "Against direct perception," *Behavioral Brain Sci.*, vol. 3, pp. 373-415, 1980.

[24] H. Wechsler and G. L. Zimmerman, "2-D invariant object recognition using distributed associative memory," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 6, pp. 811-821, 1988.

[25] C. F. R. Weiman, "Logarithmic spiral grids for image processing and display," *CGIP*, vol. 11, pp. 197-226, 1979.

[26] H. S. Yang and A. C. Kak, "Determination of the identity, position and orientation of the topmost object in a pile," in *Proc. IEEE Third Workshop Computer Vision: Representation and Control*, Bellaire, MI, 1985.

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

He is presently a Professor of Computer Science with George Mason University, Fairfax, VA. His research is in computer vision, AI, and neural networks. His major projects have been involved with: the introduction and use of joint space/spatial-frequency representations for optical flow derivation and texture analysis (grouping phenomena); fault-tolerant object recognition using distributed associative memory (DAM); and knowledge-based (KB) information systems development using distributed representation and computation. He has authored one book, over 60 scientific papers, and is presently writing a book on *Computational Vision* to be published by Academic Press in 1990. He has consulted for both government and private industry, lectured extensively in the U.S. and abroad, and held visiting academic positions in France (INRIA), Japan (Kyoto), and Israel (Technion).

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, Salt Lake City, 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.