ELSEVIER

Artificial Intelligence 78 (1995) 461-505

Artificial Intelligence

An active vision architecture based on iconic representations [☆]

Rajesh P.N. Rao, Dana H. Ballard

Department of Computer Science, University of Rochester, Rochester, NY 14627, USA

Received July 1994; revised March 1995

Abstract

Active vision systems have the capability of continuously interacting with the environment. The rapidly changing environment of such systems means that it is attractive to replace static representations with visual routines that compute information on demand. Such routines place a premium on image data structures that are easily computed and used.

The purpose of this paper is to propose a general active vision architecture based on efficiently computable iconic representations. This architecture employs two primary visual routines, one for identifying the visual image near the fovea (object identification), and another for locating a stored prototype on the retina (object location). This design allows complex visual behaviors to be obtained by composing these two routines with different parameters.

The iconic representations are comprised of high-dimensional feature vectors obtained from the responses of an ensemble of Gaussian derivative spatial filters at a number of orientations and scales. These representations are stored in two separate memories. One memory is indexed by image coordinates while the other is indexed by object coordinates. Object location matches a localized set of model features with image features at all possible retinal locations. Object identification matches a foveal set of image features with all possible model features. We present experimental results for a near real-time implementation of these routines on a pipeline image processor and suggest relatively simple strategies for tackling the problems of occlusions and scale variations. We also discuss two additional visual routines, one for top-down foveal targeting using log-polar sensors and another for looming detection, which are facilitated by the proposed architecture.

[☆] This work is supported by NSF research grant no. CDA-8822724, NIH/PHS research grant no. 1 R24 RRO6853, and a grant from the Human Science Frontiers Program.

1. Introduction

Vision in humans is not a passive open-loop image analysis task but rather one that involves a continuous interaction with the world. This interaction is characterized by the relatively frequent use of *saccades*. These discrete eye movements, which can achieve speeds of up to 700° per second, occur at the rate of about three per second. Experiments have shown that saccades are intimately related to the subject's momentary problem solving strategy [6,23,73].

The central role of eye movements in human vision has inspired an extensive modeling effort in computer vision aimed at elucidating the technical advantages of binocular camera systems capable of similar movements. The effort has been variously called *active, animate, task-oriented, dynamic, inexact* or *behavioral vision* [2,4,5,22,55,66, 74]. Such models require the ability to handle the enormous data rates associated with real-time video images. Only recently has this become possible with the availability of powerful and yet cheap hardware for real-time image processing.

Much work in passive vision, such as analysis of photographs of visual scenes, has taken advantage of extensive models that reconstruct the contents of the scene. Such a strategy has worked as the venue typically does not change and therefore there is no pressing need to process the image quickly other than that of economy. However, in active vision systems which exploit eye movements to solve complex tasks in real time, the input image changes much too frequently to allow elaborate reconstruction. For example, in human vision, fixations between saccades usually last only 300 milliseconds. Recent experiments indicate that the complexity of information retained from image to image between saccades is limited [27]. There is thus tremendous incentive to avoid the need for expensive reconstruction of the visual world.

An alternative to reconstruction and a strategy which we feel is especially attractive in the context of active vision systems is that of composing task-specific behaviors from a relatively simple set of *visual routines* [77] that compute information on demand. Such a strategy is based on the hypothesis that *vision is functional*, thereby necessitating the existence of a mechanism for spanning the space of task-dependent representations [6]. A primary motivation for visual routines comes from an appreciation of the fact that most natural human behaviors are very stereotyped and occur in limited ranges of workspace. Thus, for a significant number of behaviors, one can assume that the visual targets will be imaged in approximately the same viewing conditions as when they were initially remembered. This allows the use of simple task-directed programs that operate directly on the optic array in image coordinates. Such routines require only retinotopically indexed visual representations that can be computed quickly and which are required to be only moderately insensitive to variations in the viewing direction.

Additional motivation for visual routines comes from biological studies [48,50,78] which seem to suggest that the primate visual cortex is roughly organized into separate specialized modules (located in the parietal and temporal lobes) subserving the two complementary functions of directing eyes to new target locations and analyzing the currently foveated area. In such a setting, eye movements can be thought of as solving a succession of location and identification subtasks in the process of meeting a larger cognitive goal. This observation suggests a useful hierarchical decomposition of visual

Table 1
A hierarchical decomposition of visual behaviors related to scene interpretation. Visual problem solving behaviors can be viewed as arising from task-specific sequential composition of the simpler "What" and "Where" behaviors, which in turn rely on the more primitive ability of comparing a single model with a single image component (adapted from [5])

	Stored models	
Image components	*One*	*Many*
One	Comparing a single model with a single image component.	Identification: Recovering the identity of an object whose location is currently being fixated.
Many	Location: Finding a known object in the current image.	General visual computation.

behaviors involved in scene interpretation, or the general problem of relating internal models to external objects (Table 1). The "What" component corresponds to the problem of identification of a foveated point and the "Where" component corresponds to the problem of locating a point of interest in the current image.[1] This reduction to the complementary problems of location and identification decreases the complexity of the interpretation problem enormously. In the location task, only one model is present and in the identification task, only one location is present. Both location and identification behaviors in turn rely on the more primitive ability to compare a single model with a single image component and can be regarded as being composed of a sequence of such elementary comparisons. Complex visual behaviors can then be viewed as arising from different *task-specific sequential compositions* of the simpler "What" and "Where" behaviors.

The purpose of this paper is to propose an active vision architecture based on the hierarchical decomposition of visual behaviors discussed above. The architecture uses two visual routines, one for identifying the visual image near the fovea, and another for locating a stored prototype on the retina. The two routines are subserved by two separate memories as shown in Fig. 1. One memory is indexed by image coordinates as depicted by the box on the left-hand side of the figure. The other is indexed by object coordinates as depicted by the box on the right-hand side of the figure. Object location matches a localized set of model features with image features at all possible retinal locations. The result is the image coordinates of the best match. Object identification matches a foveal set of image features with all possible model features. The result is the model coordinates of the best match.

The central representation of the architecture is a high-dimensional iconic[2] feature vector comprised of the responses of different order derivatives of Gaussian filters (which are *steerable* [25]) at a range of orientations and scales. Such a feature vector serves as

[1] An important distinction for biological modelers to note in this context is that our use of "What" and "Where" does not imply that our processes use solely the infero-temporal and parietal cortices respectively. Our location routine uses both areas but the result is nominally in the parietal cortex. Our identification routine likewise uses both areas but the result is nominally in the infero-temporal cortex.

[2] Our use of the term *iconic* is analogous to its use by Nakayama [53] to describe small visual templates which constitute visual memory.

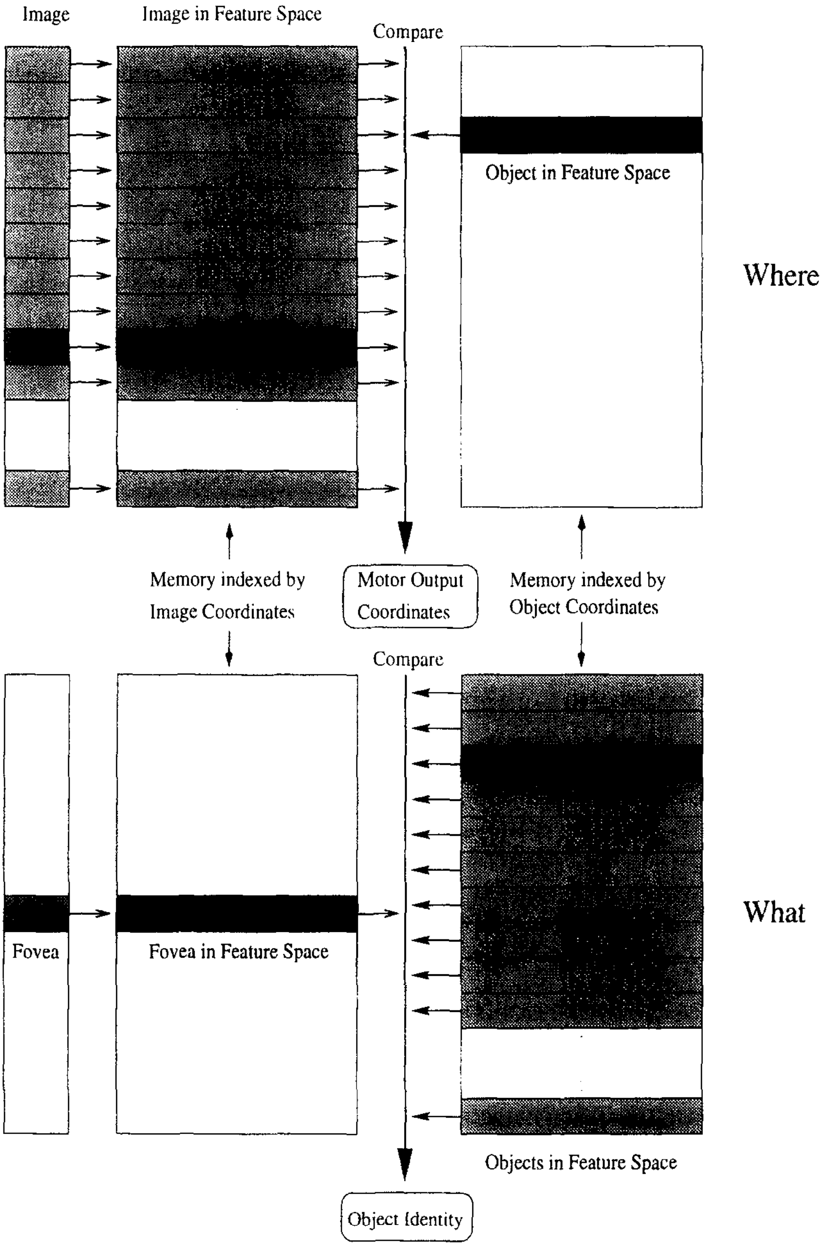


Fig. 1. The proposed active vision architecture. The architecture uses two primary visual routines and a common iconic representation. (Upper) To locate an object, its features are matched against retinal features at similar locations. The result is the location of the object in retinal coordinates. This may be augmented by a depth term obtained from binocular vergence. (Lower) To identify an object, the features near the fovea are matched against a data base of iconic models, also encoded in terms of features. The result is decision as to the object's identity.

an effective photometric description of the local intensity variations present in the image region about a scene/object point. In its most general form, the n-element feature vector is comprised of the responses of m oriented basis filters at k different scales ($n = m \times k$); for the experiments in this paper, 9 basis filters were used at 5 octave-separated scales to yield a 45-element iconic index for the local image patch near a given object point. The relatively large number of measurements at an image point makes its characteristic vector practically unique due to the orthogonality inherent in high-dimensional spaces (Section 2.1). In Section 3.1, we describe a simple normalization procedure that makes the vector invariant to rotations about the view vector. For rotations about axes other than the viewing axis, the success of the descriptors depends on their moderate view insensitivity. Our experiments indicate that the filters are insensitive to three-dimensional rotations of up to 25° at radial distances of 5 feet or more from the focal point. More drastic rotations are handled by storing feature vectors from different views as described in Section 4.5.1.

While multiscale filters have previously been applied to solve a multitude of classical early vision tasks (Table 2), their utility in active vision systems has remained largely unexplored. This paper helps to fill this void by demonstrating their usefulness in facilitating a number of important visual behaviors. We first describe visual routines that employ iconic representations of objects to solve the two primary problems of object identification (Section 4.1) and object localization (Section 4.2); experimental results for these routines with complex objects in realistic scenes are presented in Section 4.5. The multiscale iconic representations also allow an active vision strategy for handling partial occlusion near a scene point; the method, described in Section 4.6, uses a description of the occluder in the form of a template which can be obtained via active vision systems. Section 4.7 describes the use of the representation in a method for achieving top-down guidance of the fovea during visual search using log-polar sensors. Section 4.8 illustrates how the multiscale structure of the representation can be further exploited to handle the problem of variations in the scale of an object. The solution leads naturally to a simple implementation of a reflex for detection of looming objects. Finally, Section 5 touches on the issue of incorporating color information into the current representation and summarizes important aspects of our approach including comparisons with some recently proposed recognition strategies that also employ iconic object descriptions.

2. Iconic representations

The purpose of a representation medium in active vision systems is simply to "associate features with an object that makes behaviors concerning it especially easy to execute" [5]. The representation must (a) allow fast execution of the various visual routines, (b) provide enough information for directing gaze at required scene points, and (c) facilitate the development of visuo-motor learning algorithms. The above requirements imply that the visual features comprising the representation must first of all be computable in real time rather than be the result of elaborate multi-stage processes. In addition, the representation must be rich and robust enough to allow proper functioning in the presence of noise in both internal and external channels. Finally, the representation must take into account the fact that elementary scene features occur at a variety of scales and orientations. Our representation achieves the above objectives by assembling the responses of steerable Gaussian derivative filters at a number of orientations and scales into a single high-dimensional iconic feature vector.

2.1. The favorable properties of high-dimensional vectors

Pentti Kanerva was among the first to realize the advantages of high-dimensional vectors as a representation medium. In [36], he convincingly demonstrates the usefulness of high-dimensional binary vectors in formulating a sparse distributed memory system that mimics the human memory system in many ways.

By using the normal distribution with mean $n/2$ and standard deviation $\sqrt{n}/2$ to approximate the binomial distribution of the Hamming distances between an arbitrary vector and all other vectors of the space $\{0,1\}^n$ (for large n), it is easy to show that most of the vectors in the space are orthogonal (or "indifferent") to any given vector; in other words, most of the vectors lie at approximately the mean distance $n/2$ from a given vector, with only a minute fraction closer or further away. Thus, *a representation of an object of interest in the form of a high-dimensional vector can be subjected to considerable noise before it is confused with the vectorial representation of other objects.* As a result, when trying to determine whether a particular vector belongs to a specific class of similar vectors, the likelihood that it does increases rapidly as the vector approaches one of the vectors from the class only slightly [7].

The same result holds true for non-binary vectors such as the iconic representations presented in this paper. The components of these iconic feature vectors in our current implementation belong to the set $A = \{-128, \ldots, 127\}$ and span an extremely large vector space consisting of $A^{m \times k} = 256^{45}$ points where k is the number of scales used and m the number of basis filters per scale. Fig. 5(c) shows the distribution of distances measured in terms of correlations (i.e. normalized dot products) in this space between the 45-dimensional feature vector for a given model point (marked by a "+" in Fig. 5(a)) and those for 220,268 other unrelated points in a natural scene. This distribution of distances has a mean $\mu = 0.037$ with a standard deviation $\sigma = 0.263$. Most of the points of this space lie within two standard deviations of the mean and are thus effectively indifferent (correlation $\simeq 0.0$) to the given model point.

2.2. The link to visual memory

There has been recent evidence [24] that the primate visual system takes advantage of the redundancy in the visual environment by producing a sparse distributed coding that aims to minimize the number of simultaneously active cells. This finding seems to suggest a close relationship between the goal of visual coding and subsequent storage in an associative distributed memory.

Summarizing his views on the sensory interface with memory, Kanerva notes [36, p. 119]:

> The memory works with features and creates internal objects and individuals by chunking together things that are similar in terms of those features. In order for those internal objects to match objects of the world, the system's sensors must transform raw input from the world into features that are relatively invariant over small perturbations of objects.

and concludes by asserting that

> ... artificial intelligence methods need to be augmented with mathematical and statistical methods of dealing with representations in high-dimensional spaces.

In this regard, representations in the form of iconic feature vectors may be considered an effective medium for vision-related memory storage since they are both relatively invariant to minor disturbances in the environment and at the same time retain the attractive matching properties of high-dimensional vectors. Hypothetically, the visual memories could consist of these iconic representations stored in a distributed manner along the lines suggested by Kanerva. Stored representations may then be activated by either incoming visual signals or by other iconic representations to which they are associatively linked; this activation would be mediated by specific visual routines. "Visual perception" for the agent then becomes synonymous with this activation of memory. The visual memory would form part of a larger memory holding sensori-motor programs for a wide range of behaviors. Such a memory may play a crucial role in the design of anthropomorphic systems capable of complex visuo-motor behaviors. A modest start in establishing a link between visual representations and visual associative memory is made in [64] (see also [63]) where the idea of using iconic feature vectors for indexing into a visual memory based on Kanerva's sparse distributed memory is explored.

2.3. *Basis functions from natural images*

Unlike random collections of pixels, images of natural scenes are characterized by a high degree of statistical regularity. For instance, pixel values in a given neighborhood tend to be highly correlated owing to the morphological consistency of objects. Thus, a pixel-wise representation of objects obtained from a camera is highly redundant and some form of redundancy reduction is desirable.

The optimal linear method (in the mean squared error sense) for reducing redundancy is the Karhunen-Loéve transform or eigenvector expansion via Principal Component Analysis (PCA) (see [15] for an introduction). PCA generates a set of orthogonal axes of projections known as eigenvectors or *principal components* of the input data distribution in the order of decreasing variance. The eigenvectors form a set of orthogonal basis functions for representing the input. Given n input images, all n eigenvectors of the input distribution are required in principle to completely represent the input image set but due to the statistics of natural images, it is usually the case that only a small number m of eigenvectors ($m \ll n$) account for almost all of the variance in the input data. Thus, by using only the first m dominant eigenvectors as basis functions (or orthogonal axes) for projecting inputs, considerable computational savings can be achieved.

Table 2
The trend from variational methods towards the use of filters for solving specific problems in computer vision

Computing methodology	Problem being solved	References
Calculus of variations	Optic flow	[32]
	Shape from shading	[33]
Filters at single scale	Brightness edge detection	[10,31]
	Curved line grouping	[45,57]
Filters at multiple scales	Optic flow	[1,30,80]
	Shape from shading	[58,59]
	Texture segmentation	[40,46]
	Stereo correspondence	[35,38]
	Scene interpretation	[11,82]
	Biometric signatures	[19]

Recent work by Turk and Pentland [76] employed PCA to synthesize the eigenvectors ("eigenfaces") of a training set of face images; face recognition was achieved by using a template-matching strategy with the vectors obtained by projecting input face images along a small number of eigenfaces. Murase and Nayar [51] have used a similar approach for solving the general problem of object recognition and pose estimation. Both these methods require recomputation of the eigenvectors when new faces/objects are encountered. It is therefore natural to ask what the results of PCA would be if one were to take the above process to its limit i.e. to perform PCA on a set of arbitrary natural images containing a variety of natural and man-made stimuli. Hancock et al. [29] shed some light on this interesting question by using a neural network introduced by Sanger [67] to extract the first few principal components of an ensemble of natural images. They discovered that *regardless of the scale of analysis, the eigenvectors obtained were very close approximations of different oriented derivative-of-Gaussian operators.*

2.4. Iconic representations from Gaussian derivative filters

There has been a recent surge of interest in the use of multiscale spatio-temporal filters for proto-visual analysis and as computing machinery for solving complex vision-related problems which had previously been tackled by variational methods. This historic trend is depicted in Table 2.

Here, we focus on the class of Gaussian derivative filters. By employing an ensemble of these linear spatial filters at a variety of orientations and scales, an iconic representation of an image region can be obtained. This iconic representation compromises on the ideal of strict view invariance. Instead, image features are judged useful even if they are only relatively insensitive to variations in view. One example of such a feature is image color as a measure of surface albedo [70,71]. The photometric feature vectors obtained by using Gaussian derivative filters behave very much like color in that they are tolerant to modest variations in view.

The choice of Gaussian derivative filters in our iconic representation is motivated by the underlying belief that these filters form an ideal set of *natural basis functions* for *general-purpose object recognition*. Part of the rationale for this belief stems from the fact that these basis functions are obtained as a result of applying the principle of dimensionality-reduction as embodied by PCA (see previous section) to large collections of images containing a plethora of elementary features from natural as well as man-made structures rather than just the training images of particular objects or faces as was the case for the basis functions used in [51,76]. Further support for this belief comes from the observation that correlation filters generated by principal component expansion are statistically optimal in the sense that they maximize signal-to-noise ratio and yield much sharper correlation peaks than traditional raw image cross-correlation techniques (see, for instance, [43]). Indeed, Canny [13] has shown the first- and second-order Gaussian derivatives to be close to optimal for detecting the elementary features of edges and bars respectively. Finally, the Gaussian derivatives filters allow strategies for rotation normalization in the image plane because they are known to be *steerable* [25].

2.4.1. Steerable filters

Steerable filters are a set of oriented basis filters that have the important property that the response of a filter at an arbitrary orientation can be synthesized from linear combinations of the basis filters. It is well known [25] that using a circularly symmetric Gaussian function in Cartesian coordinates (with scale and normalization factors set to 1 here for notational convenience),

$$G(x,y) = \mathrm{e}^{-(x^2+y^2)}, \tag{1}$$

we can define two first-order basis filters G_1^0 and $G_1^{\pi/2}$ as:

$$G_1^0 = \frac{\partial}{\partial x}G(x,y) = -2x\mathrm{e}^{-(x^2+y^2)} \tag{2}$$

and

$$G_1^{\pi/2} = \frac{\partial}{\partial y}G(x,y) = -2y\mathrm{e}^{-(x^2+y^2)}, \tag{3}$$

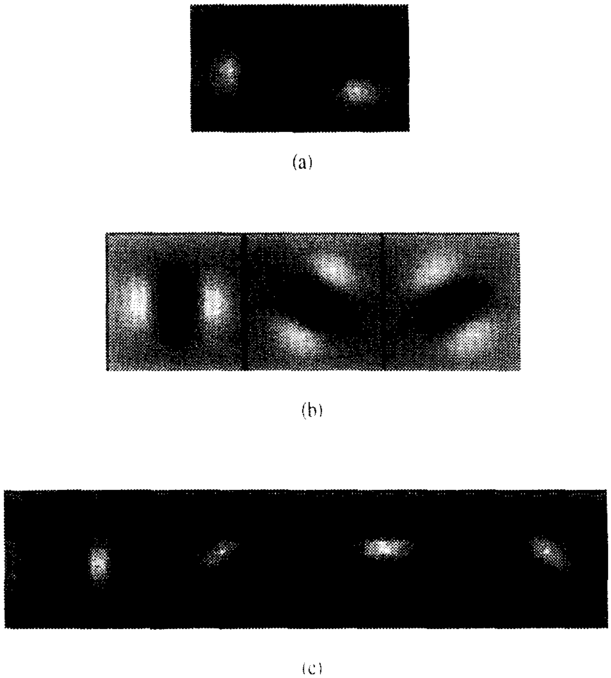
such that the directional derivative in an arbitrary direction θ can be synthesized as:

$$G_1^\theta = \cos(\theta)G_1^0 + \sin(\theta)G_1^{\pi/2}. \tag{4}$$

Higher-order filters at arbitrary orientations can be synthesized analogous to the first-order case by using basis functions denoted by:

$$G_n^{\theta_n}, \qquad n = 1,2,3, \quad \theta_n = 0,\ldots,k\pi/(n+1), \quad k = 1,\ldots,n, \tag{5}$$

where n denotes the order of the filter and θ_n the orientation of the filter. Fig. 2 shows these filters for a particular scale. Different interpolation functions are needed to *steer* the different order Gaussian filters, as shown by Freeman and Adelson [25]. The number of interpolation functions that are needed for steering is one more than the filter order. So, for example, the first-order filters can be steered with two interpolation functions



(a)

(b)

(c)

Fig. 2. The impulse response of nine oriented Gaussian derivative basis filters of up to the third-order (shown here at an arbitrary scale). (a) G_1; (b) G_2; (c) G_3. (Bright regions denote positive magnitude while darker regions denote negative magnitude.) In our implementation (Section 4.4), discrete versions of these filters are used in the form of 8 × 8 kernels for convolutions with a five-level low-pass pyramid of the input image.

given basis measurements at 0° and 90°, the second-order filters can be steered with three functions given basis measurements at 0°, 60°, and 120°, and so on. In particular,

$$G_n(\theta) = \sum_{i=1}^{n+1} G_n^{(i-1)\pi/(n+1)} k_{in}(\theta), \tag{6}$$

where the first-order interpolants ($n = 1$) are given by,

$$k_{i1}(\theta) = [\cos(\theta - (i-1)\pi/2)], \quad i = 1, 2. \tag{7}$$

For $n = 2$, we have

$$k_{i2}(\theta) = \tfrac{1}{3}[1 + 2\cos(2(\theta - (i-1)\pi/3))], \quad i = 1, 2, 3, \tag{8}$$

and for $n = 3$,

$$\begin{aligned} k_{i3}(\theta) = \tfrac{1}{4}[2\cos(\theta - (i-1)\pi/4) + 2\cos(3(\theta - (i-1)\pi/4))], \\ i = 1, 2, 3, 4. \end{aligned} \tag{9}$$

3. The multiscale iconic index

The feature index used in our architecture uses Gaussian derivative filters of up to third order. For maintaining steerability of up to this order, at least nine basis filters need to be used as described in the previous section. Larger numbers of basis filters could be used in order to counter the effects of noise but for the purposes of this paper, we retain the minimal basis set of nine filters. The response of an image patch I centered at (x_0, y_0) to a particular basis filter $G_i^{\theta_j}$ is obtained by convolving the image patch with the filter:

$$\begin{aligned} r_{i,j}(x_0, y_0) &= (G_i^{\theta_j} * I)(x_0, y_0) \\ &= \iint G_i^{\theta_j}(x_0 - x, y_0 - y) I(x, y) \mathrm{d}x \, \mathrm{d}y. \end{aligned} \tag{10}$$

This results in nine independent photometric measurements at each image point.

Further information regarding the image region centered at that image point is obtained by using the filters at different image scales. Since there are nine measurements per scale, there are $9 \times k$ total measurements where k is the number of scales. For the experiments, five different scales were used, for a total of forty-five measurements per point.

The different responses at different scales are sensitive to the width of the templates, so the responses, to be comparable across scales, have to be normalized. As shown in, for example, [46], the easiest way to do this is to divide by the filter energy defined as:

$$e_i = \iint G_i^0(x, y)^2 \mathrm{d}x \, \mathrm{d}y, \quad i = 1, 2, 3. \tag{11}$$

Now define the normalized response of a set of filters to the area surrounding a specific point in the image as the vector

$$\boldsymbol{r} = (r_{i,j,k}), \qquad i = 1, 2, 3, \quad j = 1, \ldots, i + 1, \quad k = s_{\min}, \ldots, s_{\max}, \tag{12}$$

where $r_{i,j,k}$ denotes the response of a filter with i denoting the order of the filter, j denoting the number of filters per order, and k denoting the number of different scales. The iconic index $\mathbf{r}$ thus obtained effectively serves as a photometric description of the local intensity variations present in the image region near the image point at which the filters were applied.

3.1. Rotation normalization and view insensitivity

An attractive property of Gaussian derivative filters is their steerability (Section 2.4.1). In this section, we exploit this property to obtain a simple method for making the iconic object indexes described above invariant to rotations about the viewing axis (assuming the scale remains relatively unchanged).

First, select the orientation of the first-order filters as a reference. This is a good strategy for two reasons: (1) the orientation can be computed directly from the filter responses, and (2) these filter responses are usually the most stable.

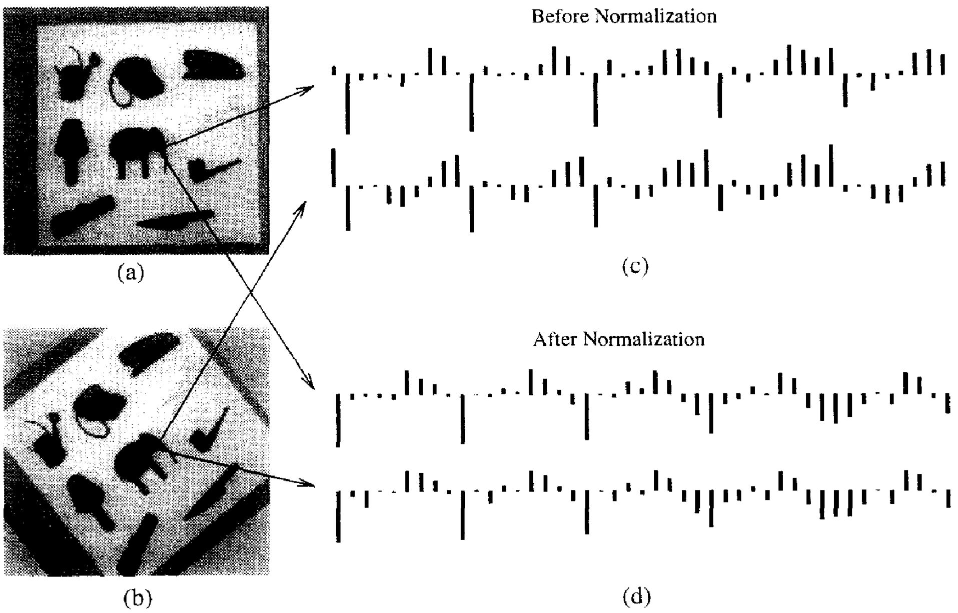


Fig. 3. Rotation normalization. (a) A test image; (b) the same image rotated 38° counterclockwise; (c) the filter response histograms for corresponding points near the elephant's mouth in the two images before normalization; (d) the response histograms after normalization. (Positive responses are represented by upward bars proportional to the response magnitude and negative ones by proportional downward bars with the nine smallest scale responses at the beginning, the nine largest ones at the end and the intermediate scales in between in the order of increasing scale.)

Given a vector of raw filter responses, the current orientation can then be computed as:

$$\alpha = \arctan 2(r_{1,2,s_{\max}}, r_{1,1,s_{\max}}). \tag{13}$$

The filter responses are then rotated using the steering formulae to a canonical direction (horizontal) to obtain the normalized responses:

$$r'_{i,j,k} = \sum_{j'=1}^{i+1} r_{i,j',k} k_{j'i}(\alpha + \theta_j), \qquad i = 1,2,3, \quad j = 1,\ldots,i+1, \quad k = s_{\min},\ldots,s_{\max}. \tag{14}$$

Note that this normalization makes the matching process more powerful than that produced with rotation invariant templates. The latter sacrifice variability in the angular direction. Instead the filters capture the variations in angle, and preserve it in their components. Another feature of the normalization process is that it can be done without additional convolutions since the operation of convolution is linear; the interpolation properties of the existing filters allow it to be carried out with a single basis set of convolutions as given by Eq. (14). Fig. 3 illustrates the rotation normalization procedure. It

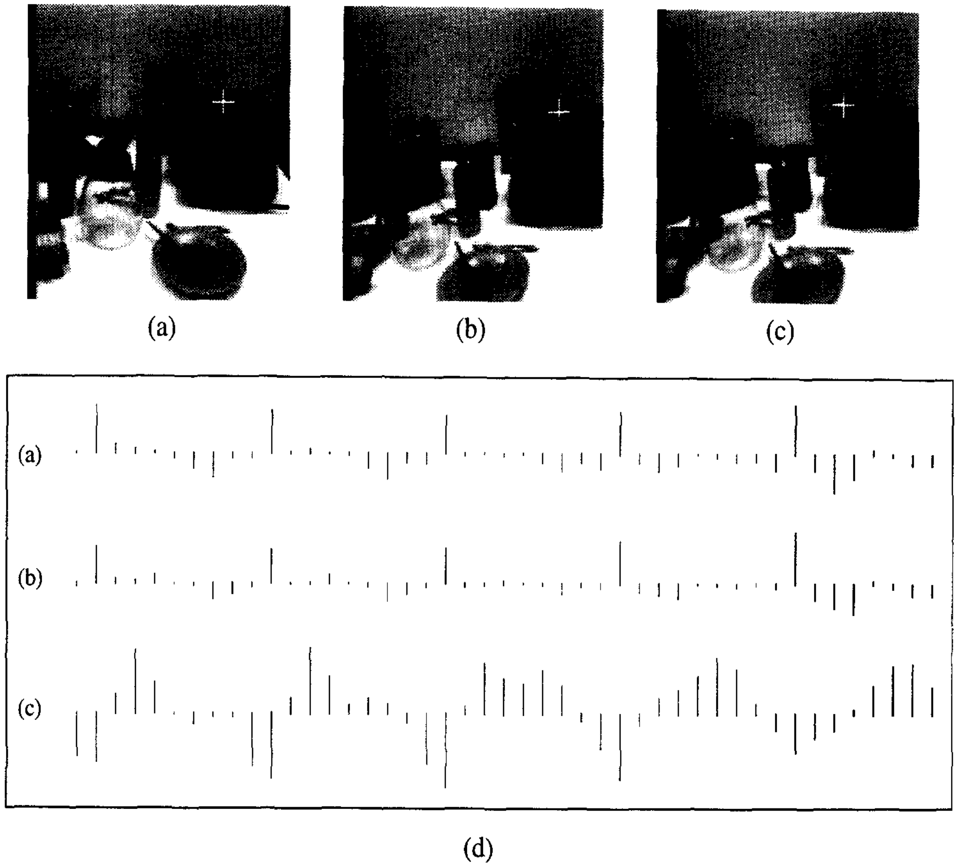


Fig. 4. Tolerance to modest view variations. (a) A point on the original image; (b) the same point correctly located by the algorithm in a second image with a 22.5° 3D rotation; (c) an unrelated point in the rotated image for the purpose of comparison; (d) the 45 filter responses (in the form of response histograms) for the point in (a) (top), the point in (b) (middle) and the point in (c) (bottom).

is clear that the procedure has rendered the two relatively uncorrelated model response vectors of the same point to be almost identical.

Rotations about an image plane axis are ameliorated in two ways. First, the reliance on a large number of responses renders the multiscale index robust to changes in the responses of a few individual filters caused by the geometric effect of change in viewing position. More importantly, the filter responses are dominated by a cosine envelope, so that there is a useful range of rotations for which the responses will be effectively invariant. This fact is illustrated in the example shown in Fig. 4. A more complicated scheme would be to make use of learned geometric distortion estimates such as the one proposed by Kass [39] to dynamically adjust the relative weighting of each filter as a function of its sensitivity profile. Drastic rotations are handled by storing feature vectors from different views as described in Section 4.5.1.

Table 3
Sensitivity of the match value to the length of the iconic vector (= number of scales used × nine). The figures shown are the average of the results for three pairs of corresponding points

Number of filters	Rank of matching point	Difference in distance
9	18.3	−8.1
18	4.3	−6.3
27	1.3	1.0
36	1	5.6
45	1	10.8

3.1.1. The importance of multiple scales

The use of filter responses at multiple scales greatly enhances the perspicuity of the iconic representations. In order to experimentally confirm the superiority of multiscale iconic representations over single scale ones, the Euclidean distances between the response vectors for corresponding points in 2D rotated and unrotated images from Fig. 3 were tested as a function of the number of scales used. Table 3 shows these results.

With less than three scales, the matching point is not the best point selected. However, with three or more scales it is ranked the best. The third column compares the distance measures used in the match of the best and second-best point in the case where the matching point is ranked first. In the case where the matching point is not the best the distance is that of the best minus that of the matching point. This column shows that even after the matching point is the best, its perspicuity continues to improve with additional scales.

To further illustrate the greater perspicuity offered by multiscale vectors, we computed the distances, this time in terms of correlations, between a model response vector for a given point (indicated by a "+") and all other points in a natural scene (Fig. 5(a)). It can be seen that a much sharper correlation peak is achieved when responses from multiple scales are used (Fig. 5(c)) than when only a single scale is used (Fig. 5(b)) to form the iconic representations. In particular, as many as 3011 points had a correlation greater than 0.90 with the model point in the case of single scale templates whereas the number of points in the case of multiple scales was only 39, most of them being located close to the model point. When the correlation threshold was raised to 0.94, 936 candidate points still remained in the single scale case whereas only one point (the model point) remained in the multiple scale case.

3.2. Iconic object representations

The filter response vectors described in the previous sections serve as iconic descriptions of image patches centered at individual scene points. For the active vision system to learn a representation of a given model object of interest with a set of such vectors, two principle issues need to be addressed: (a) some form of figure-ground segmentation of the currently foveated object, and (b) location of suitable points within the object from which model response vectors can be extracted.

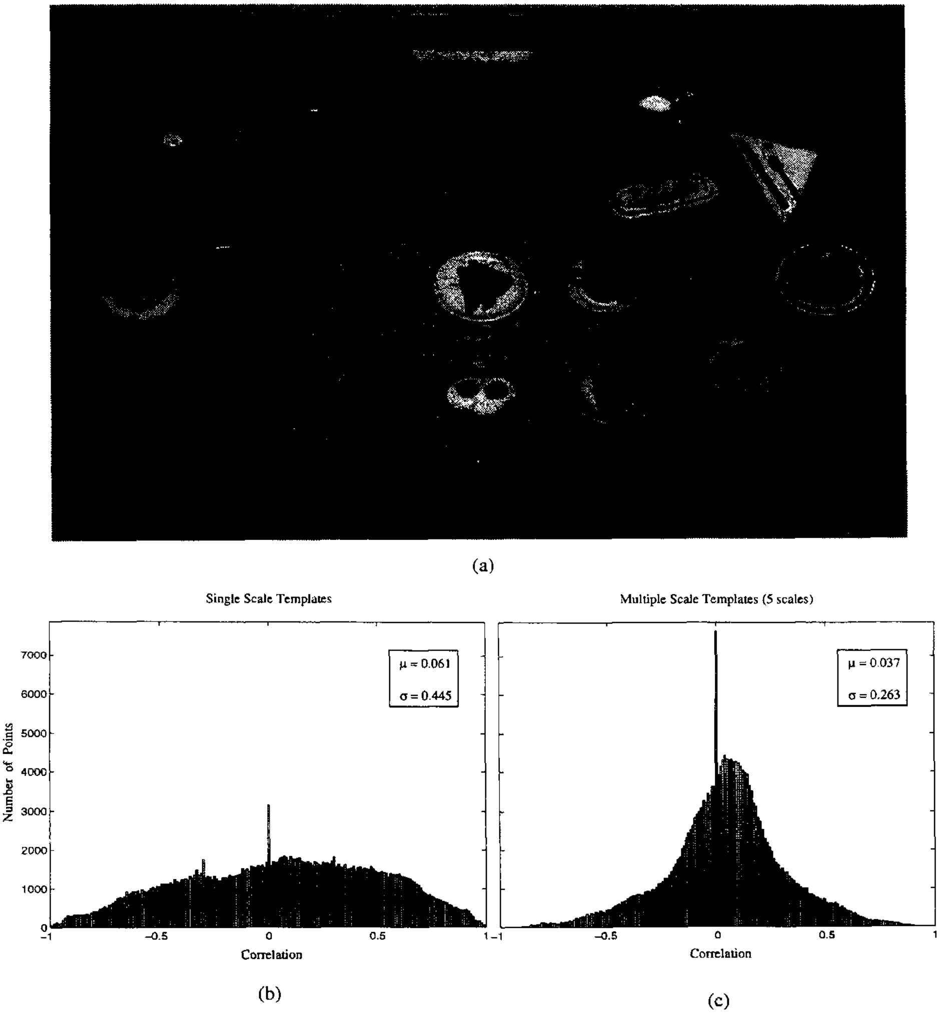


Fig. 5. A comparison between single scale and multiple scale iconic templates. (a) Shows a dining table image and a selected point (indicated by a "+") on a model object on the table. The distribution of distances (in terms of correlations) between the response vector for the selected model point and all other points in the scene is shown below for single scale response vectors (b) and multiple scale vectors (c). Using responses from multiple scales (five in this case) results in greater perspicuity and a sharper peak near the indifference distance of 0.0; only one point (the model point) had a correlation greater than 0.94 in the multiple scale case (c) whereas 936 candidate points fell in this category in the single scale case (b).

The problem of figure-ground segmentation is much simpler than the general segmentation problem especially when active vision systems are being used (as in our case). One possible strategy which has been shown to yield satisfactory results in reasonably cluttered scenes is the use of stereo in conjunction with a technique such as *zero disparity filtering* [17]. The zero disparity filter is a simple nonlinear image filter that suppresses features that have nonzero disparity; in other words, it only passes image

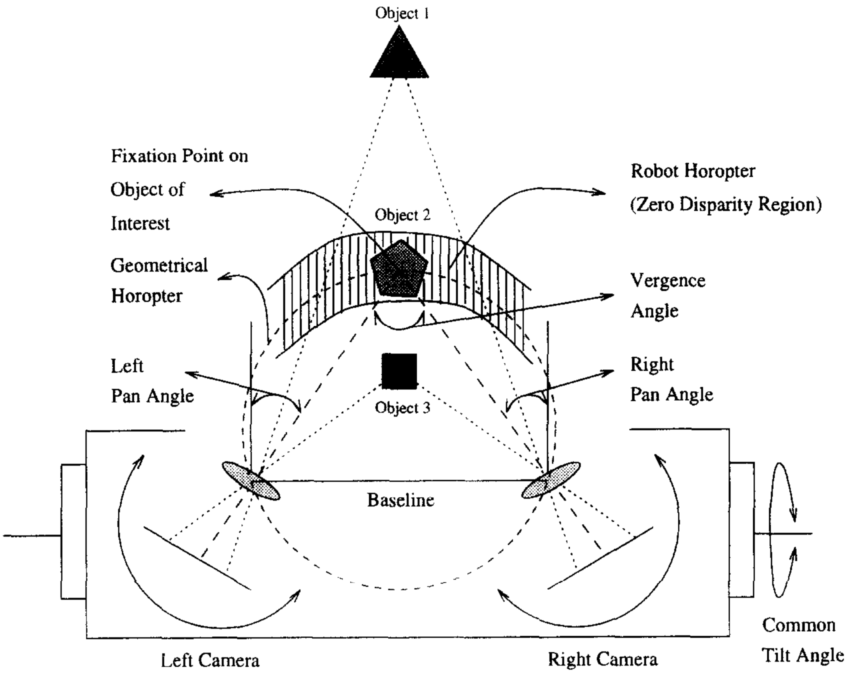


Fig. 6. Top view of the camera geometry for zero disparity filtering. The robot horopter is the region of space located approximately at the fixation distance with zero or close to zero stereo disparity. In the figure, the points on the pentagon project to approximately the same locations on the left and right image planes thereby falling in the horopter while points on the triangle and square possess respectively negative and positive disparity. The zero disparity filter passes energy only in the robot horopter, in this case, points corresponding to the pentagon, thereby achieving the required segmentation. The relative location of the horopter can be changed by manipulating the pan and tilt angles of the binocular head and adjusting the vergence angle (see [17] for more details).

energy in the horopter. Such a filter is well-suited to perform the necessary figure-ground segmentation of an object amidst a cluttered background. Fig. 6 illustrates the binocular camera geometry for the filtering technique in a hypothetical scene with three objects. Further details regarding the implementation of the zero disparity filter can be found in [17]. This approach to the figure-ground segmentation problem is similar to the one proposed by Grimson et al. [28] who use stereo to extract features that lie within a narrow disparity range about the fixation point and input these features to an alignment-based recognition system. Like Coombs, Grimson et al. use edge-operator-based matching within a predefined disparity range for aligning the cameras whereas in our case, stereo matching can be carried out by simply using the response vectors themselves [35,38] or a subset of the responses.

Assuming an approximate boundary of the object has been determined, a small number of points can be chosen within this boundary and the responses of these points can be used to represent the object. For the experiments, two different strategies were tested:

- Pick the object centroid and each of the points lying on the intersections of radial lines with concentric circles of exponentially increasing radii centered on the centroid as shown in Fig. 9(c). Only points lying within a specified distance from the approximate object boundary are used. This method ensures a dense representation of the region near the centroid while at the same time including information of other object regions as well. As described in Section 4.5.1, this strategy performed satisfactorily in our object indexing experiments with a model data base of twenty objects.
- Pick a sparse number of salient points within the object. Saliency can be readily determined using the filter responses within the object region. In our experiments, we used the relatively simple strategy of associating saliency of an object point with an estimate of its *spectral power* (*or photometric energy*) as given by the square of point's response vector magnitude:

$$\|\boldsymbol{r}\|^2 = \sum_{i,j,k} r_{i,j,k}^2,$$
$$i = 1,2,3, \quad j = 1,\ldots,i+1, \quad k = s_{\min},\ldots,s_{\max}. \tag{15}$$

  A point is deemed salient if its photometric energy is greater than an arbitrary threshold T which can be computed based on, for instance, the mean and standard deviation of the energy associated with points within the object region. This method was used for selecting object points in the location experiments in Section 4.5.2.

A particularly promising strategy (though as yet untested) is to utilize a combination of the above two strategies: points are picked from concentric circles centered at the centroids of the most salient regions of the object. Such a strategy would combine the advantages of both of the above methods with minimal computational overhead. We intend to explore the use of this technique in future implementations.

3.3. The role of multichannel visual analysis by Gaussian derivatives

While our choice of using Gaussian derivative filters for obtaining iconic descriptions may at first seem arbitrary, there exist a number of interesting properties that accrue to these filters which make them especially suited for the purposes of indexing arbitrary objects.

3.3.1. Principal components of natural images

As mentioned in Section 2.3, the different oriented derivative-of-Gaussian operators have been shown by Hancock et al. [29] (see also [63]) to be close approximations to the dominant eigenvectors of natural images. In particular, Hancock et al. used a neural network introduced by Sanger [67] for principal component analysis (PCA) to extract the first few principal components of an ensemble of natural images. Random image patches of sizes 32 × 32, 64 × 64, and 128 × 128 (windowed by a Gaussian to avoid the distortions caused by square windows) were used as input to the network from a set of 40 natural images.

Regardless of the scale of analysis, the first dominant eigenvector was found to be an approximation of the zeroth-order derivative of a Gaussian, representing the DC bias of the input signals. The second and third resembled the vertical and horizontal first-order derivative-of-Gaussians respectively. The fourth, fifth, and sixth components closely approximated other higher-order Gaussian derivatives. These six eigenvectors resembling Gaussian derivatives of orders less than 4 accounted for approximately 80% of the variance in a test set of 10,000 randomly chosen inputs.

Thus, by employing Gaussian derivatives in our representation, we implicitly retain some of the virtues of dimensionality-reduction offered by PCA. We however omit the zeroth-order derivative in order to reduce illumination dependence; we also do not use orders higher than 3 since the outputs of the higher-order filters tend to be highly correlated to the outputs of lower-order filters [42]. We additionally incorporate non-orthogonal basis functions at the second and third orders in order to preserve the ability to interpolate to arbitrary orientations and to achieve rotational invariance by using steerability as described in Section 3.1.

3.3.2. Neurophysiological correlates

Gaussian derivatives were used by Young to model primate cortical receptive field profiles [83]. An extensive analysis of these profiles revealed that the different order derivatives of the Gaussian provided the best fit among the different mathematical functions (such as Gabor functions) suggested in the literature. Orders as great as 10 were reported but the most abundant ones were found to be of order less than 4. It is interesting to note that traditional image processing algorithms use much less differential structure (usually only first or second order); the abundant use of higher-order derivatives in the primate visual systems points to the need for a much richer description of image regions than those provided by traditional edge maps in order to be able to handle complex scenes. The multiscale filter representation can be regarded as an example of such an image description.

3.3.3. Mathematical properties

A number of mathematical properties, as elucidated in [41,42], help justify some of the choices made in our representation:

- The responses obtained by applying the different order Gaussian filters at a point characterize an image patch centered at that point and form the terms of a truncated Taylor series expansion of the retinal illuminance function blurred to a degree dependent on the scale of the Gaussian. In general, an arbitrary function can be approximated by a linear combination of Gaussian derivatives in a manner analogous to Fourier series expansion in terms of sinusoidal basis functions. Such an expansion in terms of Gaussian derivative basis functions has been termed the *Gram–Charlier* series.
- The *n*th derivative of the blurred function equals the blurred *n*th derivative of the function which in turn equals the convolution of the function with the *n*th derivative of the blurring Gaussian kernel. Thus the filters may be interpreted naturally as computing "fuzzy derivatives".

- Mixed partial derivatives, which were incidently found by Young to be conspicuously absent among primate receptive field profiles, are in fact unnecessary since the other oriented filters yield a complete basis [41].
- Besides the Gaussian derivative model, a number of other functions such as Gabor functions [26] or difference of offset Gaussians have been proposed as models of cortical receptive field profiles. The need to search for the "ideal" filter model however is somewhat obviated by the fact that the asymptotic form for a very high-order derivative of a Gaussian is just a sine (odd order) or cosine (even order) modulated by a Gaussian envelope, in other words, a Gabor function. In addition, the difference of Gaussians model can be looked upon as simply a hardware implementation of the Gaussian derivative operator [83].
- The spatial description in terms of partial blurred derivatives is equivalent to a spectral description in terms of a local Fourier decomposition of a windowed portion of the retinal image via a set of bandpass spatial filters. This observation agrees with the classical neurophysiological understanding of the primate visual system in terms of "spatial frequency channels" [12,21].

The Gaussian derivatives also have the property that they approximately function as "matched filters" to some of the most significant elementary visual stimuli. For example, the first- and second-order Gaussian derivatives are known to be close to optimal for detecting edges and bars respectively [13]. This may not be surprising in light of the fact that these derivatives approximate the principal components of natural image patches comprised of different elementary visual features.

Image features such as bars and edges are usually present at a multitude of scales; the scant information usually available regarding the scales at which features of interest may be located render the problem of scale selection an especially tricky one. The incorporation of responses from multiple scales avoids the need to make a priori and often unwarranted choices regarding the scale of visual analysis.

The presence of elementary features at a variety of orientations point toward the need for filters that can detect features across a continuum of orientations. The isotropic Laplacian of Gaussian may seem to be the most obvious choice in this case as suggested by Marr [47] but signal-to-noise ratio arguments [13] support the use of oriented filters. The fact that Gaussian derivatives are not only easy to compute but also easily steered to different orientations using relatively simple interpolation functions (see Section 2.4.1) make them an especially desirable choice in our representation.

The representation in the form of a long vector of responses has considerable noise immunity. No single filter can be expected to produce the same response under varying viewing conditions but the representation combines information from a large collection of nearly independent image measurements, thereby making it much more robust than traditional representations that rely on fewer image measurements.

4. Visual routines

The normalized multiscale filter response vectors (or "zip-codes" [7]) serve as signatures of the photometric distributions surrounding various points within an object.

The iconic object descriptions formed by a set of such multiscale filter response vectors can be readily embedded in visual routines to solve the object identification and object location problems. In the identification problem, vectors from a single localized point in space can be matched against stored model vectors of different objects. In the location problem, a single model vector (from the set describing a particular object) can be matched against the entire image described as a collection of such vectors, one for every point.

In order to compare the response vector $\boldsymbol{r}^{\mathrm{i}}$ from an image point and the response vector $\boldsymbol{r}^{\mathrm{m}}$ from a model object, a similarity metric is required. One commonly used metric is the sum of squared distances (SSD) metric:

$$d'_{im} = \|\boldsymbol{r}^{\mathrm{i}} - \boldsymbol{r}^{\mathrm{m}}\|^2. \tag{16}$$

For most of the experiments however, we chose to use the related metric of normalized dot-product (or correlation) of two vectors:

$$d_{im} = \frac{\boldsymbol{r}^{\mathrm{i}} \cdot \boldsymbol{r}^{\mathrm{m}}}{\|\boldsymbol{r}^{\mathrm{i}}\| \, \|\boldsymbol{r}^{\mathrm{m}}\|} \tag{17}$$

primarily because the dot-product operation can be efficiently implemented using convolutions on video-rate image processors such as the Datacube MV200. In addition, the normalization by vector length helps to make the matching process resilient to global contrast changes caused by varying lighting conditions.

4.1. Object identification

In the general case, more than one model object can share the same iconic index. Let $M(\boldsymbol{r}^{\mathrm{m}})$ denote the set of models (represented by their labels) that have $\boldsymbol{r}^{\mathrm{m}}$ as part of their set of response vectors.

The identification algorithm proceeds as follows:

(1) First obtain, for each chosen response vector $\boldsymbol{r}^{\mathrm{i}}$ on the image of the object to be identified, the model response vectors $\boldsymbol{r}^{\mathrm{m}}$ such that $d_{im} \geqslant T$ (assuming correlation is used as the distance metric), where T is a prechosen threshold.

(2) For each model M_i, initialize the "evidence" array $E(M_i)$ to 0.

(3) For each $\boldsymbol{r}^{\mathrm{m}}$ from step (1) and each model $M_i \in M(\boldsymbol{r}^{\mathrm{m}})$, set

$$E(M_i) := E(M_i) + 1.$$

(4) Output the model label M such that $E(M) = \max\{E(M_i)\}$.

In other words, the outcome of the identification algorithm is determined by a straightforward voting process, with the model obtaining the largest number of votes being deemed the winner. The threshold T can be determined experimentally as described in [65] where the above routine is realized in a slightly modified form in the context of Kanerva's sparse distributed memory model [36].

4.2. Object location

The location routine crucially depends on the fact that only a single model object is being matched to an image at any instant. Let us denote this model that is to be located in the current image as

$$M = \{\boldsymbol{r}^{\mathrm{m}}, m = 1, \ldots, m_{\max}\}. \tag{18}$$

The location algorithm in its most general form proceeds as follows:

(1) Assuming that the distance metric being used is correlation, for each response vector $\boldsymbol{r}^{\mathrm{m}}$ representing some model point m, create a distance image I_m defined by

$$I_m(x, y) = \min[\beta d_{im}, I_{\max}], \tag{19}$$

where d_{im} is computed between the model vector $\boldsymbol{r}^{\mathrm{m}}$ and the image vector $\boldsymbol{r}^{\mathrm{i}}$ for the point (x, y), $I_{\max}$ is the maximum possible image intensity value and β is a suitably chosen scaling constant (this makes the best matching point the brightest spot in the image).

(2) Find the best match point (x_{b_m}, y_{b_m}) in the image for each m using the relation

$$(x_{b_m}, y_{b_m}) = \arg\max\{I_m(x, y)\}. \tag{20}$$

(3) Construct a binary "salience" image $S(x, y)$ where

$$S(x, y) = \begin{cases} 1, & \text{if } (x, y) \in \{(x_{b_m}, y_{b_m})\}, \quad m = 1, \ldots, m_{\max}, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

(4) Output the location of the object in the current image as (x_b, y_b) where

$$(x_b, y_b) = \arg\max\{S(x, y) * B(x, y)\} \tag{22}$$

and B is an appropriate blurring or local averaging function whose size is usually known in active vision environments.

For the sake of convenience and clarity in understanding the performance of the algorithms, we present our results in terms of the distance image rather than the results after applying the blurring operation.

4.3. A simple visual task using visual routines

Fig. 7 illustrates the use of the visual routines in a naive visual/motor task involving replication of patterns of square blocks located on one part of a large board onto another part of the board. Suppose gaze is first fixed on an initial target pattern (marked by a "+" in Fig. 7(a)) and its feature vector memorized (this involves an implicit "What" operation). When gaze has been shifted to a different point as shown in (b), there occurs the problem of moving gaze back to the previously foveated point. But this is simply the "Where" problem, which can be tackled using the location algorithm discussed above to obtain a best match point (shown in (c) as the brightest point in the distance image).

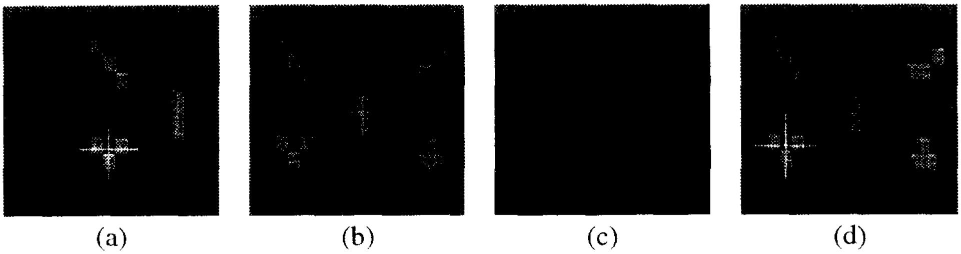


Fig. 7. Solving a simple visual task using the What/Where routines. (a) Initial gaze location whose iconic representation is stored in short-term memory ("What"). (b) A new (arbitrary) gaze point. (c) To get back to the original location ("Where"), the "distance image" is computed: the brightest spot represents the location whose iconic feature vector is closest to that of the original gaze point. (d) Location of best match is marked and an oculo-motor command (obtained, for instance, using a learned motor map [62]) can be executed to foveate that point.

Gaze can then be transferred to the retinal position marked by "+" (as shown in (d)) by issuing an appropriate oculo-motor command using, for instance, a learned motor map [62].

4.4. Implementation of the visual routines

Both the identification and location algorithms described in the previous sections have been implemented using an active vision system comprised of the University of Rochester binocular head with two movable color CCD cameras (Fig. 8) that provide input to a Datacube MaxVideo^{TM} MV200 pipeline image processing system. A single servo-motor controls the tilt of the two-camera platform while two separate motors control each camera's pan angle, thereby providing independent vergence control. The use of a binocular head allows for strategies such as zero disparity filtering for figure-ground segmentation and occluder detection as described in Section 3.2 and Section 4.6 respectively.

The MV200 is comprised of a single integrated 6U VME circuit board capable of a wide range of frame-rate image analysis capabilities. Of particular interest to our work here is its ability to perform convolutions at frame-rate (30 per second).

Both the location and identification algorithms require a large number of distance computations: for location, a given model vector must be compared to the response vectors for all the points in the current image while in the case of identification, a response vector from the image must be compared to all the model vectors stored in memory. Our implementation greatly optimizes this step by *using convolutions for distance computations* since the similarity metric used for comparing two vectors is correlation i.e. normalized dot product of the two vectors.

We briefly describe here the implementation of the location algorithm. A similar strategy can be applied for the identification algorithm as well, as described in [65]. Given a live input image (of size 512 × 480) from the camera, the MV200 executes nine convolutions using nine different 8 × 8 discrete Gaussian derivative kernels on

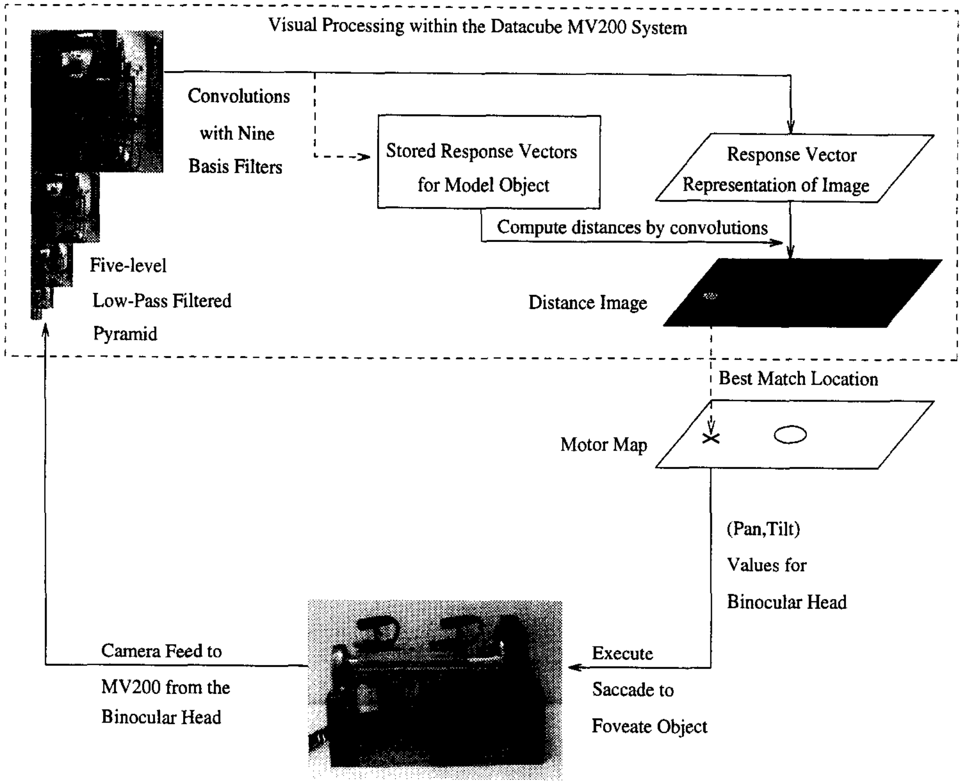

Visual Processing within the Datacube MV200 System
Convolutions with Nine Basis Filters
Stored Response Vectors for Model Object
Response Vector Representation of Image
Compute distances by convolutions
Five-level Low-Pass Filtered Pyramid
Distance Image
Best Match Location
Motor Map
(Pan,Tilt) Values for Binocular Head
Camera Feed to MV200 from the Binocular Head
Execute Saccade to Foveate Object

Fig. 8. Implementation diagram for the location algorithm.

a low-pass filtered five-level pyramid of the image to obtain the response vectors for all points in the current image; these vectors are stored in a "memory surface" S. During the training phase, filter responses are extracted for each of the sparse number of points located within the segmented object according to the criteria described in Section 3.2.

During the location phase, a model response vector is loaded into the 8×8 convolution kernel and convolved with the memory surface S containing the response vectors for each point of the input image; the closest vectors can be selected by simply thresholding the results of the convolution at individual thresholds to obtain candidate match points. For the experiments, the point whose response vector achieved the highest correlation with the model vector was chosen as the location of the model point in the input image. The final step involves a foveation or centering of the selected point; the motor commands for achieving this foveation were learned autonomously by the system by using a motor map and a variant of Kohonen's learning rule. The reader is referred to [62] for more details. A summary of the current implementation of the location algorithm is given in Fig. 8. The diagram for the identification algorithm is similar and can be found in [65].

4.5. Experimental results

This section contains experimental results for both the object identification and location routines described above. The former was tested using a well-known object data base containing 20 complex 3D objects. The latter was tested under a wide range of viewing conditions ranging from image variations caused by object/camera motion to distortions due to clutter and minor occlusions; the performance of the algorithm was also tested on a cluttered 3D scene as function of camera displacement.

4.5.1. Object identification results

We tested the identification algorithm on the Columbia object data base that was originally used in [51] by Murase and Nayar and which accompanies the SLAM software package from Columbia. The data base contains 3D objects exhibiting a wide variety of properties ranging from uniform reflectance and simple shapes to complex textural properties that are hard to model geometrically. The identification algorithm itself was realized within the framework of an associative model of visual memory based on Kanerva's sparse distributed memory model [36]. This form of memory provides a convenient platform for learning the association between an object's appearance (in terms of response vectors) and its identity (given by a binary vector label) and offers a number of specific advantages such as interpolation between stored views, constant indexing time (due to a constant number of storage locations), possibly greater storage capacity over sequential memory, and anthropomorphic learning behavior. Further details can be found in [64, 65] (see also [63] where a topographic form of the memory model is used).

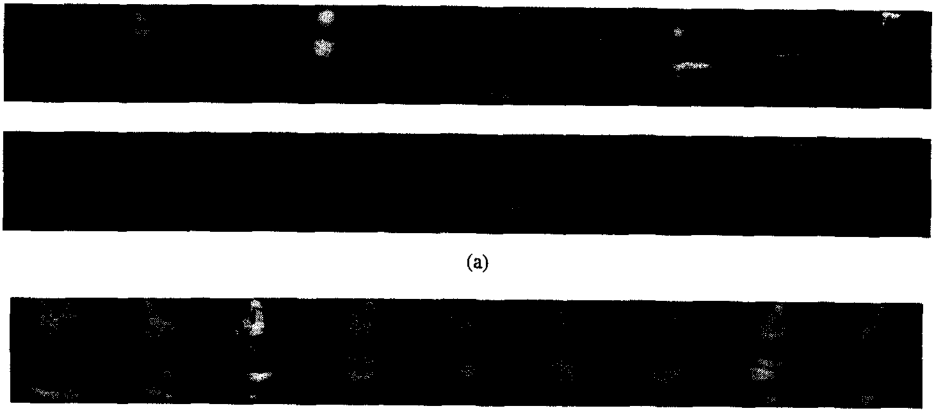
Fig. 9(a) shows the 20 3D objects in the data base for a given pose. The data base contains 72 presegmented images of each object (imaged at 5° rotational increments in pose), each image 8-bit quantized and normalized for brightness at a size of 128 × 128 pixels. During the training phase, 36 canonical views of each object at 10° increments in pose were used to extract response vectors for storage in memory; twelve of these are shown in Fig. 9(b). For testing the indexing scheme, we randomly selected images of objects corresponding to poses that lie exactly in between the training poses; the testing set size was thus 720 images, the same as the training set size (Fig. 9(d)). The recognition results are summarized in Fig. 9(e). Even when only one point at the object centroid was used per object, 70% of the test cases were successfully recognized. Addition of more points per object increased the recognition rate until 100% accuracy was achieved when 25 points were used to describe each object.

4.5.2. Object location results

Location performance was tested using live camera input from the binocular head. All images were obtained within the Rochester Robotics and Vision Laboratory, each image containing a wide variety of 3D objects. For the experiments, salient model points on objects were picked according to the saliency criterion discussed in Section 3.2.

Effect of object and camera motion

We first tested location performance in the presence of object and camera movements. In the first experiment, the wrist of the Unimate Puma Robot Arm was chosen as



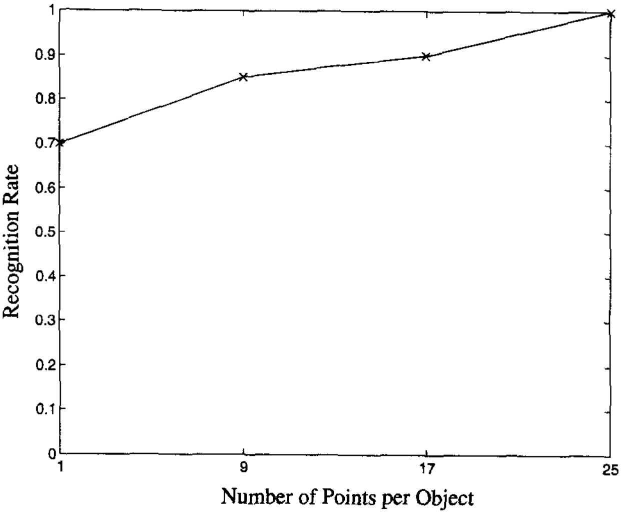
(a)

(b)

(c)

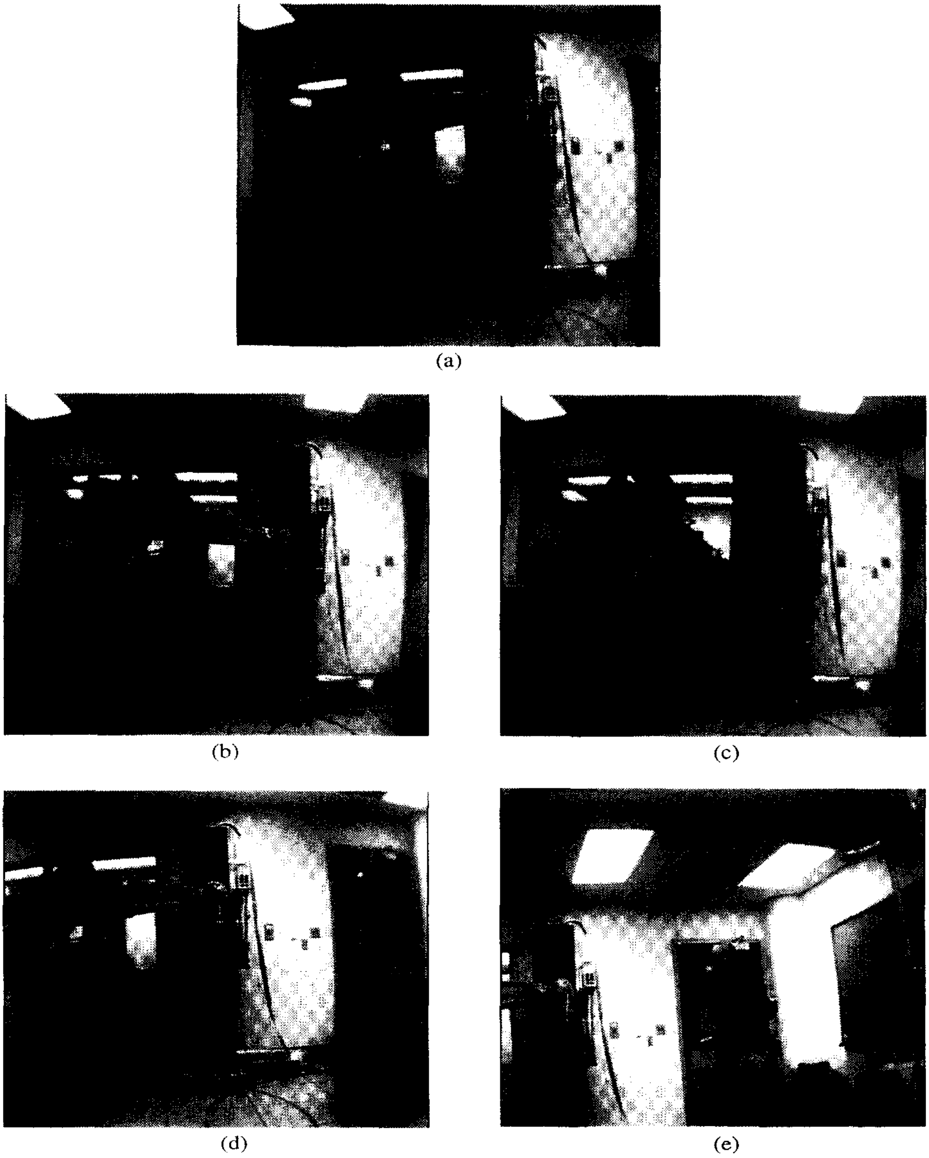
Number of Model Objects	20
Number of Poses per Object	36
Total Number of Training Images	720
Testing Set Size	720

(d)



(e)

Fig. 9. Identification results. (a) The 20 objects used in the experiment. (b) For each object, 36 images were used at 10° rotational increments in pose to represent the entire pose space. Nine of these images for a particular model object are shown in the figure. (c) For a given object in a particular pose, response vectors were selected for the points of intersections of radial lines with concentric circles of exponentially increasing radii centered on the object centroid. (d) summarizes the experimental parameters. Random images were selected for testing the indexing method from the 720 images in the testing set. (e) Recognition rate (fraction of test images correctly recognized) plotted as a function of number of points used per object for identification. All test cases were successfully recognized when 25 points were used as indicated in (c) to characterize an object.



(a)

(b)

(c)

(d)

(e)

Fig. 10. Location in the presence of object and camera motion. (a) The wrist of the robot arm was chosen as the model object and the response vector from a salient point (marked by "+") on its tip was stored for future comparisons. (b) and (c) show the results of the location algorithm on images obtained after two discrete vertical movements of the arm. (d) and (e) show the results after two movements of the binocular head.

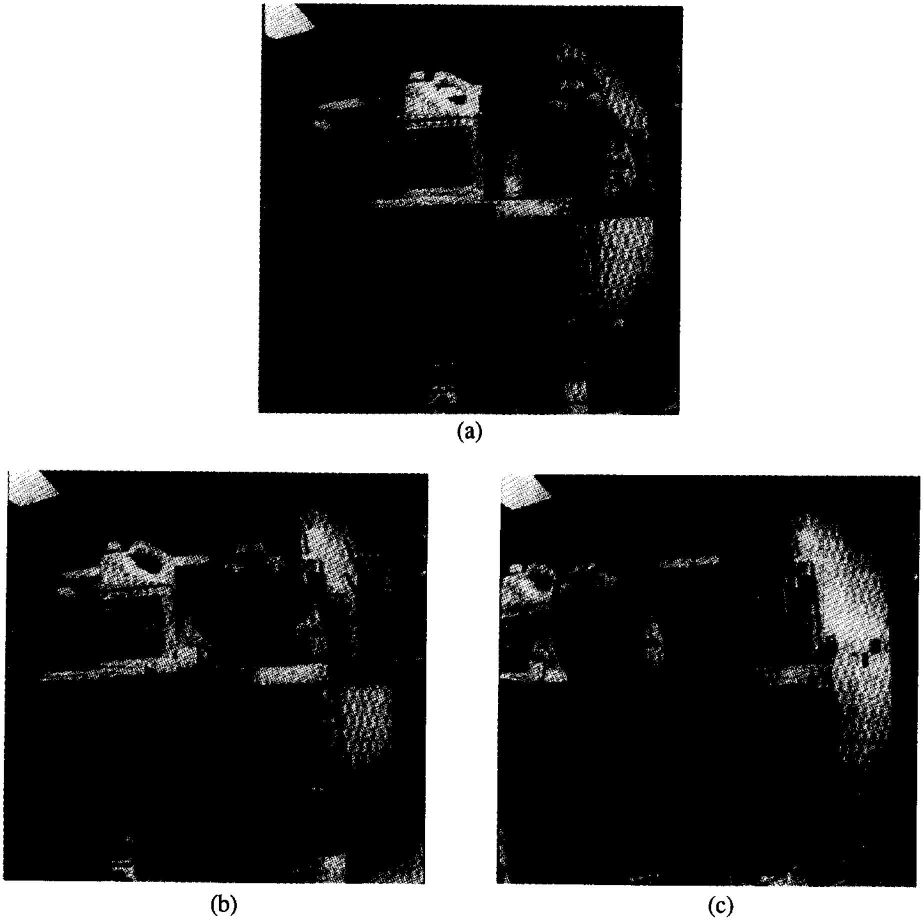


Fig. 11. Object location in the presence of perspective distortions and background variations. (a) The response vector for a salient point near the centroid of the model object was used for the experiment. (b) and (c) show the results of running the location algorithm on two images obtained after varying amounts of motion of the model object.

the model object and the model response vector for a salient point (marked by a "+" in Fig. 10(a)) on the tip of the wrist was used for location. Figs. 10(b) and 10(c) show the images obtained after two movements of the vertical joint of the arm and the corresponding best matching points found by the algorithm. The experiment was repeated for a 10° horizontal camera movement (Fig. 10(d)) followed by a 20° horizontal movement and a 10° vertical movement. In both cases, the algorithm was able to locate the model with a sufficiently high degree of accuracy as indicated by the figures.

In a second experiment, we checked for the effects of perspective distortions and changing background caused by object motion. The model object was represented by the response vector from a salient point near its centroid as shown in Fig. 11(a). The

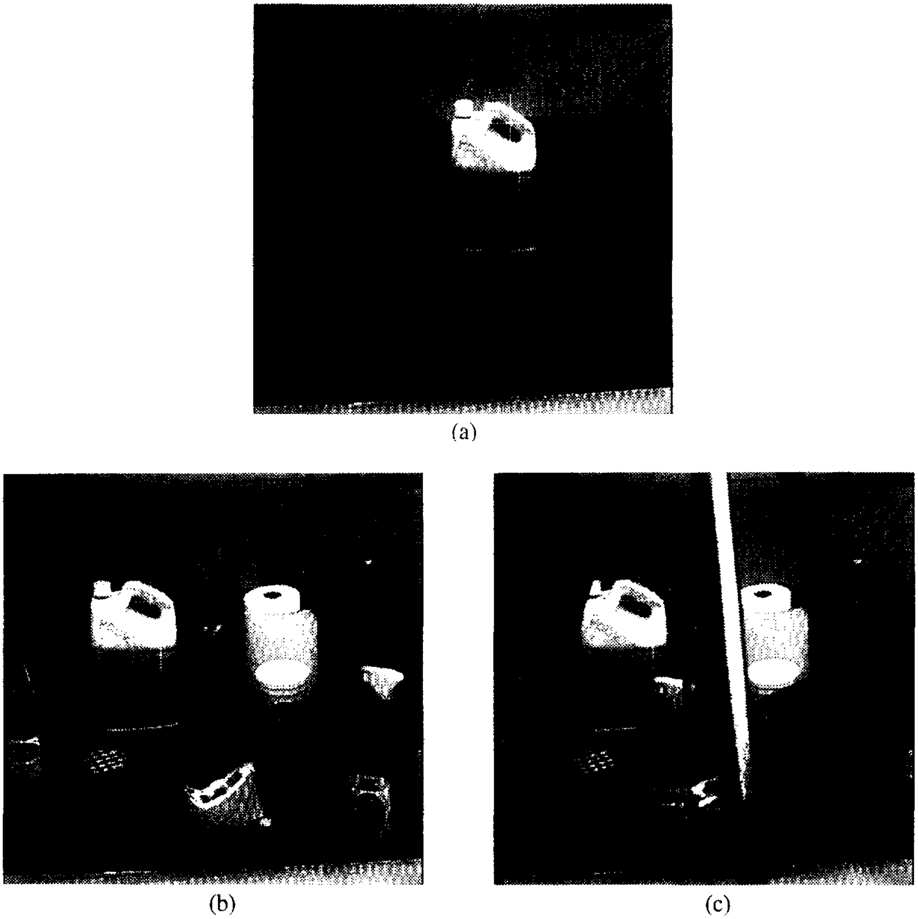


Fig. 12. Object location in the presence of scene clutter and minor occlusions. (a) shows the salient points whose response vectors were used to characterize the model object. (b) and (c) show the output of the location algorithm on images obtained after the model object was moved and varying amounts of scene clutter/occlusions were introduced.

two figures below show the images obtained after horizontal motion of the same object; the closest matching points found by the location algorithm for the two images are shown marked by a "+". In both cases, the object was successfully located despite the variations caused by changes in perspective and different backgrounds.

Effect of clutter and minor occlusions

In order to verify the location algorithm's resilience to clutter and minor occlusions, a white can of paint thinner with textured markings near its centroid was used as the model object. Three salient points in different locations within the object (shown in Fig. 12(a)) were used to form the object's iconic representation. The object was then moved to a different location and significant clutter was introduced. As can be seen from

Fig. 12(b), the location algorithm was able to find close matches for all three model points. We also introduced noise in the form of occlusions near the model object; the corresponding points found by the location routine are shown in Fig. 12(c).

Location performance

The location algorithm's performance was measured for a given set of objects in a scene as a function of the image distortions caused by varying degrees of camera movements. Fifteen objects were selected for the experiment and each was represented by a single salient point as shown in Fig. 13(a). The objects were imaged by one of the cameras of the binocular head placed at a distance of approximately five feet from the objects. For each model object, the binocular head was made to execute horizontal saccades of varying amplitudes to obtain new images on which the location algorithm was run to locate the model object. Fig. 13(b) shows the location rate plotted as a function of saccade amplitude. All objects were successfully located for amplitudes less than 16°; the location algorithm located all but one of the fifteen objects for amplitudes of 16° and 20°.

4.6. Handling partial occlusions

The iconic object representations are robust to minor occlusions since these can essentially be treated as noise as illustrated in some of the experiments in the previous section; however, for larger occlusions, the recognition algorithms will fail if nothing is done since the occluder will usually distort the filter responses in larger amounts than can be tolerated simply as noise.

Interestingly, humans have a similar problem. Fig. 14 shows the experimental setup designed by Nakayama and Shimojo [54] to test subjects' ability to identify faces in the presence of negative and positive occlusion cues. In one instance, the face is painted on a picket fence (shown as dark horizontal bands of noise in the figure); in the other, it is behind the picket fence. The results show that identification is markedly improved in the latter case. One interpretation of this result is that the early visual system performs figure-ground segmentation in such a way that the occluders, whenever positive (as in Fig. 14(b)), are automatically segmented away from the object of interest. This observation forms the inspiration for our solution [8].

Suppose that an active imaging system is used. As a consequence we can assume that the occluder can be detected by a method such as disparity filtering [17]. As described in Section 3.2, disparity filtering is a way of creating a filter that passes image energy only in the horopter. Ideally one can create a template $T(x,y)$ such that $T(x,y) = 1$ for material in the horopter and $T(x,y) = 0$ otherwise. We assume the existence of such a template for our subsequent calculations.

4.6.1. Occlusion algorithm

The filter responses are the responses for a set of basis functions. As a consequence the image intensities near every point can be reconstructed by appropriately combining the responses and filter functions. As the functions are not orthogonal, a pseudo-inverse must be used to do this [35].

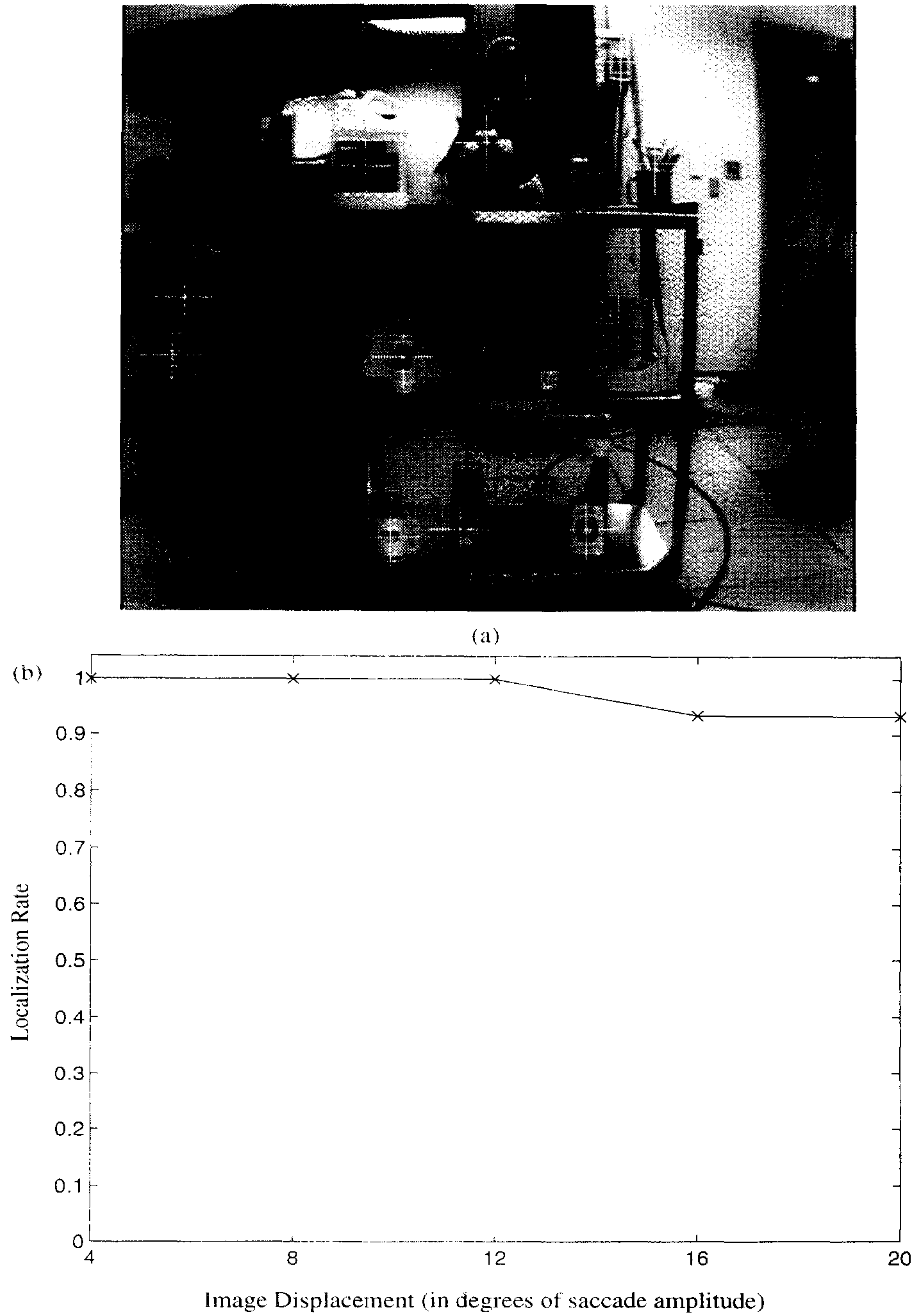


Fig. 13. Location performance. (a) indicates the salient points chosen to represent the fifteen model objects used for the experiment. (b) shows the success rate of the location algorithm as a function of horizontal camera movement in degrees. Apart from one object (for camera movements of 16° and 20°), each of the fifteen objects was correctly located in each of the new images obtained after the different amounts of camera movements.

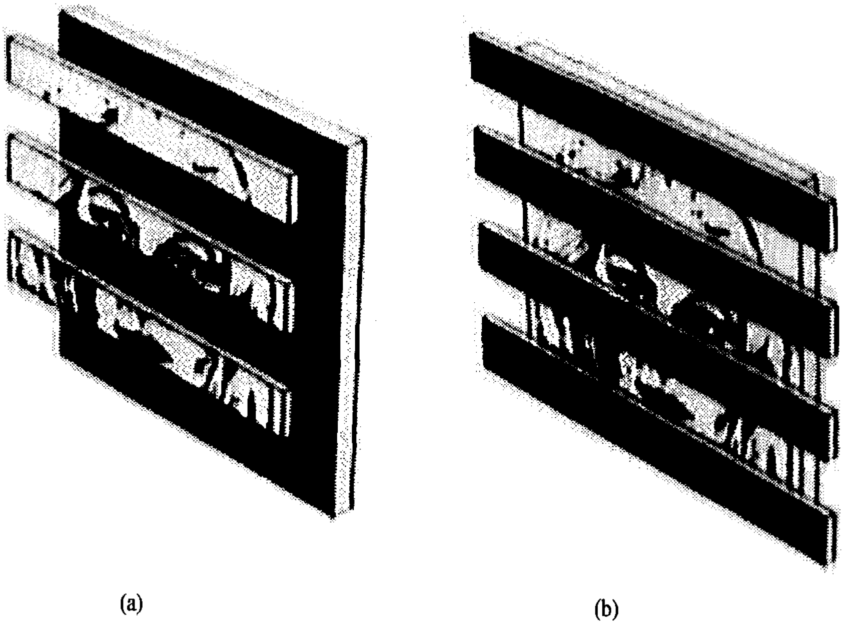


Fig. 14. The role of stereo disparity in human recognition performance. The figure (from [56]) shows a simple rendition of typical stimuli used by Nakayama and Shimojo in their experiments to judge recognition performance in the presence of negative (a) and positive (b) occlusions in the form of dark horizontal bands of noise. In all cases, subjects recognized faces with higher accuracy in situation (b) than in situation (a).

Any spatial filter with a finite impulse response can be represented as an $p \times 1$ vector F_i, where p = number of pixels in the support of the filter and $i = 1, \ldots, n$, n being the number of basis filters used in the iconic representation. A set of such filters can be stacked side by side to form a $p \times n$ matrix F. For an image patch represented as an $p \times 1$ vector I, the $n \times 1$ response vector is

$$r = F^{\mathrm{T}} I. \tag{23}$$

Applying *singular value decomposition* [69] to F^{T} results in

$$F^{\mathrm{T}} = U \Sigma V^{\mathrm{T}}, \tag{24}$$

where U is an $n \times n$ orthogonal matrix , Σ is an $n \times p$ diagonal matrix, and V^{T} is a $p \times p$ orthogonal matrix. We can now *reconstruct* an image patch given a response vector r by using the relation

$$I' = V \Sigma^{-1} U^{\mathrm{T}} r, \tag{25}$$

where Σ^{-1} denotes a $p \times n$ diagonal matrix whose diagonal entries are the multiplicative inverses of the corresponding diagonal entries of Σ.

Note that the matrix $V \Sigma^{-1} U^{\mathrm{T}}$ is independent of the response vector and hence can be precomputed and stored. Reconstruction then merely involves the multiplication of a $p \times n$ precomputed matrix by an $n \times 1$ vector. This ability to reconstruct the local

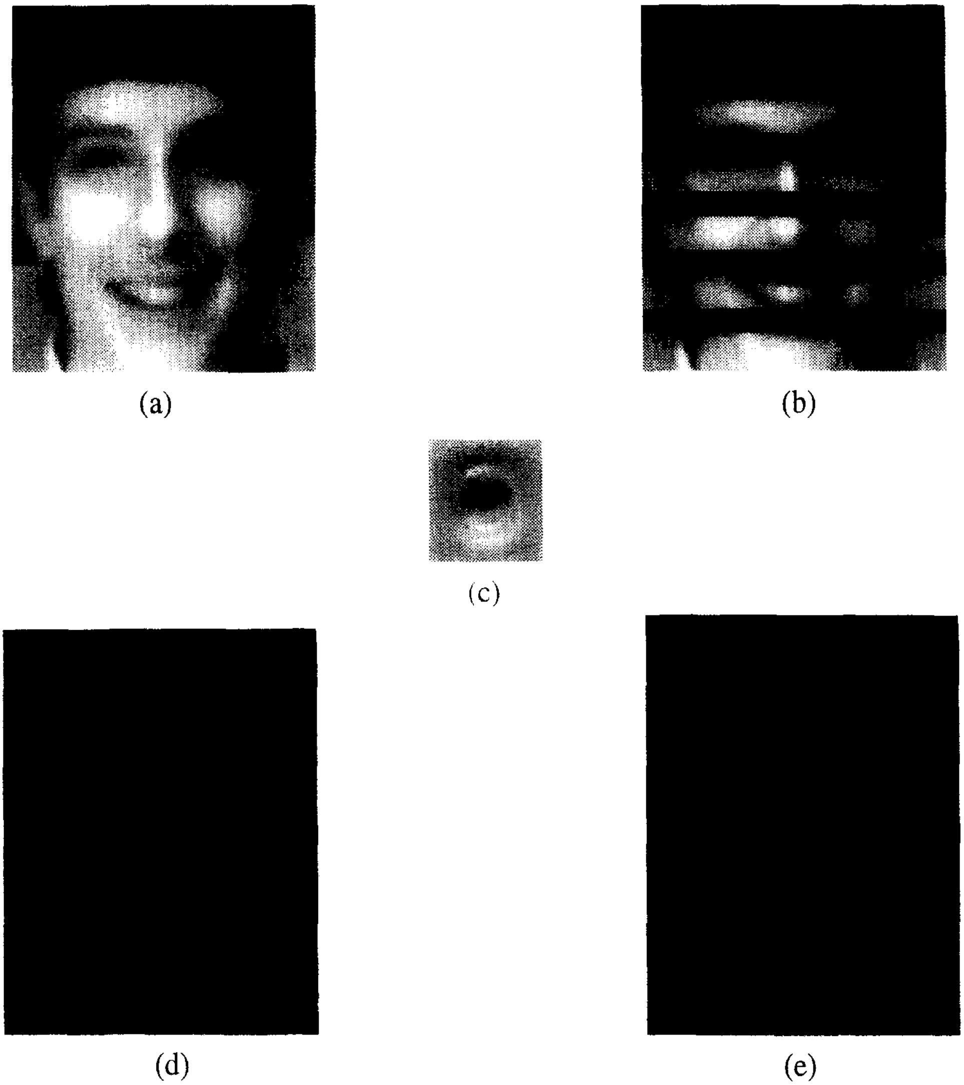


Fig. 15. A test of the occlusion algorithm. (a) The original image; (b) the occluded image; (c) the reconstructed patch of the left eye (unmasked); (d) the distance image showing the left eye correctly located (brightest point in the image) by using responses from the masked eye patch; (e) the result of directly comparing the unoccluded responses from (c) with those from the occluded image.

intensities allows the stored prototype to be made comparable to the occluded image responses. For every point, the reconstructed image intensities are appropriately masked using the occluding template. A similar process is done to the incoming image. Thus the masked reconstructed image and the masked input image are now in the same coordinate system and can be compared by differencing their filter responses. This is formalized in the following algorithm for occlusion near a point (x_0, y_0):

(1) Use model response vector to reconstruct the local image patch, $I'(x_0, y_0)$.

(2) For every point (x, y) in the image do:
Compute $I''(x, y) = T(x, y)I(x, y)$ for all (x, y) in a local domain of appropriate size.
Compute new filter responses f'' from I''.
Compare those with the filter responses f computed from $I'(x_0, y_0)T(x_0, y_0)$ to compute $d(x', y')$.

(3) The sought after point is given by $\arg\min d(x', y')$.

To demonstrate the occlusion algorithm, we created a face image similar in spirit to that of Fig. 14. Fig. 15 shows the results of using the occlusion algorithm on the face image. Just to make the obvious point, if the raw filter responses in the occluded image are compared to the previous point, then, as they are not comparable, the best match is not correct. This computation is shown in Fig. 15(e). In order to test the sensitivity of the algorithm to the size and relative location of the occluder with respect to a point of interest, we ran the algorithm on a simple table top scene (Fig. 16(a)) in the presence of increasing occlusion in the form of vertical dark bands of noise in the region of a test point near the end of the spatula's handle. A unique best matching point was correctly found when 20% or even 40% of the image patch centered at the selected point was occluded as given by the brightest points in the respective distance images of Figs. 16(d) and 16(f). The original point remained among the possible candidates for matches, though it was no longer the only contender, when the occluder occupied more than 60% of the area centered at the test point (Figs. 16(g) and 16(h)).

4.7. *Using iconic representations with space-variant sensors*

There has been recent interest in the use of space-variant sensors in active vision systems for tasks such as visual search and object tracking [75]. Such sensors realize the simultaneous need for wide field-of-view and good visual acuity; in addition, they decrease the amount of information that needs to be processed, thereby reducing the computational load on the active vision system. One popular class of space-variant sensors are *log-polar sensors* which have a small area near the optical axis of greatly increased resolution (the fovea) coupled with a peripheral region that witnesses a gradual logarithmic falloff in resolution as one moves radially outward. These sensors are inspired by similar structures found in the human and primate retina where one finds both a peripheral region of gradually decreasing acuity and a circularly symmetric *area centralis* characterized by its greater density of receptors and a disproportionate representation in the optic nerve [14]. The peripheral region, though of low visual acuity, is more sensitive to light intensity and movement.

The existence of a region optimized for discrimination and recognition surrounded by a region geared towards detection thus allows the image of an object of interest detected in the outer region to be placed on the more analytic center for closer scrutiny. Such a strategy however necessitates the existence of methods to determine which location in the periphery to foveate next. In the case of humans, the "where-to-look-next" issue is addressed by both bottom-up strategies such as motion or salience clues from the periphery as well as top-down strategies such as search for a particular form or color.

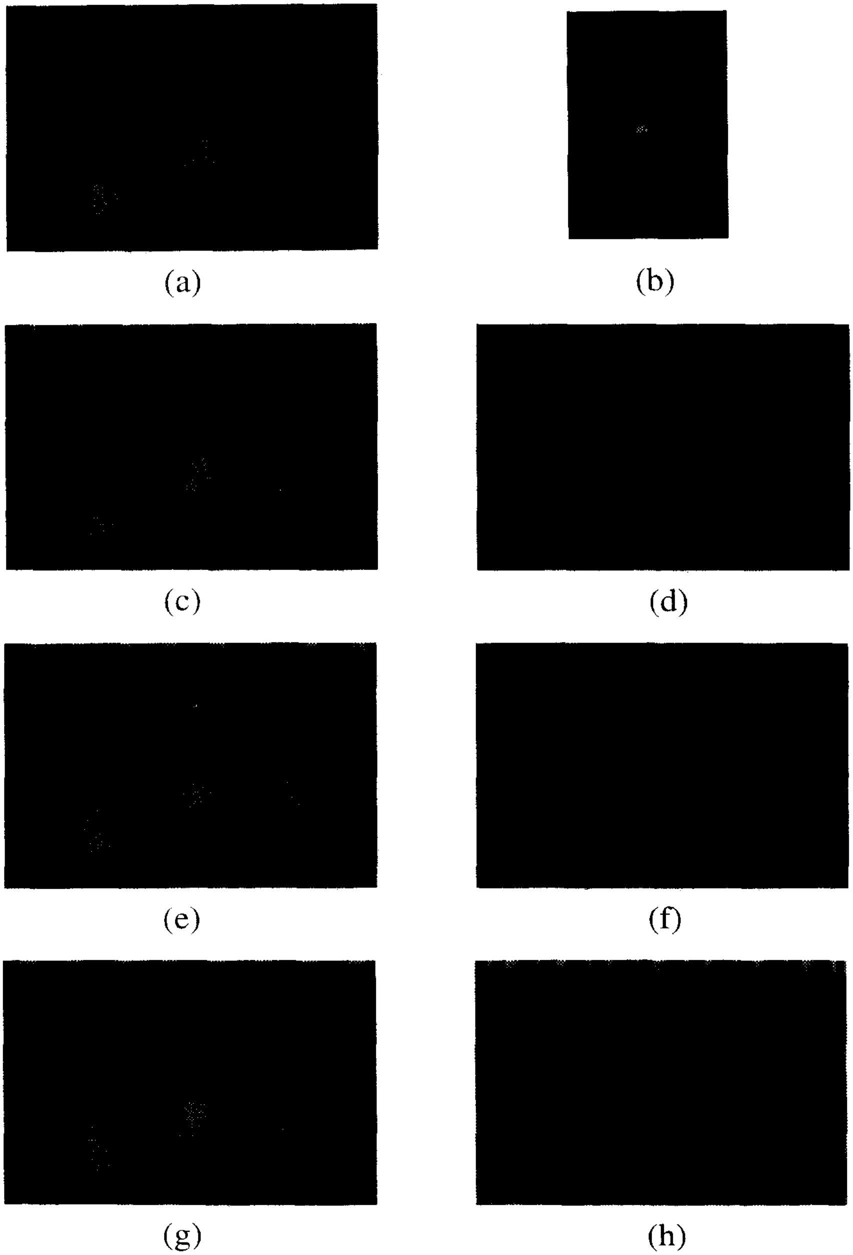


Fig. 16. Sensitivity to degree of occlusion. (a) Unoccluded table top scene and (b) reconstructed patch of a test point at the tip of the spatula's handle; (c) partially occluded scene with 20% of the image patch near selected point occluded and (d) distance image showing the test point correctly located (brightest point in the image); (e) and (f) correct point is still the unique best match (brightest point) when occlusion is increased to 40% of image patch; (g) and (h) best match is no longer unique when the degree of occlusion is increased to 60%.

While bottom-up alerting cues such as motion have recently been incorporated in some active vision systems to initiate "capture saccades" [52], the possibility of using top-down cues for foveal redirection has remained relatively unexplored (an exception is the use of color in [72]), even though it is well known that humans have the ability to use information from the low-resolution periphery in guiding visual search [16].

In Section 4.2, we illustrated the use of the location algorithm in achieving top-down foveal targeting using a uniform resolution sensor. Using the same location algorithm with sensors exhibiting nonuniform resolution characteristics will obviously result in failure. However, the multiscale structure of the response vectors can be effectively exploited to obtain a modified location algorithm [61].

Consider first the case when the scale of the object is fixed. Then if the sensor turns away from the object, the decrease in radial resolution causes an effective reduction in scale (in addition to some other minor distortions) of previously foveated regions as they move towards the periphery. The filter responses usually vary smoothly between scales; it is thus possible to establish a correspondence between the two response vectors of the same point on an object imaged at different scales by using a simple *interpolate-and-compare* scale matching strategy. That is, in addition to comparing an image vector and a model vector directly as in the location algorithm, interpolated versions of the image response vector are also compared with the model response vector (the interpolation step can be carried out using, for instance, *radial basis functions* [60]).

In the case where the scale of the object itself changes, the increase or decrease in the scale of the object is matched to a certain extent by the corresponding decrease or increase in resolution of the sensor. The degree of this match effectively modulates the extent to which scale matching will have to be done.

The use of the interpolate-and-compare scale matching strategy in conjunction with space variant sensors is illustrated in Fig. 17. The sequence of "cortical images" (a) through (f) in Cartesian coordinates were obtained by passing appropriate portions of a uniform resolution original image through a log-polar mapping [81], with an effective radial drop-off rate (geometric) of 1.075 pixels of the original image per pixel of the new image. The resulting sequence simulates the effect of moving a log-polar space variant sensor from a point (marked by a "+") in the fovea (indicated by a circle) towards the right. (g) illustrates the scale matching strategy of interpolating (in this example, simply shifting to the next scale) and comparing the response vectors; the obvious correlation between the response vectors as a function of changes in scale can be clearly seen. Once the appropriate scale correction is factored in, the location algorithm retains its utility as an effective routine for foveating peripheral targets.

4.8. Routines for scale invariance and looming detection

In order to be successful in a dynamic environment, a recognition mechanism must have the ability to handle scale changes in the projection of an object in an image caused, for instance, by motion of the camera towards or away from the object. Our experiments seem to indicate that the location and identification algorithms can handle up to 5-10% variations in scale but larger scale changes distort the filter responses causing the algorithms to fail.

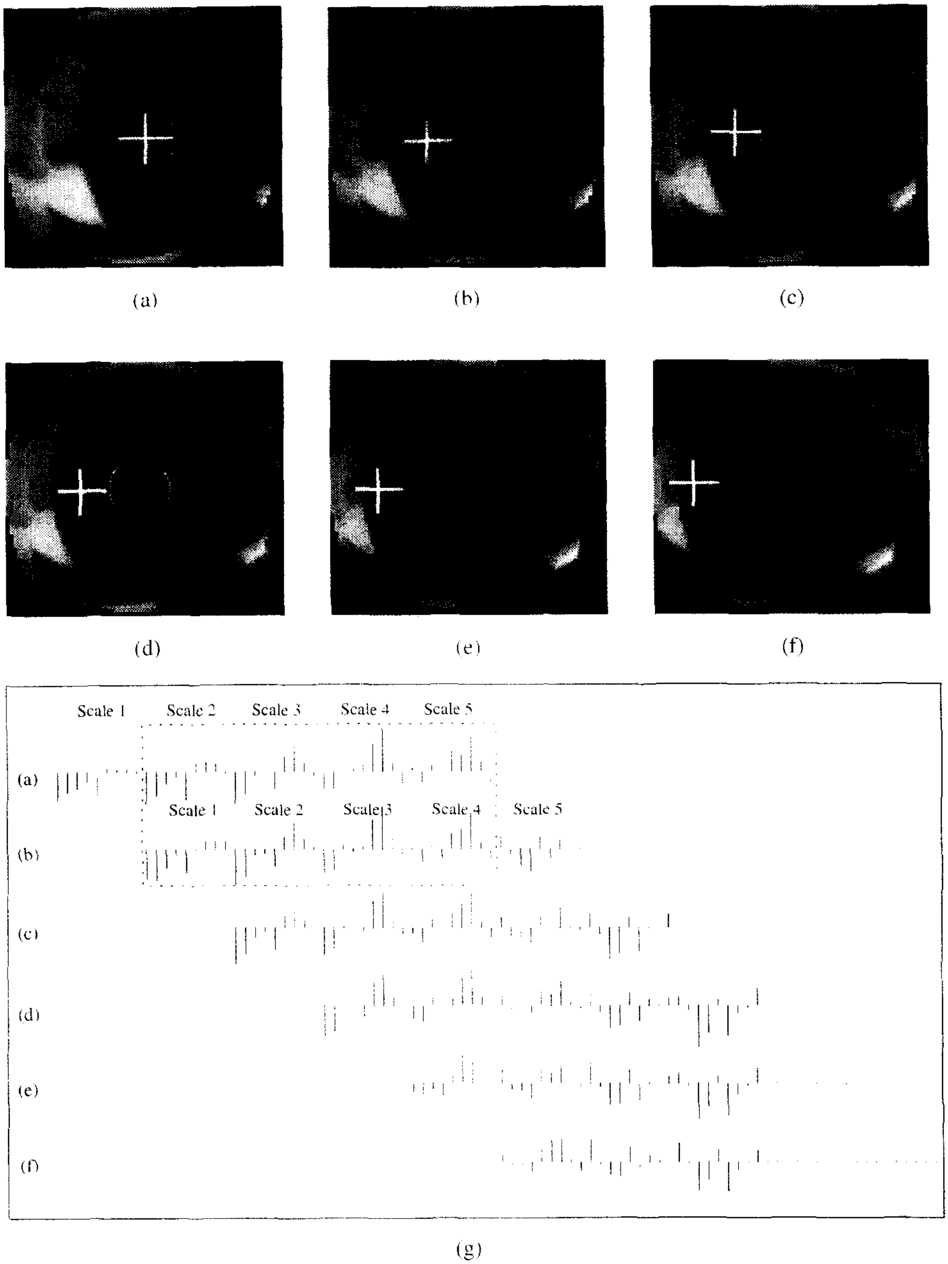


Fig. 17. Using response vectors with a log-polar sensor. (a) through (f) represent the sequence of "cortical images" in Cartesian coordinates obtained by movement of the sensor from a point (marked by a "+") in the foveal region (indicated by a circle) towards the right. (g) shows the interpolation required (in this case, shifting response histograms by one scale) for each new response vector for the original model point in order to maximize correlation with the initial model response vector as the point moves towards the periphery of the sensor. The obvious correlation between vectors as a function of changes in scale implies that the location algorithm will still work on peripheral targets once the scale correction is factored in.

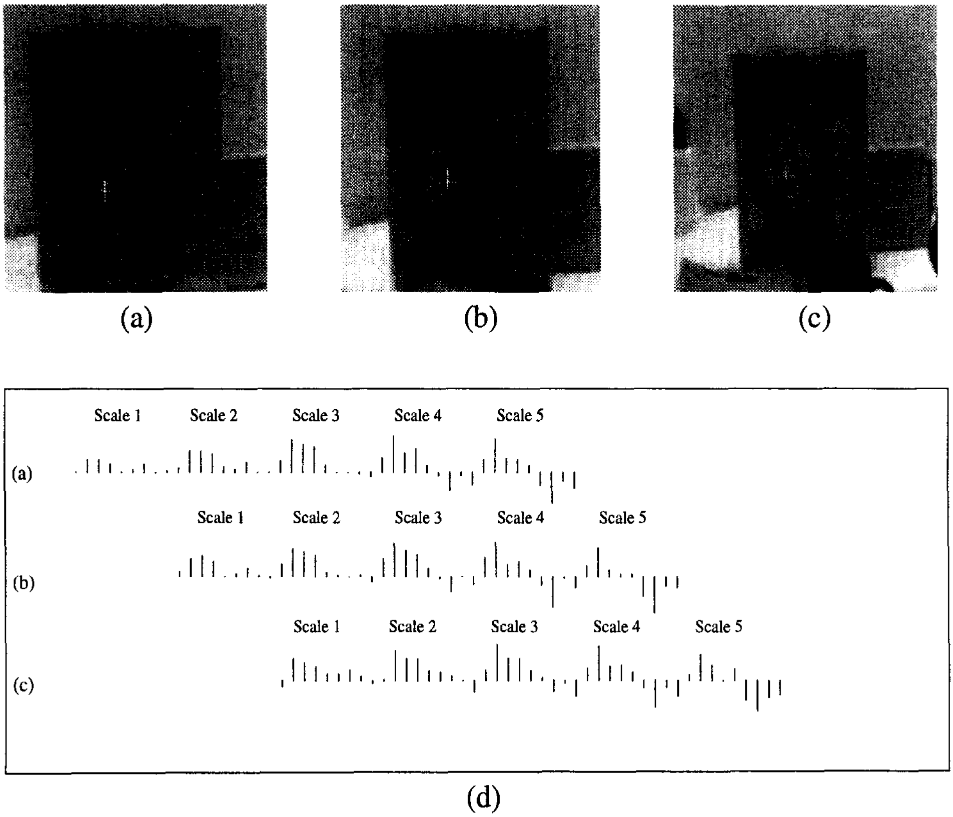


Fig. 18. Handling changes in scale. (a), (b), and (c) are images of an object at three different scales. (d) depicts the process of interpolating response vectors (for the point marked by "+") across scales. In this simple case, interpolation amounts to shifting response histograms to the right by one scale. The correlation between vectors as a function of changes in scale that is evident in (d) ensures that the visual routines will still work in the presence of scale changes once the scale corrections have been applied.

However, this problem can be tackled in at least three ways. First, since the approximate distances are usually known by the active vision system, and the dimensions of the viewed object are usually small compared to the viewing distance, the scale can be actively adjusted prior to the matching process by using, for instance, zoom lenses. Second, a strategy such as that used by Murase and Nayar [51] for scale normalization by subsampling or oversampling the image of an object to a canonical size can be adopted though this might introduce noise or other distortions due to the sampling process.

Perhaps the most feasible strategy for handling scale changes is the interpolate and compare method that was introduced in Section 4.7. The use of this procedure for handling scale variations is illustrated in Fig. 18, where interpolation amounts to a simple shift of response vectors to the right. The correlation between response vectors as a function of changes in scale that is clearly seen in Fig. 18(d) ensures that the visual routines will still work in the presence of scale variations once the necessary scale correction gets factored in. Note that this strategy also yields the approximate % change in scale from the original model scale as a by-product of the matching procedure and could thus be used in estimating current distance from the point in the 3D environment if desired.

Visual looming, or the expansion of the projection of an approaching object in the retina, is an important cue in the human reflex for obstacle avoidance [9]. Numerous approaches such as using flow field divergence [55], measuring optic flow via correlation [3], and computing the relative change in texture density [34] have been explored. The scale matching strategy used above suggests a natural and extremely simple strategy for tackling this problem. At any given point, if an object is approaching the camera, the filter responses at a coarse scale at time $t + \Delta t$ will roughly match those at a finer scale from time t. Thus, looming detection can be achieved by comparing, using scale interpolation, the response vectors extracted from points near the center of gaze from one frame to response vectors extracted from the same points in the successive frame(s); this procedure is illustrated in Fig. 19. The time-to-contact (or time-to-collision [44]) can likewise be estimated by measuring the scale change (amount of interpolation yielding the best match) induced by motion of the object. Experimentally evaluating the efficacy of these visual routines for looming and time-to-contact remains a topic for future research.

5. Conclusion

We have shown that an architecture that uses two banks of iconic filter vectors can form the core of an active vision system. Such an architecture allows the construction of a number of useful visual routines such as object location, object identification, looming detection and foveal targeting using log-polar sensors.

The architecture has a number of favorable properties that make it especially attractive:

- *It allows functional application.* The proposed architecture divides the general task of achieving complex problem solving behaviors related to scene interpretation into the complementary behaviors of location and identification (Section 1). This leads to considerable economy since it allows iconic feature vectors to be used in the context of simple task-directed programs or visual routines [77] which compute information on demand in lieu of scene reconstruction.
- *It benefits from the favorable matching properties that accrue to high-dimensional vectors.* The iconic descriptions used in the architecture exploit the tendency toward orthogonality inherent in high-dimensional spaces to achieve accurate indexing in the presence of noise in internal and external channels (see Section 2.1). This property allows them to be used in the context of a sparse distributed memory [36] that facilitates visual learning and offers a number of specific advantages over conventional modes of storage for visual memory [63,64].
- *It is well-suited for general-purpose object indexing.* The architecture employs object descriptions which are obtained by projecting image patches along the axes given by various derivative of Gaussian filters which are known to form the principal components of arbitrary natural images as described in Section 3.3.1. In addition,

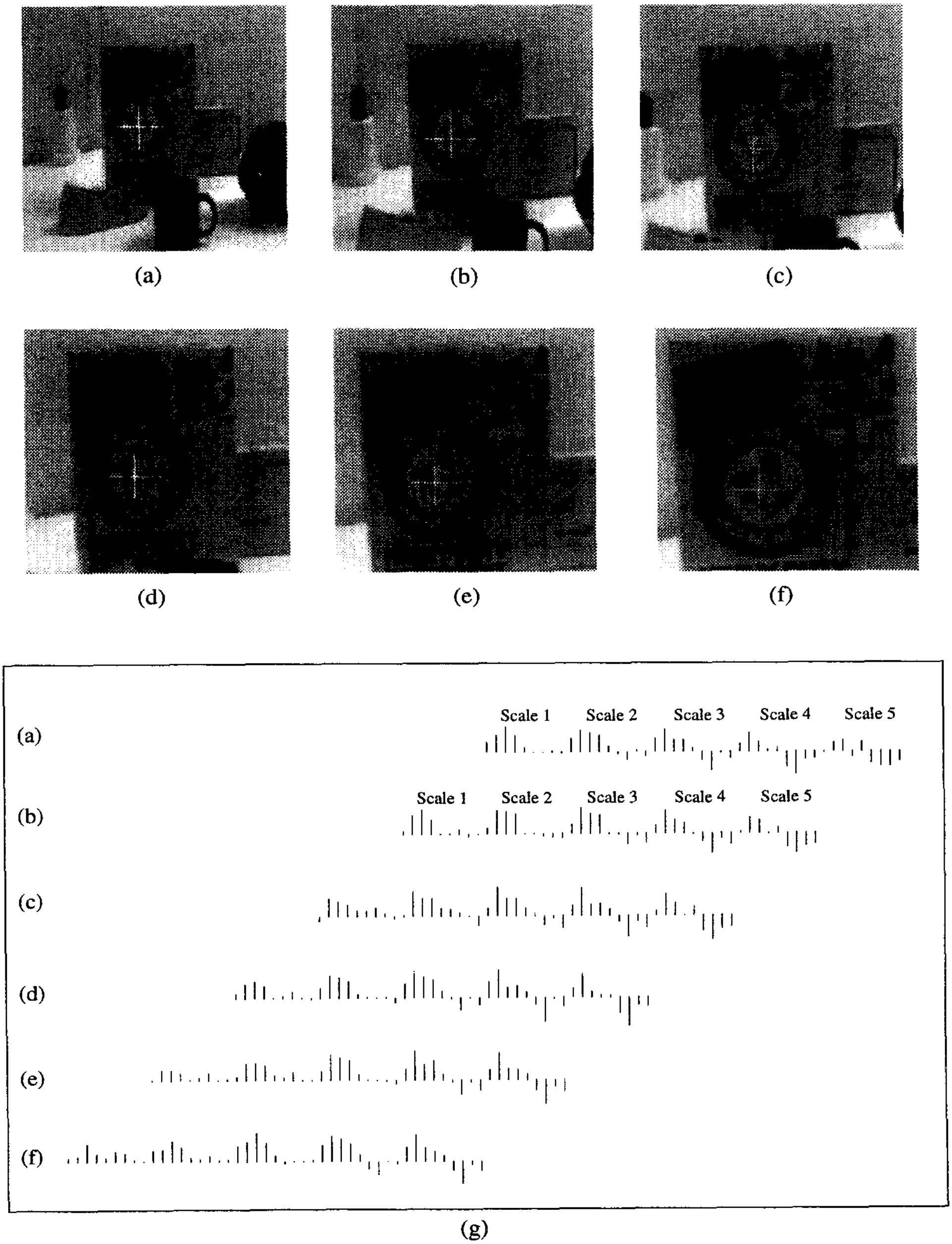


Fig. 19. Using iconic representations to detect looming. (a)–(f) show a sequence of images simulating the effect of a looming object. (g) shows the suggested procedure for interpolating and comparing responses across scales for determining best partial matches.

it is known that principal component expansion generates correlation filters that are statistically optimal in that they maximize signal-to-noise ratio and yield much sharper correlation peaks than traditional cross-correlation techniques [43].

- *It facilitates real-time implementations.* The models used in the architecture encode objects directly in terms of photometric codes; these can be computed much more efficiently than geometrical models. This wasn't possible until memory and correlation became cheap. In particular, the availability of pipeline image processors that can perform convolutions at frame-rate has made possible real-time implementations of photometric recognition architectures such as the one proposed here.

Some of the above properties are also shared by some recently proposed object recognition schemes. The location/identification dichotomy was used by Swain and Ballard in their work on indexing objects using color histograms [71]. Object representations in the form of high-dimensional feature vectors have recently been used by Mel [49] and Viola [79]. Mel, who is concerned exclusively with the object identification problem, uses color conjoined with edge/curvature detectors. He argues that multiple objects can be handled with additional features, whereas we propose that this problem is resolved by an encompassing active vision system which provides the necessary figure-ground segmentation. Viola also focuses on object identification and uses an index based on "complex" local features. Two other schemes employing filter-based vector representations are those of Daugman [19] and Buhmann et al. [11]. Daugman used multiscale 2D Gabor wavelets to generate long 256-byte "iris codes" for a human eye which he uses in a scheme for personal identity verification; he is however solely concerned with iris recognition rather than recognition of arbitrary objects. Gabor-based wavelets are also used by Buhmann et al. to form composite feature detectors called "jets," which are used in an elastic graph-matching strategy for recognition; like Mel, their emphasis appears to be on image interpretation where as our approach follows the relatively more tractable path of decomposing the problem into its location/identification components with an encompassing active vision system that provides the necessary figure-ground segmentation.

PCA-based basis functions have been used by Turk and Pentland [76] for face recognition, and by Murase and Nayar [51] for pose estimation and recognition. Our representation makes implicit use of PCA and its role in our representation is different from the above two approaches in that our representation includes non-orthogonal components since these allow an extremely simple rotation normalization procedure while similar ones for the above two approaches are not at all obvious; there has also been recent evidence suggesting that the directions of high-variance in natural scenes are not necessarily all orthogonal [24]. In addition, our basis functions are based on PCA of image patches at multiple scales—the use of image patches keeps the computational load constant and allows efficient strategies for active sensing while the use of multiple scales affords simple algorithms for scale invariance and space-variant sensing. Finally, we use fixed basis functions obtained after the process of PCA is taken to its limit by exposing the active vision system to a variety of images during an initial phase concerned solely with the development of the basis functions. As a result, the functions do not have to be recomputed upon the introduction of new objects as in the above two approaches.

A potentially problematic issue (at least in the case of the identification routine) is the assumption of some form of figure–ground segmentation. While there are certainly cases where segmentation may be extremely crude, the issue assumes lesser significance in light of the fact that even crude segmentation of the currently foveated object from the background will suffice in many cases for the routines described in the paper. For example, when identifying a currently foveated object using the circular template strategy, only approximate determination of the centroid is required due to tendency of nearby points to be highly correlated. Also, exact determination of object boundary is not essential since the feature vectors are extracted only from points lying within a specified distance from the approximate object boundary. Our current method relies on figure–ground segmentation using stereo. In cases where segmentation using stereo is not possible, alternate strategies can be employed such as motion (as illustrated by Murase and Nayar [51] in their recognition system), color [70], or texture [46]. In the latter case, the filter responses themselves can be used to implement a form of pre-attentive segmentation in the image region currently being foveated.

A second issue to be addressed is that of feasibility in terms of computation time. Our routines involving iconic representations are clearly computation-intensive but as mentioned above, the advent of real-time image processing hardware that can be readily exploited for both feature extraction and indexing purposes considerably lessens the relative importance of such objections. The location algorithm (Section 4), though employing brute-force search, is still able to compute best match locations close to real time in the current implementation on the Datacube MV200. The algorithm can however be further optimized, for instance, by employing a *coarse-to-fine* search strategy when comparing the multiscale filter responses of locations.

Our view-based approach to representing objects raises the question of scalability: will the method fail when extremely large model bases of objects are used and arbitrary 3D pose is allowed? It is however not hard to see that the use of more than one vector per object in conjunction with independent Kanerva memories for vectors from the different retinal locations potentially allows an extremely large number of objects to be handled. To see this, let $M = 1000$ represent the number of physical address locations available in a single memory. Kanerva [37] estimates the capacity of the memory to be about 5% of M i.e. 50 items. If $k = 25$ distinct vectors are used per object as we did in Section 4.5.1, the number of potentially distinguishable objects is $(0.05M)^k = 50^{25}$ which is an extremely large number. Even after ruling out a significant proportion (say 99.99%) of the possible combinations as being unlikely to be encountered in practice and factoring out the number of different views per object (say $36 \times 36 \times 36$ along the three principal axes), we are still left with an extremely large number (6.3×10^{33}), much larger than the number of objects that will probably be ever encountered by the system.

An ongoing effort involves the augmentation of the filter responses with color information. The Gaussian derivative filters were obtained as a result of PCA of achromatic natural images in the spatial plane. Using PCA in the R–G–B planes yields center-surround mechanisms in an opponent color space [20]. Thus, a spatio-chromatic iconic representation can be obtained by simply augmenting the current representation with responses from a variety of color-opponent Gaussian center-surround mechanisms such as, for instance, those obtained by subtracting responses of zeroth-order Gaussian filters applied to the red/green channels at different scales.

Other directions for future research include motion-based segmentation and recognition of objects, use of motorized zoom lenses for scale interpolation, and exploring the utility of temporal representations based on the outputs of spatio-temporal filters derived from localized PCA along the temporal axis.

Acknowledgments

We would like to thank the referees of the journal *Artificial Intelligence* for their incisive criticisms and detailed comments which greatly helped in improving the quality of the paper. Thanks are also due to Mary Hayhoe and Peter Lennie for useful pointers to the literature, Shree Nayar for access to the Columbia object data base, Randal Nelson, Lambert Wixson and Greg Zelinsky for three of the images used in the experiments, Garbis Salgian for implementing some of the first routines on the Datacube MV200 system, Jim Vallino for discussions regarding the MV200, and Lambert Wixson for help during the early stages of this work.

References

[1] E.H. Adelson and J. Bergen, Spatiotemporal energy models for the perception of motion, *J. Opt. Soc. Am.* **2** (2) (1985) 284-299.

[2] J. Aloimonos, A. Bandopadhay and I. Weiss, Active vision, *Int. J. Comput. Vision* **1** (4) (1988) 333-356.

[3] N. Ancona and T. Poggio, Optic flow from 1D correlation: application to a simple time-to-crash detector, AI Memo 1375, MIT AI Lab, Cambridge, MA (1993).

[4] R. Bajcsy, Active perception, in: *Proc. IEEE* **76** (1988) 996-1005.

[5] D.H. Ballard, Animate vision, *Artif. Intell.* **48** (1991) 57-86.

[6] D.H. Ballard, M.M. Hayhoe and P.K. Pook, Deictic codes for the embodiment of cognition, Technical Report 95.1, National Resource Laboratory for the study of Brain and Behavior, University of Rochester, Rochester, NY (1995).

[7] D.H. Ballard and L.E. Wixson, Object recognition using steerable filters at multiple scales, in: *Proceedings IEEE Workshop on Qualitative Vision* (1993).

[8] D.H. Ballard and R.P.N. Rao, Seeing behind occlusions, in: *Proceedings Third European Conference on Computer Vision (ECCV)*, Stockholm, Sweden (1994) 274-285.

[9] P.J. Beek, Perception-action coupling in the young infant: An appraisal of von Hofsten's research programme, in: *Motor Development in Children: Aspects of Coordination and Control* (Martinus-Nijhoff, Dordrecht, Netherlands, 1986) 187-196.

[10] T.O. Binford, Inferring surfaces from images, *Artif. Intell.* **17** (1981) 205-244.

[11] J.M. Buhmann, M. Lades and C. von der Malsburg, Size and distortion invariant object recognition by hierarchical graph matching, in: *Proc. IEEE International Conference on Neural Networks* **II**, San Diego, CA (1990) 411-416.

[12] F.W. Campbell and J.G. Robson, Application of Fourier analysis to the visibility of gratings, *J. Physiol. (Lond.)* **197** (1968) 551-566.

[13] J.F. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **8** (1986) 679-698.

[14] R.H.S. Carpenter, *Movements of the Eyes* (Pion, London, 1988).

[15] C. Chatfield and A.J. Collins, *Introduction to Multivariate Analysis* (Chapman and Hall, New York, 1980).

[16] K.M. Cohen, The development of strategies of visual search, in: *Eye Movements: Cognition and Visual Perception* (Lawrence Erlbaum, Hillsdale, NJ, 1981) 271-288.

[17] D.J. Coombs, Real-time gaze holding in binocular robot vision, Ph.D. Thesis, Technical Report 415, University of Rochester, Computer Science Department, Rochester, NY (1992),

[18] J.G. Daugman, Two-dimensional analysis of cortical receptive field profiles, *Vision Res.* **20** (1980) 447-456.

[19] J.G. Daugman, High confidence visual recognition of persons by a test of statistical independence, *IEEE Trans. Pattern Analysis and Machine Intelligence* **15** (11) (1993) 1148-1161.

[20] J.B. Derrico and G. Buchsbaum, A computational model of spatiochromatic image coding in early vision, *J. Visual Commun. Image Representation* **2** (1) (1991) 31-38.

[21] R.L. De Valois and K.K. De Valois, *Spatial vision* (Oxford University Press, New York, 1988).

[22] E.D. Dickmanns, An integrated approach to feature based dynamic vision, in: *Proceedings Conference on Computer Vision and Pattern Recognition* (1988) 820-825.

[23] K. Eberhard, M. Tanenhaus, M. Spivey-Knowlton and J. Sedivy, Investigating the time course of establishing reference: evidence for rapid incremental processing, in: *Proceedings Eight Annual CUNY Sentence Processing Conference*, Tucson, AZ (1995).

[24] D.J. Field, What is the goal of sensory coding? *Neural Comput.* **6** (1994) 559-601.

[25] W.T. Freeman and E.H. Adelson, The design and use of steerable filters, *IEEE Trans. Pattern Anal. Mach. Intell.* **13** (9) (1991) 891-906.

[26] D. Gabor, Theory of communication, *J. IEE* **93** (1946) 429-459.

[27] J. Grimes and G. McConkie, On the insensitivity of the human visual system to image changes made during saccades, in: K. Akins, ed., *Problems in Perception* (Oxford University Press, Oxford, 1995).

[28] W.E.L. Grimson, A. Lakshmi Ratan, P.A. O'Donnell and G. Klanderman, An active visual attention system to play "Where's Waldo", in: *Proceedings ARPA Image Understanding Workshop* (1994).

[29] P.J.B. Hancock, R.J. Baddeley and L.S. Smith, The principal components of natural images, *Network* **3** (1992) 61-70.

[30] D. Heeger, Optic flow using spatiotemporal filters, *Int. J. Comput. Vision* **1** (4) (1987) 279-302.

[31] B.K.P. Horn, The Binford-Horn linefinder, AI Technical Report 285, MIT AI Lab, Cambridge, MA (1971).

[32] B.K.P. Horn and B.G. Schunck, Determining optical flow, *Artif. Intell.* **17** (1981) 185-203.

[33] K. Ikeuchi and B.K.P. Horn, Numerical shape from shading and occluding boundaries, *Artif. Intell.* **17** (1981) 141-184.

[34] K. Joarder and D. Raviv, A new method to calculate looming for autonomous obstacle avoidance, in: *Proceedings Conference on Computer Vision and Pattern Recognition* (1994).

[35] D.G. Jones and J. Malik, A computational framework for determining stereo correspondence from a set of linear spatial filters, in: *Proceedings Second European Conference on Computer Vision*, Genova, Italy (1992).

[36] P. Kanerva, *Sparse Distributed Memory* (Bradford Books, Cambridge, MA, 1988).

[37] P. Kanerva, Sparse distributed memory and related models, in: M.H. Hassoun, ed., *Associative Neural Memories* (Oxford University Press, New York, 1993) 50-76.

[38] M. Kass, Computing visual correspondence, in: *Proceedings Image Understanding Workshop* (1983) 54-60.

[39] M. Kass, Linear image features in stereopsis, *Int. J. Comput. Vision* (1988) 357-368.

[40] H. Knuttson and G.H. Granlund, Texture analysis using two-dimensional quadrature filters, in: *IEEE Workshop on Computer Architecture for Pattern Analysis and Image Database Management* (1983) 206-213.

[41] J.J. Koenderink, Operational significance of receptive field assemblies, *Biol. Cybern.* **58** (1988) 163-171.

[42] J.J. Koenderink and A.J. van Doorn, Representation of local geometry in the visual system, *Biol. Cybern.* **55** (1987) 367-375.

[43] V.K. Kumar, D. Casasent and H. Murakami, Principal-component imagery for statistical pattern recognition correlators, *Optical Eng.* **21** (1) (1982) 43-47.

[44] D.N. Lee, A theory of visual control of braking based on information about time-to-collision, *Perception* **5** (1976) 437-459.

[45] J. Malik and Z. Gigus, A model for curvilinear segregation, *Invest. Ophthalmol. Vis. Sci. (Supplement)* **32** (4) (1991) 715.

[46] J. Malik and P. Perona, A computational model of texture segmentation, in: *Proceedings Conference on Computer Vision and Pattern Recognition* (1989) 326-332.

[47] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (W.H. Freeman, San Francisco, CA, 1982).

[48] J.H.R. Maunsell and W.T. Newsome, Visual processing in monkey extrastriate cortex, *Ann. Rev. Neurosci.* **10** (1987) 363-401.

[49] B. Mel, A neurally-inspired approach to 3-D visual object recognition, Presentation at Telluride Workshop on Neuromorphic Engineering, Telluride, CO (1994).

[50] M. Mishkin and T. Appenzeller, The anatomy of memory, *Sci. American* (1987) 80-89.

[51] H. Murase and S.K. Nayar, Visual learning and recognition of 3D objects from appearance, *Int. J. Comput. Vision* **14** (1995) 5-24.

[52] D.W. Murray, K.J. Bradshaw, P.F. McLauchlan, I.D. Reid and P.M. Sharkey, Driving saccade to pursuit using image motion, *Int. J. Comput. Vision* (submitted).

[53] K. Nakayama, The iconic bottleneck and the tenuous link between early visual processing and perception, in: C. Blakemore, ed., *Vision: Coding and Efficiency* (Cambridge University Press, New York, 1990) 411-422.

[54] K. Nakayama and S. Shimojo, Towards a neural understanding of visual surface representation, in: T. Sejnowski, E.R. Kandel, C.F. Stevens and J.D. Watson, eds., *Proceedings Cold Spring Harbor Symposium on Quantitative Biology* **55**: *The Brain* (1990).

[55] R.C. Nelson and J. Aloimonos, Using flow field divergence for obstacle avoidance in visual navigation, *IEEE Trans. Pattern Anal. Mach. Intell.* **11** (10) (1989) 1102-1106.

[56] M. Nitzberg, D. Mumford and T. Shiota, *Filtering, Segmentation and Depth* (Springer-Verlag, New York, 1993).

[57] P. Parent and S. Zucker, Trace inference, curvature consistency, and curve detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **11** (8) (1989) 823-839.

[58] A.P. Pentland, Shape information from shading: a theory of human perception, in: *Proceedings 2nd International Conference on Computer Vision*, Tampa, FL (1988) 404-412.

[59] A.P. Pentland, From 2-D images to 3-D models, in: K.N. Leibovic, ed., *Science of Vision* (Springer-Verlag, New York, 1990) 422-438.

[60] T. Poggio and F. Girosi, Networks for approximation and learning, *Proc. IEEE* **78** (1990) 1481-1497.

[61] R.P.N. Rao, Top-down gaze targeting for space-variant active vision, in: *Proceedings ARPA Image Understanding Workshop*, Monterey, CA (1994) 1049-1058.

[62] R.P.N. Rao and D.H. Ballard, Learning saccadic eye movements using multiscale spatial filters, in: G. Tesauro, D.S. Touretzky and T.K. Leen, eds., *Advances in Neural Information Processing Systems* **7** (MIT Press, Cambridge, MA, 1995)

[63] R.P.N. Rao and D.H. Ballard. Natural basis functions and topographic memory for face recognition, in: *Proceedings IJCAI-95*, Montréal, Que. (1995).

[64] R.P.N. Rao and D.H. Ballard, Object indexing using an iconic sparse distributed memory, in: *Proceedings International Conference on Computer Vision (ICCV)* (1995).

[65] R.P.N. Rao and D.H. Ballard, Object indexing using an iconic sparse distributed memory, Technical Report 559, Department of Computer Science, University of Rochester, Rochester, NY (1995).

[66] R.D. Rimey and C.M. Brown, Task-oriented vision with multiple bayes nets, Technical Report 398, Computer Science Department, University of Rochester, Rochester, NY (1991).

[67] T.D. Sanger, Optimal unsupervised learning in a single-layer linear feedforward neural network, *Neural Networks* **2** (1989) 459-473.

[68] D.G. Stork and H.R. Wilson, Do Gabor functions provide appropriate descriptions of visual cortical receptive fields? *J. Opt. Soc. Am. A* **7** (8) (1990) 1362-1373.

[69] G. Strang, *Linear Algebra and its Applications* (Harcourt Brace Jovanovich, San Diego, CA, 3rd. ed., 1988).

[70] M.J. Swain, Color indexing, Technical Report 360, University of Rochester, Computer Science Department, Rochester, NY (1990).

[71] M.J. Swain and D.H. Ballard, Color indexing, *Int. J. Comput. Vision* **7** (1991) 11-32.

[72] M.J. Swain, R.E. Kahn and D.H. Ballard, Low resolution cues for guiding saccadic eye movements, in: *Proceedings Conference on Computer Vision and Pattern Recognition* (1992).

[73] M. Tanenhaus, M. Spivey-Knowlton, K. Eberhard and J. Sedivy, Integration of visual and linguistic information in spoken language comprehension, *Science* (to appear).

[74] W.B. Thompson, Inexact vision, in: *Proceedings Workshop on Motion, Representation, and Analysis* (1986) 15-22.

[75] M. Tistarelli and G. Sandini, Dynamic aspects in active vision, *Comput. Vision, Graph. Image Process. Image Understanding* **56** (1) (1992) 108-129.

[76] M. Turk and A.P. Pentland, Eigenfaces for recognition, *J. Cognitive Neurosci.* **3** (1) (1991) 71-86.

[77] S. Ullman, Visual routines, *Cognition* **18** (1984) 97-160.

[78] L. Ungerleider and M. Mishkin, Two cortical visual systems, in: D. Ingle, M. Goodale and R. Mansfield, eds., *Analysis of Visual Behavior* (MIT Press, Cambridge, MA, 1982) 549-585.

[79] P. Viola, Feature-based recognition of objects, in: *Proceedings AAAI Fall Symposium on Learning and Computer Vision* (1993).

[80] J.W. Weber and J. Malik, Robust computation of optical flow in a multi-scale differential framework, Technical Report 709, Department of Electrical Engineering and Computer Science, University of California at Berkeley (1992).

[81] C.F.R. Weiman and G. Chaikin. Logarithmic spiral grids for image processing and display, *Comput. Graph. Image Process.* **11** (1979) 197-226.

[82] L. Wiskott and C. von der Malsburg, A neural system for the recognition of partially occluded objects in cluttered scenes: a pilot study, in: *IJPRAI* **7** (1993) 935-948.

[83] R.A. Young, The Gaussian derivative theory of spatial vision: Analysis of cortical cell receptive field line-weighting profiles, General Motors Research Publication GMR-4920 (1985).