COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 63, No. 1, January, pp. 50–65, 1996
ARTICLE NO. 0004

An Active Foveated Vision System: Attentional Mechanisms and Scan Path Covergence Measures

HIROYUKI YAMAMOTO

Center for Intelligent Machines, McGill University, 3480 University Street, Montréal, PQ, H3A 2A7 Canada; and Media Technology Laboratory, Canon Inc., 890-12, Kashimada, Saiwai-ku, Kawasaki, Kanagawa 211, Japan

YEHEZKEL YESHURUN

Department of Computer Science, Tel Aviv University, 69978 Tel Aviv, Israel

AND

MARTIN D. LEVINE

Center for Intelligent Machines, McGill University, 3480 University Street, Montréal, PQ, H3A 2A7 Canada

Received October 24, 1993; accepted September 26, 1994

We present a testbed implementation for a foveated robot vision system. To achieve foveation, the visual sensor simulates nonuniform sampling and is mechanically directed toward a specific fixation point. An interest operator is used to select a sequence of fixation points. Successive snapshots of high foveal and low peripheral resolution are combined to create a wide-angle representation of the scene. Such visual processing has been studied with primate and human subjects from a biological point of view. The purpose of this work is to create an active vision system with which we can study foveated machine vision. The current implementation incorporates a CID camera positioned by a PUMA robot to pan and tilt around a fixed point, a SIMD parallel computer, and conventional computers to construct gray-level, edge, and interest maps from several fixations. The system is highly modular, and its architecture permits the efficient incorporation of sequential and parallel components for real-time operation. We demonstrate the modularity of the system and its potential as a testbed for active vision by incorporating two different attentional mechanisms and quantitatively evaluating their performance on artificial and natural images. We propose three types of norms that can be used for this performance evaluation. © 1996 Academic Press, Inc.

1. INTRODUCTION

Where do we focus our attention next? When faced with a new environment, we view it by moving our eyes to fixation points (*saccadic* eye movements) along a *scan-path* at the approximate rate of three per second [1]. After a short time—a few seconds at most—we have access to visual data which represent a field of view of approximately 200°. This somewhat small amount of time is achieved by directing computational resources to selected areas only. To accomplish this, we possess movable space-variant visual sensors (our eyes) which combine a high-resolution central fovea with decreasing resolution in the periphery [2]. Thus high-resolution processing is applied only where necessary. It would require 1000 times more pixels to represent the whole field of view at the highest (*foveal*) resolution than the number needed for the actual space-variant sensor. This perspective implies a remarkable reduction of data and computational resources. How should this observation be taken into account for machine vision? Where should the robot look next? If we adopt the philosophical position that biological vision is the upper limit of a feasible mechanism for sensory information processing, then we have a clear motivation to produce biologically motivated solutions in the context of machine vision. Until recently, machine vision applications have used only uniform resolution and multiresolution images [3]. However, this is changing, as some vision systems with nonuniform sampling (*foveated vision*) have appeared [4, 5], as well as active vision systems [6, 7] and mechanical systems to support sensory movement (*robot heads*) [8, 9].

Following each saccade, a foveated vision system faces at least two basic tasks (the so-called what?/where? dichotomy): (i) to analyze the information in the fovea and (ii) to select the next gaze position from data projected onto the peripheral region of the retina (*focus-of-attention*). The latter task is one of the major characteristics that distinguishes a foveated vision system from a conventional machine vision system. Foveated vision is heavily dependent

1077-3142/96 $18.00

on this mechanism. A realistic biological model of focus-of-attention is beyond the scope of current computational vision, since this process is significantly context-dependent and involves high-level processes of pattern analysis and object recognition. Similarly, a computer vision implementation of high order gaze control also requires specific knowledge of and expectation about the objects in the environment [10, 11]. However, we believe that it is still of interest to study context-free, pattern-based image operators (*interest operators*) which select regions of interest by invoking only low-level vision computations. Using the context-free approach, many researchers have defined interest points as points of high curvature (e.g., [12, 13, 14]), while others have suggested a measure of "busyness" involving the smoothed absolute value of the Laplacian of the data [15] or rapid changes in image gray levels [16]. It has also been suggested that generalized symmetry is a more general concept for defining such points [17]. However, until now these interest operators have only been applied to the usual uniformly sampled images. In our work, we study the problem of robot focus-of-attention using foveated eye-in-hand sensors.

Another characteristic of human foveated vision is that the visual system gathers partial information about the surrounding environment by moving the focus-of-attention and then determining the next step for the specific task. While a single image is clearly represented in an "iconic" (though spatially transformed) manner on the *retina* and in the *primary visual cortex,* it is uncertain whether successive fixations are actually integrated to yield an "integrative iconic buffer" in our brain [18, 19, 20]. In the machine vision context, however, it is clearly useful to have such a buffer in order to use existing object recognition methods. Since information-merging methods and data structures using a sequence of gaze positions have not been investigated in great detail in the context of machine vision, further examination with an actual foveated machine vision system is desirable.

In this paper, we present a testbed implementation of our system, called **Fovia** (FOVeated Image Application), which will be used to study the applicability of visually guided foveated gaze control for machine vision. The system incorporates a CID camera positioned by a PUMA robot to pan and tilt around a fixed point in 3D-space, a SIMD parallel computer which simulates a foveated sensor, and conventional computers to construct gray-level, edge, and interest maps obtained from several fixations. Figure 1 shows a scan path obtained by using our system. This partial drawing by Paul Klee was used by Noton and Stark over 20 years ago in their pioneering research on human focus-of-attention. Although we do not claim to actually simulate human perception, nevertheless the gaze paths exhibited by humans and our system are surprisingly similar. This kind of scan path can serve as a subjective evaluation of the attentional algorithm. However, it is also useful to have an objective estimate in order to quantitatively compare different attentional mechanisms. Thus as an experimental tool, we propose three types of norms that are used for this performance evaluation. Section 2 of this paper describes the system configuration and details each of the modules. In Section 3, experimental results with focus-of-attention mechanisms as well as the performance evaulation method are discussed. Discussions and conclusions are given in Section 4.

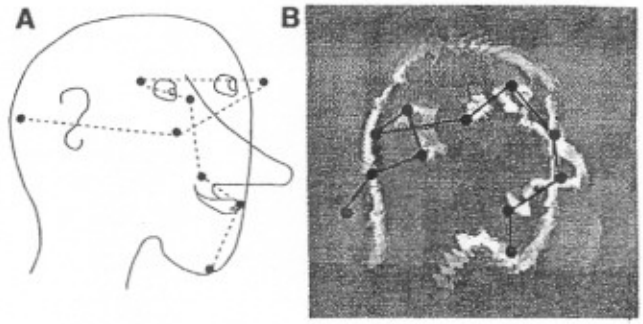

FIG. 1. Scan paths. (A) A human observer. (B) This system.

2. SYSTEM IMPLEMENTATION

Figure 2 shows the current **Fovia** system configuration. In our current research we are primarily interested in gaze control and image integration. Thus we have implemented space-variant sampling, context free mechanisms for focus-

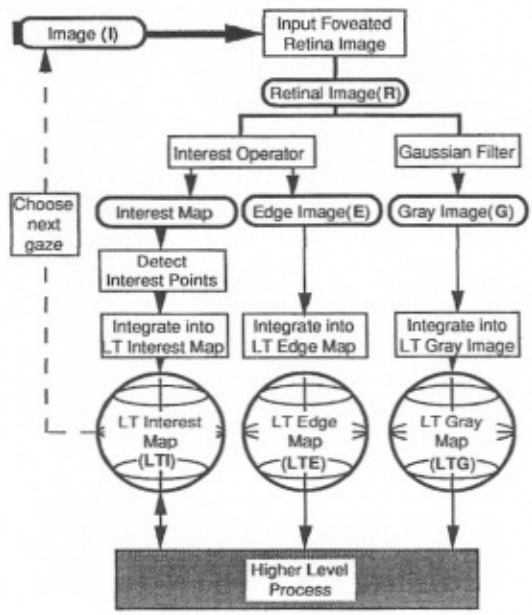

FIG. 2. **Fovia** system configuration.

ıttention, image integration over saccades, system cali-
tion, and robot control. In this paper, we reserve the
n "image" for the data acquired by a single fixation of
camera, and "map" for the data merged by images
:ained from several fixations. The system iterates the
lowing steps:

(1) Input a wide-angle gray-level image **R** according the space-variant sampling function.

(2) Apply the focus-of-attention mechanism to **R**.

(3) Calculate a foveated gray-level image **G** and a fove-
:d edge image **E** from **R**.

(4) Merge **G** and **E** into unified long-term maps yield-
g a gray-level map **LTG** and an edge map **LTE**.

(5) Update an interest map **LTI** by recording the reso-
tion (as a measure of interest) with which the cell at the
)rresponding point in **LTG** and **LTE** is stored. This map
modified by a focus-of-attention mechanism described
Section 2.3.2.

(6) Select the next gaze position according to the spe-
fic attentional mechanism and **LTI**.

(7) Move the camera to the new selected position.

Currently, Step (4) and (5) are performed by conventional
/orkstations (Silicon Graphics Personal Iris); the other
teps are run on a SIMD parallel machine (MasPar), con-
ıected to a Datacube frame-grabber and PUMA robot
:ontroller. These computers communicate with each other
ısing a specified protocol. However, as long as the different
nodules use the specified communication protocol, they
:an be developed separately and complied and run on
ıny available machine. For example, the image integration
module was developed independently of the other mod-
ules. It supports the specified communication protocol and
is a stand-alone process on the workstation. This is true for
both robot control and top-down gaze control processes,
although these are not discussed in this paper.

The main components of the system are space-variant image sampling, bottom-up focus-of-attention, and integrative iconic memory. In the following, we describe each component in detail.

2.1. *Space-Variant Sampling*

The human monocular field of view is 208° [21]. The combined binocular field of view is also 208° due to the overlap in the monocular representation. Visual data flows from the retina to the striate cortex, via the lateral geniculate nucleus. Since each hemisphere of the cortex processes only one visual hemifield, a single hemispherical cortical image spans only 104°. From a practical point of view, a 208° lens for the retina is not feasible. Thus we have arbitrarily restricted ourselves to a field of view about half this size, which spans about 100°.

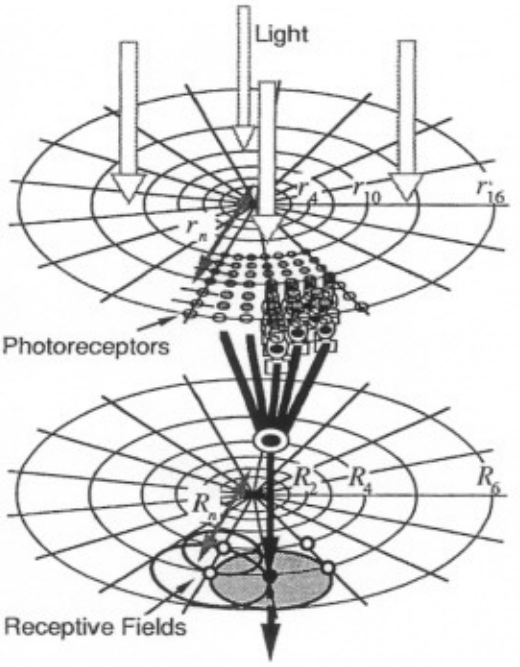

FIG. 3. Space-variant image sampling.

The spatial sampling that covers this relatively wide angle of view is not well defined, since the visual system cannot be readily described in terms of pixels. In general terms, however, it is possible to describe two main retinal areas according to their spatial cone density, a foveal (up to 3°) area with constant sampling rate and a peripheral area where the sampling rate decreases according to a power law [22]. To simulate this biological aspect in the machine vision context, we define a computational model of image sampling. In the following subsections, we describe our model of space-variant image sampling and its implementation in the system.

2.1.1. *Sampling model.* The log-polar mapping model is widely used as a model of space-variant image sampling in machine vision, since it is rotation- and scale-invariant, is useful for pattern matching [23], and conforms well with the psychophysical evidence [2]. Wilson [24] proposes a model for such a space-variant receptive field arrangement and explains how the human *contrast sensitivity function* arises from it. However, he makes no distinction between the photoreceptors and the receptive fields. Since the ratio of cones (photoreceptors) to ganglion cells (receptive field) is 1:1 in the fovea and definitely larger in the periphery [25], this difference should be considered. Therefore, we have modified his model to include space-variant sampling of both photoreceptors and receptive fields by invoking the log-polar mapping.

In our sampling model, which is shown in Fig. 3, the

receptive fields in the periphery are arranged in a similar fashion to Wilson. In the periphery, (i) receptive field size increases as a linear function of eccentricity and (ii) spacing between adjacent receptive fields is such that the diameters of the two fields overlap by a constant fraction, about 50%. Thus we have the following:

- In the periphery, the center of a receptive field is located on a ring the center of which is at the fovea.
- The eccentricity of the nth ring R_n is

$$R_n = R_0\left(1 + \frac{2(1 - o_v)c_m}{2 - (1 - o_v)c_m}\right)^n, \qquad (2\text{-}1\text{-}1)$$

where

R_0 is the radius of the fovea (in degrees).
c_m is the ratio of the diameter (in degrees) of the receptive field to the eccentricity (in degrees) of that receptive field from the center of the fovea.
o_v is the overlap factor. If the receptive fields touch each other, $o_v = 0$. If they extend to the center of the next field, $o_v = 0.5$.

The above equation is equivalent to $R_n = R_0 e^{wn}$,

where

$$w = \log\left(1 + \frac{2(1 - o_v)c_m}{2 - (1 - o_v)c_m}\right). \qquad (2\text{-}1\text{-}2)$$

- The radius of the receptive field on the nth ring is $\frac{c_m \cdot R_n}{2}$. (2-1-3)
- The number of receptive fields is $\frac{2\pi}{c_m(1 - o_v)}$ per ring. (2-1-4)

We also use a log-polar mapping to describe the arrangement of the photoreceptors within these receptive fields. However, in this case we assume the following: (i) The density of the photoreceptors in the fovea is equal to that just outside it. (ii) In the fovea, each photoreceptor possesses its own output and thus functions independently as its own receptive field. (iii) The photoreceptors are distributed uniformly in the fovea. (iv) Each receptive field forms a disk of radius p photoreceptors and therefore contains the same number of photoreceptors. This is the number of photoreceptors within a disk-shaped window of dimensions $(2p + 1) \times (2p + 1)$ placed around the center of the receptive field. From the above assumptions, we obtain:

- The eccentricity of the nth ring of photoreceptors r_n in the periphery is $r_n = R_0 e^{\frac{n \cdot w}{p}}$. (2-1-5)
- The number of photoreceptors in the periphery is $\frac{2\pi \cdot p}{c_m(1 - o_v)}$ per ring. (2-1-6)
- The number of photoreceptors in the fovea is $\left(\frac{2 \cdot p}{c_m(1 - o_v)}\right)^2$. (2-1-7)

In the fovea, every photoreceptor has its own private channel to a unique ganglion cell. This ratio of 1:1 between photoreceptors and ganglion cells increases in the periphery in a continuous manner. In our system we simulate this sampling function by defining two distinct regions; a foveal, uniformly sampled region and a peripheral region that is processed as previously described.

2.1.2. *Implementation of nonuniform sampling.* In order to implement our specific choice of sampling function, it is possible to select one of the following mechanisms[1];

- Optics. A custom-made lens can simulate our model and produce an image that will be projected on a uniformly sampled CCD array.
- Analog VLSI. A nonuniform CCD array can produce our foveated image with a commercially available lens [26].
- Video-rate filter. A normal video signal can be converted to the foveated image using special hardware [27].
- Software filter. Conversion software can produce a foveated image from a uniformly sampled image [5].

Although none of these can achieve enough resolution to simulate biological vision, we have chosen the last methodology. It is more flexible than the others and better fits the general idea of our system, that is, testing and experimenting with different modules and algorithms for active vision. In our system, each PE (processor element) in the SIMD parallel machine can independently and in parallel access a pixel stored in the frame grabber. Thus, simulating our sampling model is fast enough for many applications.

A wide-angle lens (Cosmicar C30402) and CID camera (CIDTEC CID2250) with exactly 512 × 512 square pixels, provide us with over a 100° viewing angle and a uniformly sampled image. The system resamples the photoreceptors (**R** in Fig. 2) from this image. The data in the fovea and in the periphery are stored on the PEs in different ways. Each PE contains one photoreceptor in the fovea; on the other hand, the periphery data are stored so that each PE has $p \times p$ photoreceptors. This scheme is shown in Fig. 4. To obtain the receptive field response (**G** and **E** in Fig. 2), Gaussian filters and the methods described below in Section 2.2 are employed. The results are also stored in the PEs, so that each PE contains one receptive field output. For example, the foveal area is taken to be the central

[1] Note that the first cases cannot include overlapping receptive fields.

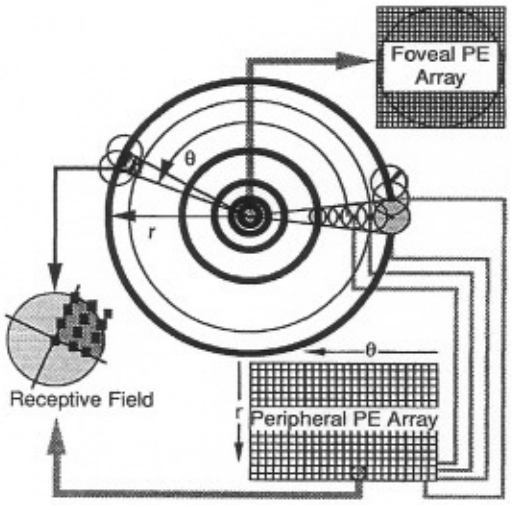

FIG. 4. Data structure for space-variant image.

54×54 photoreceptors of the uniformly sampled image, while the peripheral area contains 51×150 photoreceptors and 27×50 receptive fields, given $c_m = 0.25$, $o_v = 0.5$, $p = 3$.

2.2. *Focus of Attention*

The idea of directing computational resources to locations where they are mostly required is obviously based on the behavior of living organisms. While it is clear that this is the strategy of choice for primates, among others, it is far from obvious what are the visual cues that attract attention over the many dimensions of vision. In [28], Treisman describes a "Feature Integration Theory of Attention." In her conception, separable features such as a color and orientation are registered "early, automatically, and in parallel across the visual field," thereby forming "preattentive feature maps." Spatially focused attention is required to combine the different features. On the other hand, in [29], Hillyard *et al.* state that "the selection of different cues is initiated with latency that depends on their complexity, in approximately the following order: location, contour, color, spatial frequency, orientation, and finally conjunction of these features." According to this model, each feature should be examined to see whether it can determine the next eye position by itself, before combining it into a unified interest map.

While visual attention is a major research topic in psychology, neurobiology (see [30] for a review), and the computational [31, 32] aspects of vision, most of the work in this area relates to *covert attention*. This term refers to a situation where the gaze is fixed on a single image and the focus of attention moves covertly within that image. In our research, we are concerned with *overt attention* where the gaze is physically shifted to view a sequence of images according to some meaningful criterion. We do not seek to model the actual human visual attention process but rather to explore the use of context-free, contour- and edge-based overt attention.

By limiting the scope of attention to the visual dimensions of contour and edge data, it is clear that, generally speaking, eye movements are strongly task-, expectation-, and context-dependent [33]. On the other hand, when tracing the eye movements of newborns, it seems that there exist certain "hard wired" context-free attentional mechanisms, such as fixating corners or borders of objects [34] or around a single prominent edge feature [35]. Thus, it seems plausible that earlier, low-level, context-free mechanisms are gradually merged with more context-dependent algorithms, as more and more knowledge about the world becomes available. Thus it is very natural for a system like ours to initially employ context-free attentional mechanisms as a core for more elaborate, context-dependent algorithms.

With respect to machine vision, researchers have considered points of high curvature ([12, 13, 14]), rapid changes in image gray levels [16], and generalized symmetry [17] as features that might serve as "interest areas" and can be detected with low-level vision algorithms. We have chosen to implement the following two interest operators to compare their performance: corner detection and generalized symmetry [17]. These interest operators are applied to the peripheral region of the current image **R** and the next gaze position is selected according to the response. The details of these operators are described in the following subsections. Though these operators provide us with interest points based on the current gaze direction, we still require a focus-of-attention mechanism to create the link between this interest point information and the currently accumulated information. We describe this interaction in Section 2.3.2.

2.2.1. *Corner detector.* The most commonly used corner operator is the Moravec operator [36]. This operator calculates the sums of the squares of the differences of pixels in each of the four directions over a small window. Then the minimum of these variances is computed as the value of the interest operator. Finally, the interest points are obtained from the local maxima of these values. This operator is widely used to obtain interest points because of its simplicity. Here we have chosen a method motivated by a set of representations obtained from orientationally selective receptive fields (*simple cells*) in the visual pathway [37]. Electrophysiologically it has been reported that simple cells respond maximally to straight-line stimuli with a given orientation and location; if either of these is varied, then the response drops. Thus a set of simple cells at each image point can represent potential multiple orientations at that point corresponding to a corner or other complex

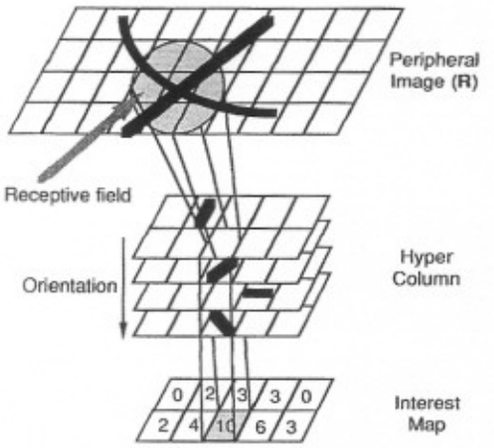

FIG. 5. Corner interest operator.

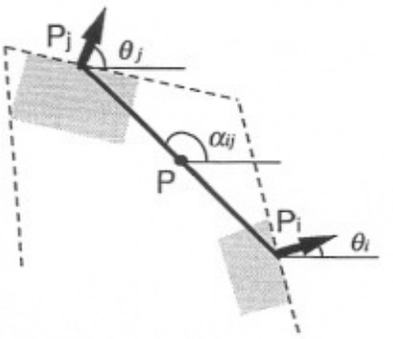

FIG. 6. Symmetry interest operator.

feature. In [37], Zucker uses an elongated difference-of-Gaussians to model these simple cells.

In our implementation, we use a simplified version of simple cells, as shown in Fig. 5. The essential characteristics of this approach are: (i) the receptive field models a step-edge template for a specified direction, (ii) these templates, for each direction, are applied to the image data **R** by a process of convolution, (iii) the responses characterize the orientational representation, and (iv) the corners are detected by choosing points that indicate multiple responses. Mathematically these templates are specified by a collection of operators,

$$T_\theta(i,j) = \begin{matrix} 1.0 & \text{if } \sin(\theta)i + \cos(\theta)j > 0.0 \\ 0.0 & \text{otherwise,} \end{matrix} \qquad (2\text{-}2\text{-}1)$$

where θ indicates the orientation of the operator and p indicates an image position. Since the templates are discrete, the orientation θ is quantized into eight different directions θ_k ($k = 1 \ldots 8$). Each of these can be interpreted as a model of an edge passing through its spatial support. After applying these eight templates to the nonuniformly sampled image, we have eight outputs $E_k(p)$ at each spatial location p. These $E_k(p)$ are the elements of the orientational representation of the scene. We can obtain corners by picking those locations that have multiple responses, that is where two template outputs are above a predefined threshold. In a similar fashion, we can also determine the edge at each location by selecting the maximum response from the orientational representation, that is, $\max_{k=(1\ldots8)} E_k(p)$. In the past, corner detectors have been defined for conventional uniformly sampled images. However, in general, orientations are not preserved under nonuniform sampling. Nevertheless, since log-polar sampling function is a conformal mapping, local orientations are still preserved. Thus, by restricting the interest operator to local neighborhoods, we can still use the conventionally defined operator for foveated images. The orientation representation is also useful for detecting curves and other image features [38].

2.2.2. *Symmetry detector.* Since natural and artificial objects give rise to a certain measure of coarse symmetry, it has been demonstrated [17, 39] that the peaks of the activity map produced by this operator are useful for detection of regions of interest. We believe that this is also true for the nonuniformly sampled image. In [17], the symmetry operator defines the measure of symmetry $S_\sigma(p, \psi)$ at each pixel position p in direction ψ as by the following equations (see Fig. 6):

$$S_\sigma(p,\psi) = \sum_{(i,j)\in\Gamma(p,\psi)} D_\sigma(i,j)P(i,j)r_i r_j, \qquad (2\text{-}2\text{-}2)$$

where

$$\nabla p_k = \left(\frac{\partial}{\partial x}p_k, \frac{\partial}{\partial y}p_k\right) \qquad (2\text{-}2\text{-}3)$$

$r_k = \log(1 + \|\nabla p_k\|)$ is the edge strength at position p_k (2-2-4)

$\theta_k = \tan^{-1}(\nabla p_k)$ is the direction at position p_k (2-2-5)

$$\Gamma(p,\psi) = \left\{(i,j)\middle|\frac{p_i + p_j}{2} = p, \frac{\theta_i + \theta_j}{2} = \psi\right\} \qquad (2\text{-}2\text{-}6)$$

$$D_\sigma(i,j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-|p_i - p_j|/2\sigma} \qquad (2\text{-}2\text{-}7)$$

$$P(i,j) = (1 - \cos(\theta_i + \theta_j - 2\alpha_{ij}))(1 - \cos(\theta_i - \theta_j)). \qquad (2\text{-}2\text{-}8)$$

The isotropic symmetry operator $S(p)$ is then defined by

$$S(p) = \int_0^{2\pi} S_\sigma(p, \psi)d\psi \qquad (2\text{-}2\text{-}9)$$

The local maxima of $S(p)$ give us the interest points in the image. As described above, this method does not require object segmentation or knowledge about the scene. Hence it is applicable to natural scenes containing multiple objects and can be used after the edge detection stage.

In our implementation, we have slightly modified the above equations to use the orientation representation described in Section 2.2.1. The measure of symmetry $S'_\sigma(p, \psi)$ at each pixel position p in the direction ψ is redefined by the equation

$$S_\sigma(p, \psi) = \sum_{(i,j)\in\Gamma(p,\psi)} \sum_{l,m=1\ldots8} D_\sigma(i,j) \cdot P'(i,j,l,m) \cdot E_l(p_i) \cdot E_m(p_j), \qquad (2\text{-}2\text{-}10)$$

where

$$P'(i,j,l,m) = (1 - \cos(\theta_l + \theta_m - 2\alpha_{ij}))(1 - \cos(\theta_i - \theta_m)) \qquad (2\text{-}2\text{-}11)$$

l and m are the indices of the orientation at image locations p_i and p_j, respectively,

$E_l(p_i)$ and $E_m(p_j)$ are the strengths of the orientational responses at image locations p_i and p_j respectively, as described in Section 2.2.1.

The isotropic symmetry operator $S'(p)$ is still defined by Eq. (2-2-9). We use the local maxima of this operator as the interest points in the scene. Again, as is the case for the corner detector, the local implementation of the operator approximates the desired results due to the fact that local angles are preserved.

2.3. *Integrative Iconic Memory*

Though there is no evidence for the existence of an iconic buffer in primates, it is obviously necessary for an active, foveated vision system to maintain an up-to-date image of the whole environment, integrated from the single foveated images of single fixations [40]. Thus, we have chosen a data structure and merging algorithm for integrative maps as described below. Gray-level (**G**) and edge (**E**) maps are transferred to this module and merged into the integrative maps **LTG** and **LTE**, respectively.

2.3.1. *Data structure.* In our current experimental implementation, we restrict the camera movements to pan and tilt around a fixed point in 3D space. This restriction is useful for quick eye movements (saccades) that do not involve visual feedback. Once the next gaze position is decided using the focus-of-attention mechanism, we can determine the direction of the target and rapidly position the fovea on it independently of the distance to the object. On the other hand, to support camera translation, we would need the distance between the current camera position and the target to determine the amount of translation in order to bring the target onto the fovea. Unless we have the distance, perhaps by using a rangefinder, we need visual feedback to check whether the target actually appears in the fovea after we move the camera.

From these assumptions, we observe that a spherical surface is much more appropriate than a planar 2D array for storing the images from different view directions. In order to implement a spherical data structure, we need to tessellate it into cells that are preferably symmetric, identical in shape and size, and controllable in specific resolution of tessellation. However, it is known that there is no tessellation method that satisfies these characteristics. There exist only five regular solids for tessellating a sphere into faces, but these do not provide enough resolution. The best that can be achieved is the well-known geodesic data construction as follows [41]: (i) use one regular solid, (ii) subdivide each face into surfaces by dividing each edge of a face into f (frequency of geodesic division) sections, and (iii) project the subdivided faces onto the sphere.

Several data structures for this geodesic dome have been proposed. Horn [42] has used a hierarchical tree structure. Fekete and Davis [43] adopted a complex node labeling scheme for a quadtree data structure to represent the tessellated sphere using an icosahedron. Chen and Kak [44] have proposed an alternate method for storing an object-centered feature map. In their approach, logical adjacency between elements of the data structure corresponds to physical adjacency between the cells on the sphere. Since we need to project a receptive field onto a circular region, this point is very important for fast processing. Therefore we have used their data structure for the integrative maps.

The degree of tessellation (that is, f) is determined by the space-variant sampling model described in Section 2.1. In the following, we assume that we need the same spatial resolution on the geodesic dome as exists in the fovea. In [42], Horn writes that a lower bound on the angular spread θ (the half-angle of the cone formed by the hexagonal disc) for a tessellation with n cells is

$$\theta = \sqrt{\frac{8\pi}{3\sqrt{3}n}} \times 180/\pi \text{ [degrees]}. \qquad (2\text{-}3\text{-}1)$$

From (2-1-7), the radius of the fovea (R_0) corresponds to

$$N_f = \frac{2 \cdot p}{c_m(1 - o_v)\sqrt{\pi}} \text{[photoreceptors]}. \qquad (2\text{-}3\text{-}2)$$

Since the angular spread of a photoreceptor is

$$\theta = \frac{R_0}{N_f}, \tag{2-3-3}$$

we get $n \cong 1{,}293{,}878$ by substituting for θ in Eq. (2-3-1) using Eq. (2-3-3), and for N_f using Eq. (2-3-2), given $R_0 = 3.0°$, $c_m = 0.25$, $o_v = 0.5$, $p = 3$. That is, we need 1,293,878 cells to provide enough resolution for mapping images onto the geodesic dome. Since in our implementation [44] we have

$$n = 10 \times f^2 + 2, \tag{2-3-4}$$

where f is the frequency of geodesic division, we therefore require $f \cong 360$ in order to obtain the necessary number of cells.

We have implemented this data structure and a merging algorithm based on a "winner-take-all" algorithm [14]. The **G** and **E** images are transferred to this module and merged into the long-term maps **LTG** and **LTE**, respectively. Simultaneously, the integrative interest map **LTI** is modified so that the cell at point p in the map **LTI**, $LTI(p)$, records the resolution with which the cells at the corresponding points in **LTG** and **LTE**, $LTG(p)$ and $LTE(p)$, are stored. Since points with low resolution are likely to be investigated next by the system, LTI represents the measure of interest by the resolution, and its local maxima indicate the previous fixation points.

2.3.2. *Merging algorithm.* For a given fixation point, there exist corresponding foveated images **G** and **E** of the original scene. Thus successive fixations provide different representations of the same scene. We employ the winner-take-all algorithm to integrate these representations into one integrative map [14]. In this algorithm a cell in the integrative maps at point p is replaced by new data only if the resolution of the new information, $r(p)$, is higher than the current value of $LTI(p)$. In this case, $LTG(p)$, $LTE(p)$, and $LTI(p)$ are replaced by $G(p)$, $E(p)$, and $r(p)$, respectively.$G(p)$, $E(p)$, and $r(p)$ are the gray-level data obtained from the current fixation that is mapped onto point p in the map **LTG**, the edge data that are mapped onto point p in the map **LTE**, and the resolution data that are mapped onto point p in the map **LTI**, respectively.

The **LTI** map is used by the focus-of-attention mechanism to simulate the scan path. Using context-free gaze mechanisms, such as our interest operators, the system is prone to get stuck at a local point of high interest value or to oscillate indefinitely between a few points [31]. To avoid these problems, we update **LTI** to reflect the best available resolution at each point, as well as to include the time elapsed from its last update. We define f_r as the forgetting rate, r_f as the resolution in the fovea, and $r_{\max}$ ($r_{\max} < r_f$) as the highest resolution in the periphery. We then make the following modifications to the normal winner-take-all algorithm described above.

(i) The cells of **LTI** that correspond to the curren foveal region are set to the value $r_f + f_r$, instead of r_i = $r(p)$, as in the normal winner-take-all algorithm. That is $LTI(p) = r_f + f_r$ if the point p is in the current foveal region

(ii) Each time the system moves its gaze to a nev fixation point, the value of the **LTI** cell is decreased by 1 That is, $LTI(p) = LTI(p) - 1$ for all points p. If $LTI(p$ is less than 0, the value is left unchanged.

(iii) For a point p to be the candidate for the nex gaze position detected by a interest operator, the value o $LTI(p)$ should be less than or equal to $r_{\max}$.

The parameters related to this algorithm can be selected b the user. This decision will influence whether a "forgettin mechanism" (i.e., points (i) and (ii) above) or the norma algorithm is employed. For example, if we set f_r t $(r_{\max} - r_f)$, the system will select the next gaze positio independently of past information. On the contrary, if w set f_r to a very large number, the system will not select th previously investigated point as the next gaze position.

3. EXPERIMENTS

Active vision systems are currently in their infancy, an much trial and error will be required in order to evaluat emerging algorithms and methods. Thus, modularity is on of the main goals of our system. While the global desig concept bears a certain resemblance to biological system in its foveated sampling function and the asynchroneousl accessed multiple maps, the system allows for testing c various algorithms for each of its modules. At the currer stage, we have chosen to test a module that is fundamenta to any active vision system, the early vision attentiona mechanism based on the detection of interest points. I the following, we describe the evaluation norms that w have developed for comparing attentional mechanism and we describe the actual experiments and conclusion that were drawn from them.

3.1. *Experimental Methods*

After the eye-to-hand calibration described in the Ap pendix, we point the camera at the scene. The field-o view is, as previously described, 104°, the viewing distanc is 50 cm, the maximal image size is 80 cm, and the maxim pan and tilt angles are 45°. When using two-dimensiona targets, they were usually affixed to a vertical room divide We were able to observe the system performance by di playing the integrative iconic memory, **LTI**, on a grapl ics workstation.

However, although the integrative iconic memory an the scan paths can serve as subjective evaluations of th functioning of the attentional algorithm, it is also usefi to have an objective estimate of system performance

order to compare different interest operators. A reasonable performance measure is the convergence rate, that is, the distance between the actual map (reference map) and that obtained by the system as a result of the number of fixations. In [14], Yeshurun *et al.* utilized the convergence rate of a specific norm as a function of the scan path. They used scenes containing outline contours rather than gray-scale images; the original known high-resolution scene model was used as the reference to construct the norm. In our case, since we employ gray-scale scenes with a field of view of more than 100° as the target scenes, and a single view cannot cover such a wide angle of view, we require a long-term gray-level reference map (**U**). To create this reference map, we scan the whole scene using only the fovea (that is, the uniformly sampled high-resolution area) before we start the experiment. This is easily implemented with a stand-alone server process which sends top-down commends to move the camera, as described in Section 2. We also use an alternative reference interest map (**K**) which is obtained by having a person subjectively choose the salient points and focusing the fovea only at these points. Thus in this map the interest areas surrounding the points we have selected have the highest value, while other areas will have lower interest values.

Once we obtain a reference map for a scene, it is easy to define norms that compare composite long-term maps (**LTG** & **LTI**) after each scan with these reference maps. We have chosen three types of norms,

$$\text{Norm-A} \quad \frac{\sum_p \|U(p) - LTG(p)\|^2}{N(LTG)} \tag{3-1-1}$$

$$\text{Norm-B} \quad \frac{\sum_p \|U(p) - LTG(p)\|^2}{N(LTG)} \times \frac{N(U)}{N(LTG)} \tag{3-1-2}$$

$$\text{Norm-C} \quad \frac{\sum_p \|U(p) - LTG(p)\|^2 \times LTI(p)/K(p)}{N(LTG)}, \tag{3-1-3}$$

where $U(p)$, $K(p)$, $LTG(p)$, and $LTI(p)$ are the values of the maps **U**, **K**, **LTG**, and **LTI** at point p, respectively. $N(U)$ is the number of pixels in **U**. $N(LTG)$ is the number of pixels for which **LTG** possesses data. $\Sigma_p \|U(p) - LTG(p)\|^2$ is the sum of the L2 norm between the reference map U and the current long-term gray map **LTG** over the area where **LTG** possesses data. **LTI** is the interest map which stores the best local spatial resolution for **LTG** at each point. K is the resolution map in which we manually select the interest points.

Norm-A is normalized by the number of pixels in **LTG**. Although this norm is quite basic, it is normalized by the size of the areas in the map that have already been acquired by the system. Thus, the system might converge even if it has only scanned a very small area of the scene. We have used Norm-B to compensate for the actual area that the system has scanned, because Norm-B is multiplied by the ratio of the number of pixels in **U** to that of **LTG**. Obviously, evaluating the distance between any two maps is a nontrivial problem, and hence any formal norm is not very likely to capture differences that humans consider important. Thus, we employ an additional norm, Norm-C, that might more closely reflect this type of subjective judgment. The **K** map actually represents the relative importance that the human operator assigns to each region (since it is constructed manually). By normalizing the L2 difference using the ratio between $LTI(p)$ (the interest value assigned by the attentional mechanism) and $K(p)$ (the interest value assigned by a human), we attempt to emphasize the performance of the system in "important" areas and to neglect its performance in relatively "unimportant" areas. In the experiments described below, we use these norms to illustrate the performance of the focus-of-attention mechanism, as well as to determine which norm is appropriate for characterizing it.

We have employed three types of scenes in our experiments: (i) single object, (ii) multiple objects, and (iii) natural. The experiments with single-object scenes permit us to observe how the gaze moves to locate a target. Different sizes of squares and disks were used as targets. Using multiple objects permits us to determine how the system selects the scan-path among them. We have presented two natural scenes in this paper; one is the well-known drawing by Noton and Stark [1], used in their experiments with humans, and the other is a typical 3D laboratory scene. In the following subsections, we illustrate the experimental results using both L2 norms and scan-paths. The results will be discussed in Section 4.

3.2. *Single-Object Scene*

In this experiment, we attached white squares and circles of size 10, 6, and 2 cm to a black background. The camera was initially pointed at a target 40 cm away from the object. The results of this experiment were rather similar for the circles and squares, and the convergence rate was also similar for all three norms. It can be seen in Fig. 7 that the symmetry and corner operators converged in a similar manner and that the convergence rate depends on the size of the object; the smaller the object, the faster the convergence. This is also to be expected, since a smaller object can be covered within fewer fixations. The main point of this experiment was to test the stability of the two benchmark operators and to assure that the attention mechanism did not get stuck at unexpected locations.

Performing the same experiment on larger objects

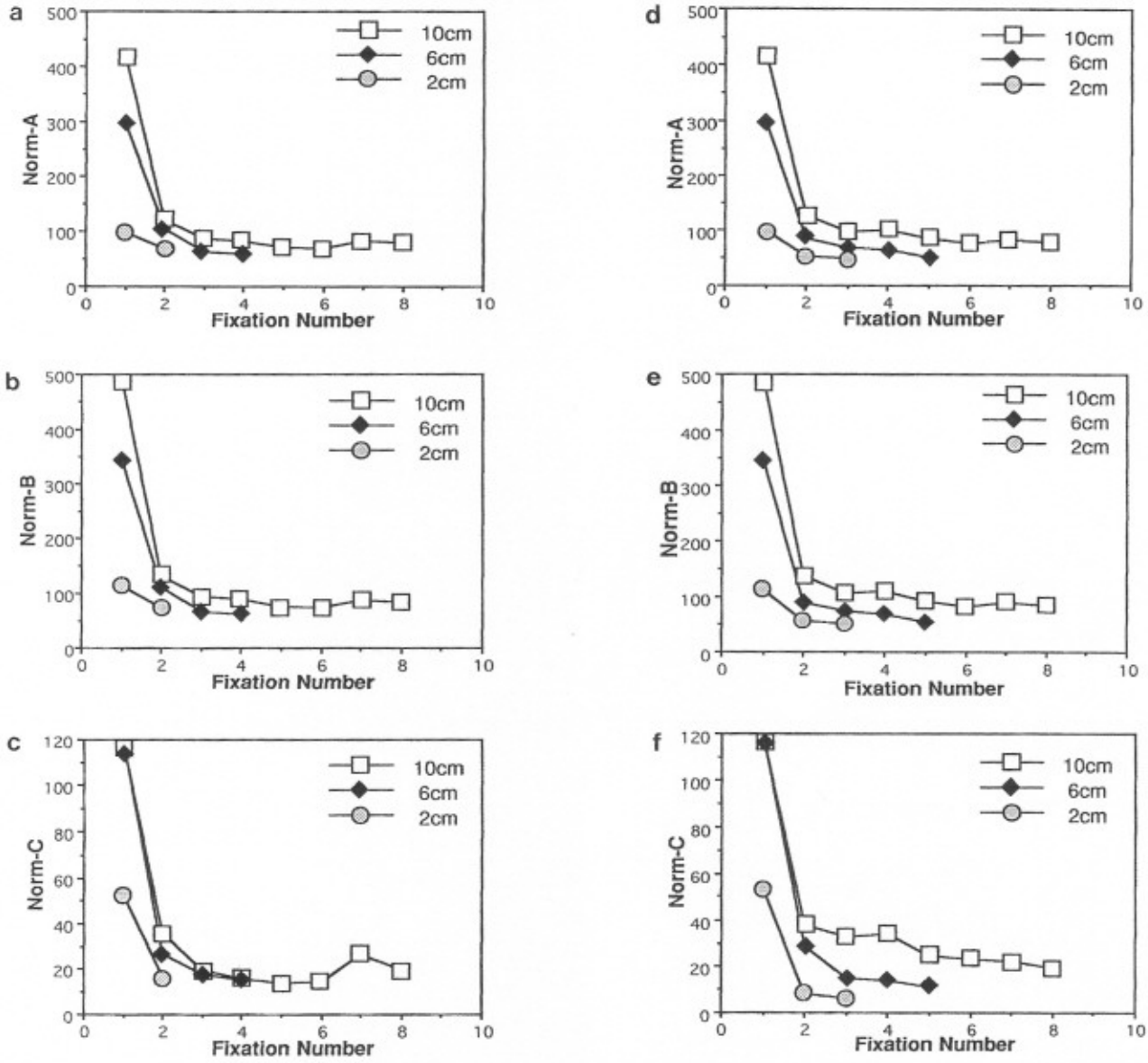

FIG. 7. Convergence graphs for a single object. (a) L2 Norm A with cornerity operator for a disk. (b) L2 Norm B with cornerity operator for a disk. (c) L2 Norm C with cornerity operator for a disk. (d) L2 Norm A with symmetry operator for a square. (e) L2 Norm B with symmetry operator for a square. (f) L2 Norm C with symmetry operator for a square.

(squares and circles of size 14 cm), we have found a difference between the behavior of the symmetry and corner operators. The symmetry operator tends to achieve better results for the first (or first two) fixations. This can be explained by the fact that the symmetry operator yields higher values in the vicinity of the center of gravity of an object. Thus the view obtained using its first fixation is usually better than that obtained by the corner operator, which will tend to fixate along the border of the object.

3.3. *Multiple-Object Scene*

In this experiment, the scene consisted of a few artificial objects. The initial position of the camera was as in the previous experiment, and the scene was scanned using symmetry and corner detection. Figure 8 shows the gaze path for multiple objects using corner detection. Analyzing the convergence graphs for such an image (Fig. 9), it is evident why it might be necessary to use different norms under different circumstances. Notice that while the convergence is nonmonotonic for Norm-B, it is much smoother and more monotonic for Norm-C. Keeping in mind that Norm-C actually takes into account those areas that humans consider important, it would seem to make sense to use this particular norm to ultimately evaluate the performance of the operators.

Three different *gaze control* strategies were tested using

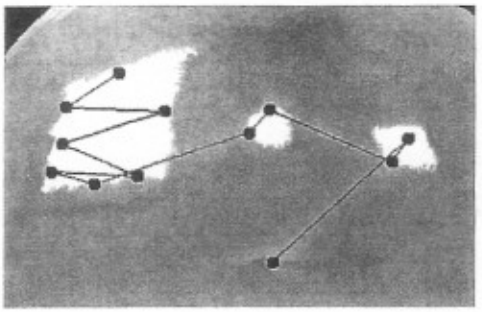

FIG. 8. Scan path for a multiple-object scene.

the strategy described in Section 2.3.2 (the scan paths and convergence graphs with symmetry detection are shown in Figs. 10 and 11, respectively): (i) Ignore any knowledge regarding the location of previous fixation points and select the locations with the highest current interest values (Fig. 10a). (ii) Use all knowledge regarding previous fixation points, thus avoiding a return to a previously selected point (Fig. 10b). (iii) Partially use previously obtained knowledge by avoiding the use of recently sampled fixation points by including a forgetting factor, as defined in Section 2.3.2 (Fig. 10c). From the experiments we have done, it is evident that the intermediate updating policy, (iii), performs better than the other two. If one were to only use the currently available data and ignore all knowledge pertaining to previous fixation points, this might lead to a situation where the system essentially becomes stuck and does not converge. However, if one saves previous fixation points, the system will not be able to find a new gaze position after several gaze movements. It is well known that human scan paths tend to repeat unless the scene changes. Therefore, updating policy (iii) permits us to simulate such a gaze movement pattern.

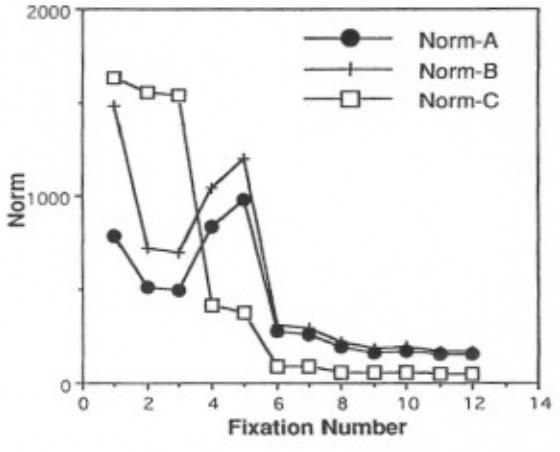

FIG. 9. Convergence graph for a multiple-object scene 1.

3.4. *Natural Scene*

The first image we used is a binary image that depicts a drawing by Paul Klee and was extensively employed for human psychophysical experiments. We have employed the symmetry operator to scan this image, and though we know that the psychophysical scan path is not very stable and might vary from person to person, the similarity between the human scan path and the symmetry operator scan path is quite striking (see Fig. 1). In another example, shown in Fig. 12, we explore a gray-level image of Einstein.

Applying the symmetry and corner attentional operators to a live natural 3D scene (our laboratory), we have found that the two methods yield similar results, with practically every updating method, and converged within 8–10 fixations. It should be noted in this regard that the natural scene we have used consisted mainly of low spatial frequencies. In order to more completely assess attentional operators on natural scenes, it would be necessary to experiment with scenes containing various spectral characteristics, since attentional operators are prone to fixate on textured areas with high spatial frequencies.

4. CONCLUSIONS AND DISCUSSION

In this paper, we have presented a testbed implementation of an active, foveated vision system. The system is highly modular and consists of processes that asynchroneously update multiple maps that represent various aspects of visual data. This design facilitates incorporation and testing of existing and emerging computer vision algorithms in a single environment.

In order to demonstrate the modularity of the system and its potential as a testbed for active vision, we have implemented three of its crucial modules, gaze control, image integration, and robot control. We have tested two attentional algorithms and carried out a benchmark study using this environment. Such a study requires a quantitative evaluation of the performance of the tested system. Thus we have developed specific measurement norms that attempt to incorporate human knowledge regarding the relative importance of interest points on the image.

The experiments we have carried out show that the system is stable and that the integrated image it acquires indeed converges. We have tested two specific attentional mechanisms and found that, in general, the strategy of choice might be a hybrid one that consists of applying the symmetry operator for the first fixations (for the early detection of targets) and then the corner operator. An interesting unresolved question is how to combine the two operators so that they at all times function in concert. This is a very important question since it is well known that there are several context-independent cues for gaze control. The convergence graphs show that while the straightforward norms of image distance (A and B) are somewhat oscilla-

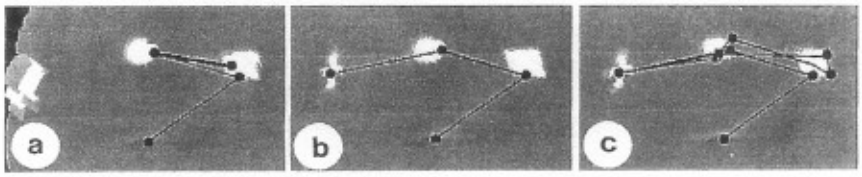

FIG. 10. Scan paths for a multiple-object scene 2. (a) No memory. (b) Infinite memory. (c) Forgetting factor 3.

tory, the human-based norm (C) was usually monotonic. We have also demonstrated the advantage of using an intermediate rate of forgetting over two other approaches, namely, a no-memory and an infinite-memory system. It remains to study how related this forgetting rate is to the contents of the scene.

In addition to the unresolved questions mentioned above, we are also studying several gaze control mechanisms and the applicability of visually guided foveated gaze control for machine vision using this system.

APPENDIX

System calibration is essential for all vision systems, especially an active vision system. Though many calibration methods have been studied for machine vision systems, they are usually rather specific to the experimental situation [45]. We require a calibration method that is suitable for our particular system. We need to know the relative position and orientation (i) between the hand (end effector) and the robot base and (ii) between the camera and the end effector. The calibration of parameters (i) is incorporated in RCCL [46], which we use to control the PUMA robot. We need to calibrate parameters (ii) to direct the camera mounted on the robot to the target direction.

A characteristic of our system is that we cannot use the exact positional information of targets on the image, except for the fovea, because the resolution changes in the periphery. However, since we are employing an active vision system, we can intentionally move the sensor to achieve calibration. This is not the case with conventional calibrating scheme. With this in mind, we have developed the following active calibration method, with which we can automatically calibrate the system before each experiment. This methodology is used to calibrate the six geometric parameters (eye-to-hand calibration) which represent the three rotational and three translational parameters (R and (x_t, y_t, z_t), respectively) that relate the end effector of the six joints of the PUMA 560 to the CID camera coordinate system. By intentionally moving the camera, the method calculates these parameters using the available estimated rotational parameters R_e and translational parameters (x_{et}, y_{et}, z_{et}).

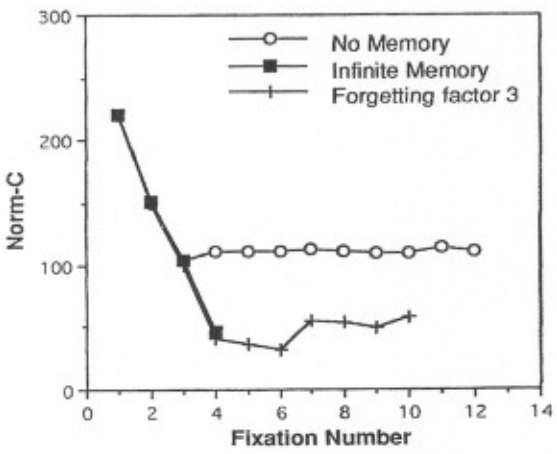

FIG. 11. Convergence graph for a multiple-object scene 2.

Calibration of Rotation Elements

In the following steps, we assume that estimates of the optical calibration and geometric parameters are already available. We can use the previously calibrated parameters or manually measured parameters as these estimates. Accordingly, we obtain

$$\begin{pmatrix} x_{ec} \\ y_{ec} \\ z_{ec} \\ 1 \end{pmatrix} = \begin{pmatrix} & R_e & & \begin{matrix} x_{et} \\ y_{et} \\ z_{et} \end{matrix} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r \\ y_r \\ z_r \\ 1 \end{pmatrix}, \qquad \text{(A-1)}$$

where $(x_{ec}\ y_{ec}\ z_{ec})$ is a three-dimensional position vector relative to the camera coordinate system ("virtual camera coordinate"). $(x_r\ y_r\ z_r)$ is a position vector of that point relative to the end effector coordinate system. $(x_{et}\ y_{et}\ z_{et})$ is an estimated transitional vector of the focal point of the camera relative to the end effector. R_e is a 3×3 matrix composed of rotational parameter estimates. Because of error, this $(x_{ec}\ y_{ec}\ z_{ec})$ is not the actual camera coordinate. We call this coordinate system the "virtual camera coordinate" (VCC). We use the notation $(x_c\ y_c\ z_c)$ for the "actual camera coordinate" (ACC) system. In Step 1.1 (below), we calculate the yaw and pitch parameters, and the roll element is calculated in Step 1.2. In these steps, R' is used to express the pitch and yaw errors in R_e; R'_e is used to

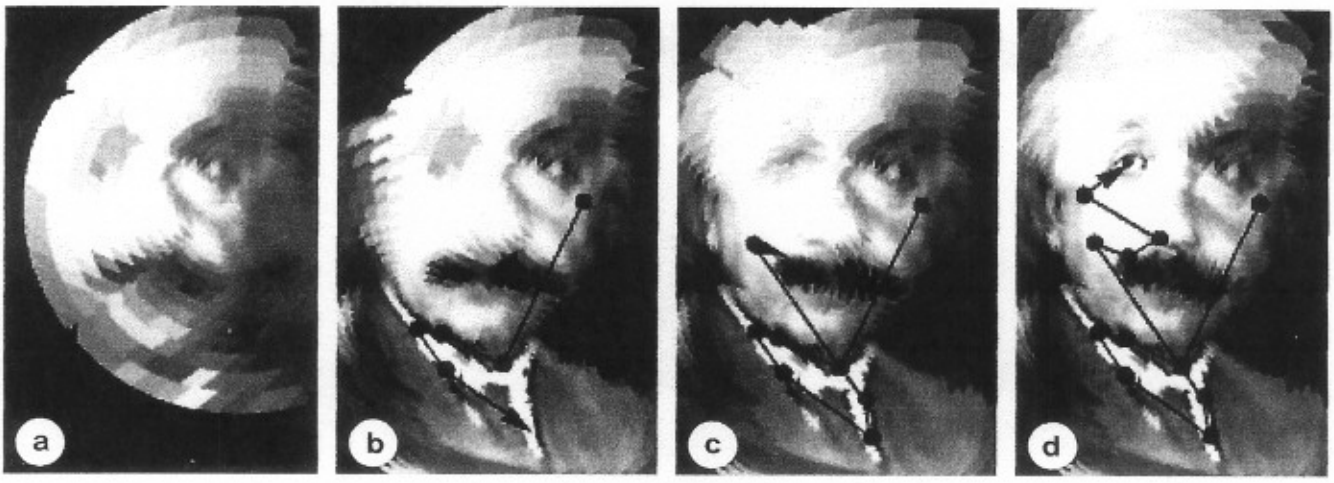

FIG. 12. Scan path for a 2D image. (a) Initial fixation. (b) Sixth fixation. (c) Tenth fixation. (d) Thirteen fixation.

represent the rotational parameters after Step 1.1. When the rotational parameters are calibrated, the translational parameters also need to be updated because they are dependent on the rotational parameters. $(x'_{et}, y'_{et}, z'_{et})$ is used for the estimated translational parameters after Step 1.1 and $(x''_{et}, y''_{et}, z''_{et})$ after Step 1.2.

Step 1.1. *Calibration of pitch and yaw angles.* First, the operator moves the camera so that a mark placed in the scene is at the center of the image, that is, along the view direction. Then the system translates the camera along the estimated view direction (Z_{ec} direction) by $d_1 > 0$ and $d_2 < 0$. If the available parameters are correct, the mark stays at the center of the image and we need not proceed further. Otherwise, the mark does move in the image. Let us define I_k $(k = 1, 2)$ as the mark position in the image and $\theta_k (k = 1, 2)$ as the angle between the direction of the mark and the direction of the image center. I_1 and I_2 are obtained by simple image processing. Since we employ a small white mark on a black background for the calibration, simple binarization will permit us to calculate its center of gravity. If the mark does move in the image, we can calculate the θ_1 and θ_2 from I_1 and I_2, respectively. From these observations, we can obtain the angle $(\Delta\theta_z)$ between the Z_c axis and the Z_{ec} axis as follows:

$$\Delta\theta_z = -\tan^{-1}\left(\frac{(d_1 - d_2)\sin(\theta_1)\sin(\theta_2)}{d_1\cos(\theta_1)\sin(\theta_2) - d_2\sin(\theta_1)\cos(\theta_2)}\right). \tag{A-2}$$

Let $\Delta\theta_x$ be the angle between the X_c axis and the direction the mark moves. From $\Delta\theta_z$ and $\Delta\theta_x$, the direction of Z_{ec} relative to ACC is

$$\left(\cos\left(\frac{\pi}{2} - \Delta\theta_z\right)\cos(\Delta\theta_x - \pi),\quad \cos\left(\frac{\pi}{2} - \Delta\theta_z\right)\sin(\Delta\theta_x - \pi),\quad \sin\left(\frac{\pi}{2} - \Delta\theta_z\right)\right).$$

That is,

$$(-\sin(\Delta\theta_z)\cos(\Delta\theta_x), -\sin(\Delta\theta_z)\sin(\Delta\theta_x),\quad \cos(\Delta\theta_z)). \tag{A-3}$$

The pitch $(\Delta\theta p)$ and yaw $(\Delta\theta y)$ to match the Z_{ec} axis to the Z_c axis are calculated by solving the equations

$$R'\begin{pmatrix} -\sin(\Delta\theta_z)\cos(\Delta\theta_x) \\ -\sin(\Delta\theta_z)\sin(\Delta\theta_x) \\ \cos(\Delta\theta_z) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \tag{A-4}$$

where

$$R' = \begin{pmatrix} \cos(\Delta\theta_p) & \sin(\Delta\theta_p)\sin(\Delta\theta_y) & \sin(\Delta\theta_p)\cos(\Delta\theta_y) \\ 0 & \cos(\Delta\theta_y) & -\sin(\Delta\theta_y) \\ -\sin(\Delta\theta_p) & \cos(\Delta\theta_p)\sin(\Delta\theta_y) & \cos(\Delta\theta_p)\cos(\Delta\theta_y) \end{pmatrix} \tag{A-5}$$

is the rotational matrix needed to compensate for the error in the available parameters, assuming the roll error is 0. Under the assumption $0 \le \Delta\theta_z < \pi/2$, we obtain

$$\Delta\theta_p = \tan^{-1}\left(\frac{\sin(\Delta\theta_z)\cos(\Delta\theta_x)}{\sqrt{\cos^2(\Delta\theta_z) + \sin^2(\Delta\theta_z)\sin^2(\Delta\theta_x)}}\right) \tag{A-6}$$

$$\Delta\theta_y = -\tan^{-1}\left(\frac{\sin(\Delta\theta_z)\sin(\Delta\theta_x)}{\cos(\Delta\theta_z)}\right). \tag{A-7}$$

Step 1.2. *Calibration of roll angle.* As a result of Step 1.1, we have estimated parameters that relate the end effector coordinate to the camera coordinate as

$$\begin{pmatrix} x'_{ec} \\ y'_{ec} \\ z'_{ec} \\ 1 \end{pmatrix} = \begin{pmatrix} & R' & & 0 \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} & R_e & & x_{et} \\ & & & y_{et} \\ & & & z_{et} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r \\ y_r \\ z_r \\ 1 \end{pmatrix} \tag{A-8}$$

$$= \begin{pmatrix} & R'_e & & x'_{et} \\ & & & y'_{et} \\ & & & z'_{et} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r \\ y_r \\ z_r \\ 1 \end{pmatrix}$$

where R'_e is the matrix defining the calibrated rotational parameters in Step 1.1.

These estimated parameters provide us with the information to rotate the end effector coordinate system so that the Z_r axis matches the Z_c axis. We still need a roll angle to match the X_r axis to the X_c axis and the Y_e axis to the Y_c axis. To calculate the roll angle, the camera needs to be moved so that we can see a mark at the center of image. Then the system translates the camera along the estimated horizontal direction in the image (X'_{ec} direction). If the current parameters are correct, the mark will move along a horizontal line and we do not need this procedure. If the mark moves in another way, we can calculate the $\Delta\theta_r$ from the direction of movement $\Delta\theta_x$ as follows:

$$\Delta\theta_r = \pi - \Delta\theta_x. \tag{A-9}$$

The final rotational parameters (R) are calculated as follows:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\Delta\theta_r) & -\sin(\Delta\theta_r) & 0 & 0 \\ \sin(\Delta\theta_r) & \cos(\Delta\theta_r) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} & R'_e & & x'_{et} \\ & & & y'_{et} \\ & & & z'_{et} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r \\ y_r \\ z_r \\ 1 \end{pmatrix} \tag{A-10}$$

$$= \begin{pmatrix} & R & & x''_{et} \\ & & & y''_{et} \\ & & & z''_{et} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r \\ y_r \\ z_r \\ 1 \end{pmatrix}.$$

The following procedure calculates the calibrated transitional vector of the camera focal point relative to the end effector (x_t, y_t, z_t) based on the translational parameters (x''_{et}, y''_{et}, z''_{et}).

Calibration of Translation Elements

Just as the rotational parameters can be determined by translating the camera, the translation parameters can be calculated by rotating it. For example, by making pan rotations of $a_1°$ and $a_2°$, and observing the target position errors ($e_1°$ and $e_2°$, respectively), we can compute the x and z elements of the translation parameters as functions of the distance between the target and the estimated pivot position. If we can obtain this distance with a rangefinder or some stereo method, we can then determine all of the translation parameters by making pan and tilt rotations of the camera. However, this approach is not robust because we cannot accurately compute e_1 and e_2, especially with a foveated sensor. Here we adopt another approach, which is more direct, as shown below.

Step 2.1. *Calibration of the x element.* First the camera is moved so that a mark placed in the scene is located at the center of the image. Then system rotates the camera around the Y_c direction by $a_1°$ and $-a_1°$. Let l_1 and l_2 be the corresponding X_c coordinates of the target made with pan rotations of a_1 and $-a_1$. If the current estimate x''_{et} is correct, the mark will move by the same distance but in the opposite direction (on the X_c axis); that is, $-l_1 = l_2$. If the mark does not move by the same amount, we update the current x element (x''_{et}) of the translation parameter and iterate this procedure until the target moves by the same distance. If the absolute value of l_1 is bigger than the absolute value of l_2, we set x''_{et} to $x''_{et} - \Delta r$. Otherwise, we set x''_{et} to $x''_{et} + \Delta r$. Δr can be taken as the maximum accuracy of PUMA movement, or some other predefined accuracy associated with the system. The updated value of x''_{et} is the calibrated x element x_t.

In this approach, if the current x element of the translation parameter is far from the correct value, it would take considerable time to complete the procedure. But since we normally use this calibration procedure only to tune the existing parameters, it usually does not take a long time.

Step 2.2. *Calibration of the y element.* We can calibrate the y element y_t of the translation vector by making tilt rotations along the X_c axis as in Step 2.1.

Step 2.3. *Calibration of the z element.* First, the camera is moved so that a mark is at the center of the image. Then the system rotates the camera around one of the X_c or Y_c directions by $a_3°$. Let l_3 be the distance of the target movement on the image. We can obtain the correct value for l_3 (L_3) from the mapping parameter. If l_3 is equal to L_3, the current parameter z''_{et} is correct. If l_3 is bigger than

L_3, we set z''_{et} to $z''_{et} - \Delta r$. Otherwise, we set z''_{et} to $z''_{et} - \Delta r$. We iterate this procedure until l_3 is equal to L_3. The final value is the calibrated z_t.

Finally we replace $(x''_{et}, y''_{et}, z''_{et})$ in (A-10) by (x_t, y_t, z_t) and obtain the calibrated eye-to-hand parameter. As indicated above, the required image processing and arithmetic operations needed for this calibration method are very simple and fast.

ACKNOWLEDGMENTS

We thank Francois Gauthier, Gilbert Soucy, Pierre Tremblay, and John Zelek for their assistance in carrying out this project. Yehezkel Yeshurun was partially supported by a grant from the US–Israel BSF. Martin D. Levine thanks the Canadian Institute for Advanced Research and PRECARN Associates for their support. This research was partially supported by the NCE IRIS program, The Natural Sciences and Engineering Research Council of Canada, the FCAR Program of the Province of Quebec, and Canon Inc.

REFERENCES

1. D. Noton and L. Stark, Scanpaths in saccadic eye movements while viewing and recognizing patterns, *Vision Res.* **11,** 1971, 929–942.
2. E. L. Schwartz, Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception. *Biol. Cybernetics* **25,** 1977, 181–194.
3. P. Burt and T. Adelson, The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.* 1983, 532.
4. M. Tistarelli and G. Sandini, Estimation of depth from motion using an anthropomorphic visual sensor, *Image Vision Comput.* **8**(4), 1990, 271–278.
5. A. S. Rojer and E. L. Schwartz, Design considerations for a space-variant visual sensor with complex-logarithmic geometry, *Proc. ICPR,* 1990, 278–285.
6. D. H. Ballard, Animated vision, Univ. of Rochester, Dept. of Computer Science, Tech. Rept. 61, TR, 1990.
7. A. Califano, R. Kjeldsen, and R. M. Bolle, Data and model driven foveation, *Proc. ICPR,* 1990, 1–8.
8. K. Pahlavan, T. Uhlin, and J. Eklundh, Integrating primary ocular processes, *Proc. ECCV,* 1992, 526–541.
9. J. L. Crowley, P. Bobet and M. Mesrabi, Gaze control for a binocular camera head, *Proc. ECCV,* 1992, 588–596.
10. D. R. Rimey and C. M. Brown, Where to look next using a Bayes net: Incorporating geometric relations, *Proc ECCV,* 1992, 543–550.
11. S. M. Culhane and J. K. Tsotsos, An attentional prototype for early vision, *Proc ECCV,* 1992, 551–560.
12. F. Attneave, Informational aspects of visual perception, *Psychol. Rev.* **61,** 1954, 183–193.
13. L. Kaufman and W. Richards, Spontaneous fixation tendencies for visual forms, *Perception Psychophys.* **5**(2), 1969, 85–88.
14. Y. Yeshurun and E. L. Schwartz, Shape description with a space-variant sensor: Algorithm for scan-path, fusion, and convergence over multiple scans. *IEEE Trans. Pattern Anal. Machine Intell.* **11**(11), 1989, 1217–1222.
15. S. Peleg, O. Federbush, and R. Hummel, Custom Made Pyramids, in *Parallel Computer Vision,* (L. Uhr, Ed.), pp. 125–146, Academic Press, New York, 1987.
16. N. Sorek and Y. Y. Zeevi, Online visual data compression along a one dimensional scan, *SPIE Vol. 1001 Visual Communication and Image Processing,* 1988, 764–770.
17. D. Reisfeld, H. Wolfson, and Y. Yeshurun, Detection of interest points using symmetry, *Proc. ICCV,* 1990, 62–65.
18. J. Jonides, D. E. Irwin, and S. Yantis, Integrating visual information from successive fixations, *Science* **215**(8), 1982, 192–194.
19. D. E. Irwin, S. Yantis, and J. Jonides, Evidence against visual integration across saccadic eye movements, *Perception Psychophys.* **34**(1), 1983, 49–57.
20. J. R. Duhamel, C. L. Colby, and M. E. Goldberg, The updating of the representation of visual space in parietal cortex by intended eye movements, *Science* **255**(3), 1992, 90–93.
21. M. D. Levine, *Vision in Man and Machine,* McGraw–Hill, New York, 1985.
22. J. J. Koenderink and A. J. VanDoorn, Visual detection of spatial contrast: Influence of location in the visual field, target extent, and illuminance level, *Biol. Cybernetics* **30,** 1978, 157–167.
23. D. Casasent and D. Psaltis, Position, rotation, and scale invariant optical correlation, *Appl. Optics* **15**(7), 1976, 1795–1799.
24. S. W. Wilson, On the retino-cortial mapping, *Int. J. Man-Machine Stud.* **18,** 1983, 361–389.
25. R. Kronauer and Y. Zeevi, Reorganization and diversification of signals in vision, *IEEE Trans. Syst. Man Cybernetics* **15**(1), 1985, 85–88.
26. J. Spiegel, F. Kreider, *et al.* A foveated retina-like sensor using CCD technology, in *Analog VLSI Implementations of Neural Networks,* (C. Mead and M. Ismail, Eds.), Kluwer, Boston, 1989.
27. C. F. R. Weiman, Video compression via log polar mapping, *SPIE Symposium on OE/Aerospace Sensing, 1990.*
28. A. Treisman. Preattentive processing in vision, *Comput. Vision Graphics Image Process.* **31,** 1985, 156–177.
29. S. A. Hillyard and T. F. Münte, Selective attention to color and location: An analysis with event-related brain potentials, *Perception Psychophys.* **36**(2), 1984, 185–198.
30. C. L. Colby, The neuroanatomy and neurophysiology of attention, *Child Neurol.* **6,** 1991, S90–S118.
31. C. Koch and S. Ullman, Shifts in selective visual attention: Towards the underlying neural circuitry, *Human Neurol.* **4,** 1985, 219–227.
32. P. A. Sandon, Simulating visual attention, *Cognit. Neuroscience* **2**(3), 1992, 213–231.
33. J. R. Antes and J. Penland, Picture context effects on eye movement patterns, in *Eye Movements: Cognition and Visual Perception.* (R. Monty, J. Senders, and D. Fisher, Eds.), pp. 157–170, Laurence Erlbaum Associates, Hillsdale, NJ, 1981.
34. N. H. Mackworth and J. S. Brunner, How adults and children search and recognize pictures, *Human Develop.* **13,** 1970, 149–177.
35. M. M. Haith, T. Bergman, and M. J. Moore, Eye contact and face scanning in early infancy, *Science,* **190,** 1977, 853–855.
36. H. P. Moravec, Towards automatic visual obstacle avoidance, *Proc. IJCAI,* 1977, 584.
37. S. W. Zucker, Early orientation selection: Tangent fields and the dimensionality of their support, *Comput. Vision Graphics Image Process.* **32,** 1985, 74–103.
38. S. W. Zucker, A. Dobbins, and L. Iverson, Two stages of curve detection suggest two styles of visual computation, *Neural Computation* (1), 1989, 68–81.
39. D. Reisfeld and Y. Yeshurun, Robust detection of facial features by generalized symmetry, *Proc. ICPR,* 1992, 117–121.
40. J. K. Tsotsos, On the relative complexity of active vs. passive visual search, *Int. J. Comput. Vision* **7**(2), 1992, 127–141.

41. A. Pugh, *Polyhedra—A Visual Approach.* Univ. of California Press, Los Angeles, 1976.
42. B. K. P. Horn, Extended Gaussian images, *Proc. IEEE* **72**(12), 1984, 1671–1686.
43. G. Fekete and L. S. Davis, Property spheres: A new representation for 3-D object recognition, *Proc. IEEE Workshop on Computer Vision,* 1984, 192–201.
44. C. H. Chen and A. C. Kak, A robot vision system for recognizing 3-D objects in low-order polynomial time, *IEEE Trans. Systems Man Cybernetics* **19**(6), 1989, 1535–1563.
45. R. Y. Tsai and R. K. Lenz, A new technique for fully autonomous and efficient 3D robotics hand/eye calibration, *IEEE Trans. Robotics Automation* **5**(3), June 1989, 345–358.
46. J. Lloyd, M. Parker, and R. McClain, Extending the RCCL programming environment to multiple robots and processors, *Proc. Int. Conf. on Robotics and Automation,* 1988, 465–469.