An Active Vision Approach for Locating Salient Features of Objects Using Log-Polar Mapping with no Camera Motion

Magued Bishay, Atsushi Kara, D. M. Wilkes, R. A. Peters II, and Kazuhiko Kawamura
Center for Intelligent Systems, Vanderbilt University, Nashville, TN, P.O.Box 1804, Sta. B
e-mail: magued@vuse.vanderbilt.edu

Abstract

For object recognition humans use explorative saccades to locate salient features of objects and then recognize them. The objective of this paper was to demonstrate the conceptual design of an active vision system that locates salient features of objects through a number of explorative saccades using of the log-polar mapping. The gaze at each feature provided information about the *identity* of the feature. The visual field was divided into a center (the fovea) and a surround (the periphery). The fovea is where the gaze was, and the periphery was the source of *new gaze targets* ("Where to look next"). Only the periphery was being processed in our current implementation. *Corners* were chosen to be the gaze targets as well as the peripheral visual features that were fixated in subsequent saccades. Through a series of explorative saccades our algorithm located all the visible corners of the object. The system was implemented with the absence of a moving camera yet active vision concepts have been used, namely: The use of space-variant sensing (by emulating the sensor), gazing at prominent features, finding new gaze targets, and performing saccades in order to fixate the new gaze targets.

1 Introduction

Active vision can dramatically simplify computations of early vision, enabling areas of interest to be examined at desired resolution without the cost of uniform high resolution sensing. Active vision often incorporates space-variant sensing [11]. A space-variant sensor is a camera image plane whose resolution varies as a function of position. The human retina is a prime example of such space-variance. The retina has a high resolution center, called the *fovea*, surrounded by a periphery whose resolution decreases as a function of radial distance from the fovea. Therefore the visual field could be divided into two regions: a center and a surround; The center is where the gaze currently is, and the surround is the source of *new gaze targets* ("Where to look next"). Similar to human retina, the features at the periphery of a space variant sensor are represented at reduced resolution, yet they must have sufficient fidelity to attract *attention*. An important question, hence, is: what are visual features which can be informative at low resolution? Once these visual features are located they are considered as new gaze targets. Explorative saccades are performed in order to fixate them. The generation of explorative saccades is the most elementary behavior in the context of eye-movement, and is used as a basis for an active vision scene analysis by means of "cognitive" saccades.

Although several researchers have addressed the problem of attention control in the context of active vision [11], animate vision [2], or purposive vision [1], there is little information on practical approaches to finding salient features in the visual periphery and directing the gaze to these features. The objective of this paper is to demonstrate how the previously highlighted concepts of active vision could be applied to locate salient features of an object through a series of explorative saccades. The salient features that we chose at this stage of development were the corners of objects. Besides being the gaze targets, corners were the peripheral visual features which are fixated in subsequent saccades, because corners can still be reliably detected at low resolution (in the periphery). The gaze at a corner enabled us to *identify* the corner. We defined the *identity* of a corner as the *number* of edges meeting at the corner and their *orientation* relative to an arbitrarily fixed vector in the image plane (we chose this vector to be along the x-axis in the image). Knowing the identity of each detected corner would help in solving the problem of matching of

corners in stereo images [8]. Through a series of explorative saccades our algorithm locates all the visible corners of the object as we shall explain in details in the subsequent sections.

Detection of corners has been shown to be extremely useful in many computer vision tasks. Jain [5] tracked the corners of objects, in the log-polar domain, over a sequence of images in order to estimate the depth of objects using a monocular camera. In autonomous vehicle navigation Von Seelen *et al* [12] used corners as prominent features to which cognitive saccades should be applied to accomplish scene analysis. Another merit of using corners is that the geometrical shape of a corner provides a strong constraint for 3-D interpretation. For example, it is possible to recover the 3-D pose of a rectangular corner from its orthographic or perspective projection [6, 7]. Also time-to-impact computation [9] for corners would provide information the 3-D structure of the object if the camera motion was known.

This paper is organized as follows: Section 2 describes the log-polar mapping. Section 3 explains our system functionality and the interaction between its different modules. Section 4 describes our algorithm along the concepts presented in Sections 2 and 3. In Section 5 we demonstrate the performance of our system on real scenes, and finally we summarize our ideas and future research in Sections 7 and ??.

2 Log-Polar Mapping

Schwartz [10] proposed a computational interpretation of the spatial mapping between the retina and the striate cortex of various vertebrates. According to him, the retino-striate mapping is characterized by a complex-logarithmic function. The complex-logarithmic function is defined as a mapping of a complex variable z to another complex variable $u = \log z$. Let $z = (\alpha + r)\exp(j\phi)$ ($\alpha > 1$ is a real number) then $u = \log z$ becomes $u = \rho + j\theta$, where

$$\begin{aligned} \rho &= \log|z| = \log(\alpha + r) \\ \theta &= \arg(z) = \phi. \end{aligned} \tag{1}$$

This representation has the following attractive features: First, the mapping is conformal, therefore connectivity between the corners of objects are preserved in the log-polar (LP) domain. The second property is that linear scaling and rotations about the mapping origin (the fovea or the fixation point) are transformed into linear shifts along the $\log(r)$ and θ axis, respectively. The complex-logarithmic mapping is, however, *not invariant* to translation. i.e., a translation of t(= a real number) maps u into:

$$u \mapsto u = \log(z + t) \tag{2}$$

which is quite a different image from the original.

2.1 The Implementation of the Mapping

The mapping equations defined in the previous section can be used for implementing the mapping as follows: An image $I(x, y)$ is mapped about the origin (F_x, F_y) to $I^*(\rho, \theta)$ as follows:

$$I^*(\rho, \theta) = I(x, y) \tag{3}$$

where

$$\rho = M \log(1 + \sqrt{(x - F_x)^2 + (y - F_y)^2}) \tag{4}$$

and

$$\theta = \arctan \frac{y - F_y}{x - F_x}. \tag{5}$$

M is a scaling constant.

The problem with this method was that it results a sparse mapping as teh radial distance to the fovea decreases [5, 3]. This motivated us to work inversely from the mapped space and setting each pixel map in this range to the corresponding image intensity. In this case the sampled ρ, θ pixel in the mapped image $I(\rho, \theta)$and the corresponding x, y pixel in the original image $I(x, y)$ are related by

$$x = (\exp^{\rho/M} - 1)\cos\theta \tag{6}$$

and

$$y = (\exp^{\rho/M} - 1)\sin\theta. \tag{7}$$

The ranges for ρ and θ are $0 \leq \rho \leq \rho_{max}$ and $0 \leq \theta \leq 2\pi$. ρ_{max} defines the width of the mapped image.

3 Interaction scheme between system modules

The interaction scheme in our system is depicted in Figure 1. The system fixated a certain corner by locating the fovea at that corner. The retinal image (cartesian image) was then transformed to the cortical image (LP image) which was then processed for finding the corners in the periphery (as will be explained in Section 4.2). The location of these corners were kept in the *interest map* since they were the "interesting points" to be gazed at next. The

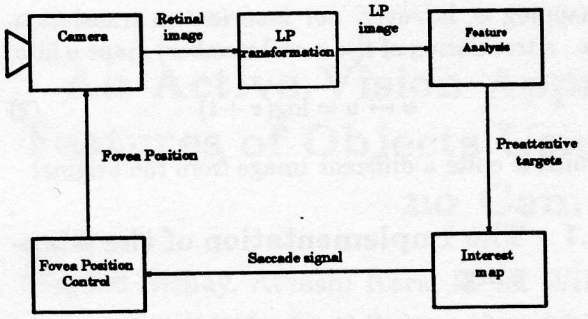

Figure 1: Interaction scheme between system modules

fovea control module then moved the fovea to these "interesting points". This means that the decision of "where to look next" in our system was feature driven [11]. Gazing at a corner resulted in knowing the "identity" of the corner: the number of edges meeting at the corner and their orientations. This facilitates establishing correspondences between detected corners in two stereo images. It should be noted that the periphery solely has been processed for the detection of the "next gaze targets" which implies that the fovea was not important for directing the gaze.

In our current system there was no camera motion. The fovea, however, could be placed at any point in the image plane. In our future work we are planning to fix the fovea in the center of the image plane, and develop camera motion strategy to fixate the each corner.

4 The Algorithm

The algorithm was based on three main tasks:

1. Locating the first corner to be fixated.
2. Gazing (foveating) at the corner in order to identify it. As was mentioned before, we defined the "identity" of the corner as the number of edges meeting at the corner and their orientations.
3. Processing the periphery of the log-polar image for locating features (in our case the corners) to which the *attention* would be directed.

The algorithm is explained in detail in the following subsections.

4.1 Locating the First Corner of the Object

The hough transform [4] was used to detect the longest edge in the image. We assumed that the

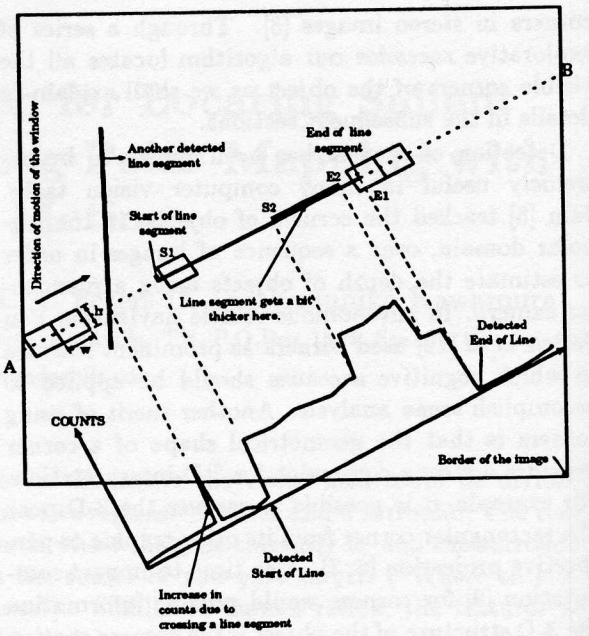

Figure 2: Algorithm for Detecting the End Points of Image Line Segment

edge detected by the hough transform belonged to the polyhedral object. This assumption is valid provided that the camera is close enough to the object. After that the start and end corners of this edge were detected, using a moving-window algorithm that is explained, using Figure 2, as follows:

The solid line represents and edge in the image. The dotted line represents the hough transform output. A window of size $h \times w = 5 \times 12$ *pixels* was moved along the dotted line. The window started at A and was moved till it reaches B where A and B were the intersections of the dotted line with the border of the image. At each position the number of white pixels inside the window were counted. The number of counts in the current window position was compared with the counts in the previous window position. The window was advanced by a distance w each time. The *first* increase of count of white pixels by 10 or more counts corresponded to the detection of a start of the line segment. Accordingly the algorithm gives point S_1 to be the start of the detected line segment, despite the fact that another increase in count happened at S_2. On the other hand the *last* decrease in the count by 10 or more corresponded to the end of the line segment. Thus in Figure 2 point E_1, not point E_2, was reported by the algorithm to be the end of the line segment. As shown in Figure 2 the spike appearing in the counts happened because the window crossed a different line segment in the image while

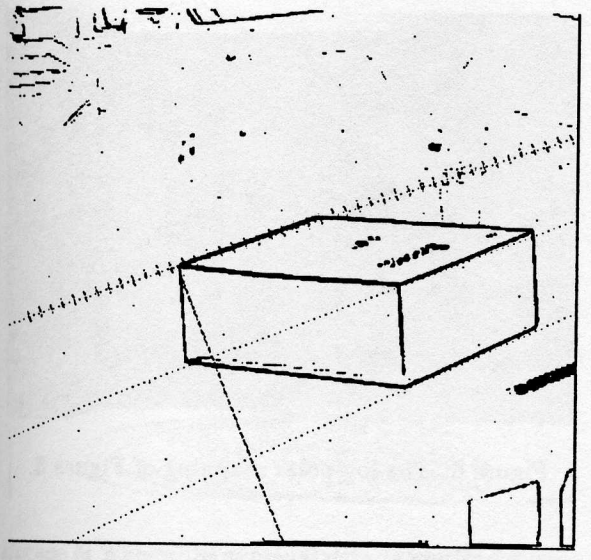

Figure 3: Finding the first corner. The line coming out from the middle of the bottom edge of the image points to the first corner to be fixated (gazed at).

it was moving on the dotted line. The spike was an increase in counts by more than 10 pixels followed by a decrease in counts by more than 10 pixels. In order to avoid that such a spike be reported as a start and end of a line segment the detected start and end of line segment should be at least $3w$ apart (w is the width of the moving window).

Figure 3 shows the performance of the hough transform and of the moving-window algorithm. We have programmed the hough transform to locate 3 edges in the image. However the algorithm for detecting the start and end corners of the edge was applied only to the edge that corresponded to the highest peak in the hough parameter space. In Figure 3 one side only of the moving window is drawn. It is depicted as small line strip drawn perpendicular to the detected edge. The length of the strip is h and the distance between two strips is w. We also plotted a line coming from the center of the lower border of the image to the point which was detected to be the start of edge. It is seen that it points to one of the object's corners. This corner was taken as the first gaze target.

4.2 Gazing at a Corner

As we mentioned before, the goal of gazing at a corner is to "identify" it, which is determining the number of edges emanating from this corner and their orientations. *Gazing at*, *foveating on*, or *fixating* a corner in our system means performing the log-polar mapping on the image with the origin of

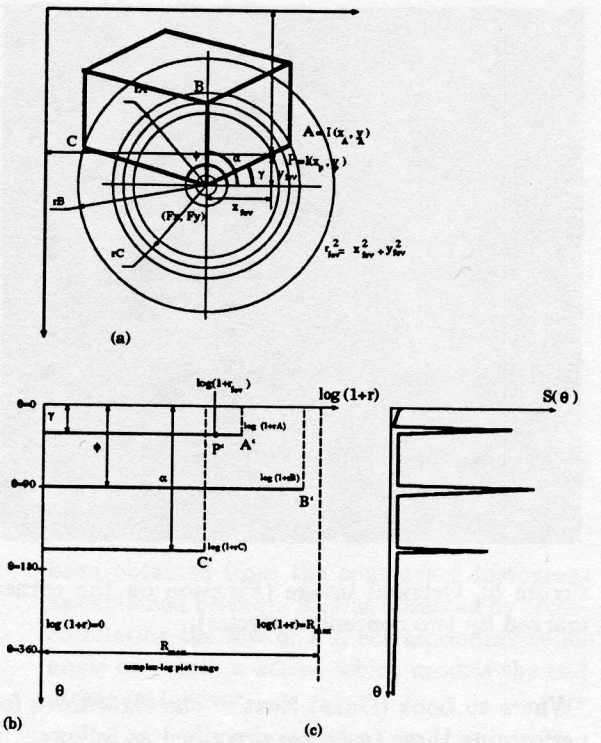

Figure 4: Algorithm of foveation at a Corner. (a) Image Plane with The fovea at a Corner. (b) The Log-Polar Mapping of Image plane in (a). (c) The Constant θ Histogram.

the mapping located at the corner position. Figure 4 illustrates the foveation on a corner. The foveation position (the origin of the log-polar mapping) is (F_x, F_y). Point A lies on an edge that passes through the foveated corner. That edge makes an angle γ with the horizontal. Point $A = (x_A, y_A)$ maps to $A\prime = (M \log(1 + r_A), \gamma)$, where $r_A = \sqrt{(x_A - F_x)^2 + (-y_A + F_y)^2}$. Similarly, points B and C are mapped to points $B\prime = (\log(1 + r_B), \phi)$ and $C\prime = (\log(1 + r_C), \alpha)$, respectively. We want to draw the attention to the fact that lines passing through the foveation origin (F_x, F_y) were mapped to *horizontal* lines in the log-polar domain. The vertical displacement of each horizontal edge in the LP domain (γ, ϕ and α) equals the angle that the corresponding edge, in the retinal (image) plane, made with horizontal direction ($\theta = 0$).

4.3 Preattentive Feature Analysis

After gazing at a corner to identify it the next gaze targets were to be located. This comprises the following two tasks: First, processing the LP image resulting from the current gaze. Second, deciding

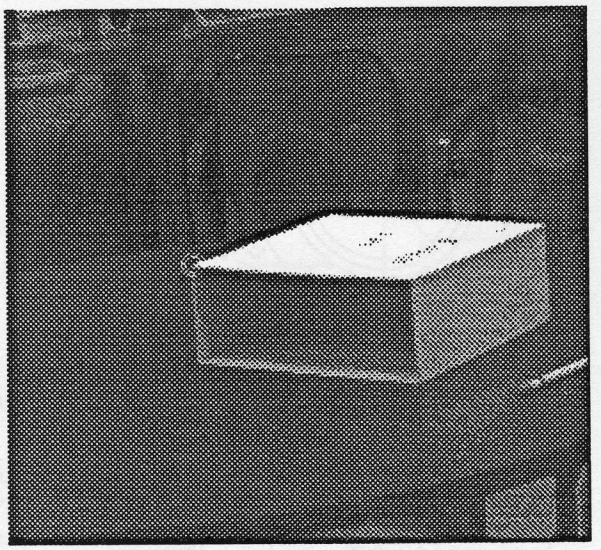

Figure 5: Original image (Fixation on the corner marked by two concentric circles).

"Where to Look (Gaze) Next". The algorithms for performing these tasks are described as follows:

1. **Processing the LP image resulting from gazing at a corner:**

   Figure 5 shows the original image of an object. The fixation is at the corner marked by two concentric circles, which was detected to be the first corner for fixation (in Section 4.1). Figure 6 shows the LP mapping of the image shown in Figure 5. The three horizontal edges corresponding to the edges meeting at the fixated corner are detected as follows:

   Assume that the original image is

   $$I(x,y) \qquad 0 \leq x < 512, 0 \leq y < 480, \qquad (8)$$

   is mapped to

   $$I(\rho,\theta) \qquad 0 \leq \rho < \rho_{max}, 0 \leq \theta < 2\pi. \quad (9)$$

   Applying the Sobel horizontal edge detector on $I(\rho,\theta)$ we obtained $SH(\rho,\theta)$, then applying the Sobel vertical edge detector we got $SV(\rho,\theta)$. Subtracting SV from SH we obtained $HL(\rho,\theta)$.

   $HL(\rho,\theta)$ contained only the horizontal edges as shown in Figure 7. The positions (α, ϕ and γ in Figure 4) of these edges were detected from the *constant θ histogram* $s(\theta)$ which was computed as follows:

   $$s(\theta) = \Sigma_{\rho=m}^{\rho_{max}} HL(\rho,\theta). \qquad (10)$$

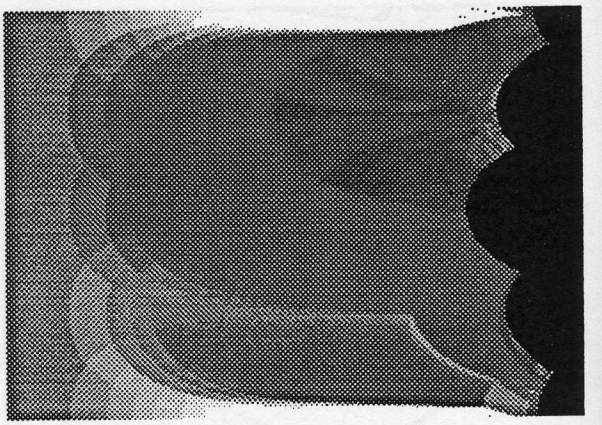

Figure 6: The log-polar mapping of Figure 5.

A horizontal edge is assumed to exist at an angle $\theta = \theta_1$ if

$$s(\theta_1) > T, \qquad (11)$$

where T is a threshold for discriminating between an $s(\theta)$ corresponding to a horizontal edge and an $s(\theta)$ that does not. Figure 8 shows $s(\theta)$ for the currently fixated corner. The three peaks correspond to the three edges meeting at the that corner. $s(\theta)$ is very low for all other θ's that did not correspond to an edge. A value of T equal to 40 was empirically chosen in our experiments. m in Equation 10 was chosen large enough such that the foveal region was discarded in computing $s(\theta)$ because the horizontal edge diverged from the horizontal direction as ρ decreased. (m = 350 was used). The algorithm was insensitive to the choice of m as long as it was large enough to discard the foveal region. On the other hand m should not be too large otherwise the $s(\theta)$ for some or all of the horizontal edges might not be large enough to be greater than T. This would result in failure to detect these edges.

2. **Deciding "Where to Look (Gaze) Next":** The decision of "where to look next" in our system was solely feature driven; the "next" corners that were to be fixated were the corners located at the end of each edge passing through the currently fixated corner. Such an approach had the advantage of obtaining the connectivity between the objects' corners. These "next corners" would be the corners A, B, and C in Figure 4 (a). Each of those corners mapped, in the LP domain, to a corner located at the end of the corresponding horizontal edge. These are points A', B', and C' in Figure 4 (b). Point A' has the coordinates (R_A, γ) (γ has

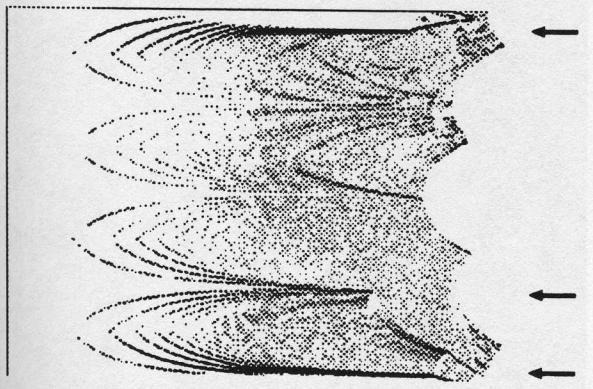

Figure 7: The result of taking the difference between the image resulting from applying the sobel horizontal edge detector and the image resulting from applying the sobel vertical edge detector. The horizontal edges only appear in this image.

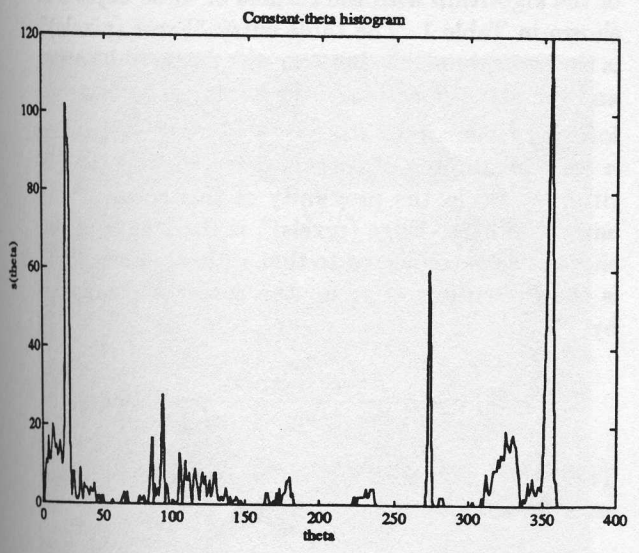

Figure 8: Peaks at the angle values that the edges make with the horizontal in the image.

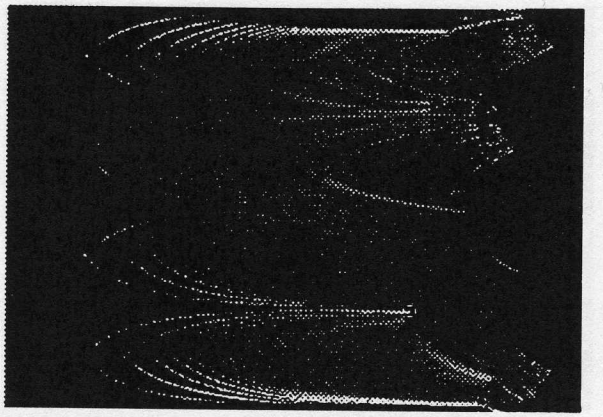

Figure 9: Detecting the "next" foveation points in the log-polar domain (small circles).

been obtained from the constant θ histogram as explained before). R_A was obtained by cross-correlating the row of HL corresponding to an angle of γ with a kernel which models the end of line as follows:

$$R_A = max(\Sigma_{u=0}^{M-1} HL(\rho, \gamma).kernel(\rho - u)), \tag{12}$$

where

$$kernel(x) = \begin{cases} 1 & -\frac{kernelsize}{2} \leq x < 0 \\ -1 & 0 \leq x < \frac{kernelsize}{2} \end{cases} \tag{13}$$

and

$$M = signalsize + kernelsize - 1. \tag{14}$$

where signalsize $= \rho_{max} - m$ and kernelsize $=$ 100.

Similarly R_B and R_C were obtained by replacing γ in the previous equation with ϕ and α respectively. The detected positions are shown by the three circles in Figure 9. These positions were mapped back to the image plane. Figure 10 shows back-mapping of these positions to the image coordinates. These positions were the position to be fixated in the future gazes.

This procedure was repeated at every detected corner in order to detect all the visible corners that belonged to the polyhedron and the connectivity between them. As the process of foveation on corners proceeds, the algorithm detects corners that have been previously identified. To avoid that, two approaches to terminate the foveation procedure and report the results have been adopted:

1. The first approach was to discard any of the new corners which was found to be in the proximity of those already kept in the "interest

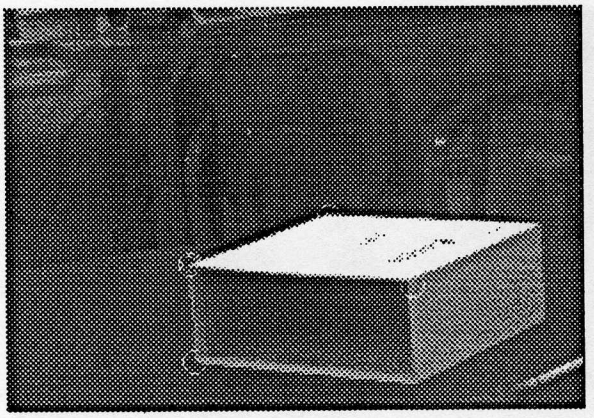

Figure 10: Next foveation positions shown as white circles.

map". The proximity distance was empirically chosen to be 25 pixels. The detection process was terminated when all the corners in the "interest map" have been foveated.

2. The second approach was to gaze at feature points irrespective of their proximity to previously detected features. In order to avoid gazing at features points indefinitely we need to know some information about the maximum number of visible corners. that number could be known if models of the objects were kept in a database. Currently we have fixed the total number of gazes to 13.

5 Experimentation with Real Images

The two approaches for keeping new corners have been tried during experimentation on different objects. Three of them are shown here. Figures 11 and 12 show the experimentation result of applying the first approach to a pair of stereo images. The performance is shown in Table 1 in the entries Box (L) and Box (R) respectively. In Figure 11 it could be seen that a corner (of coordinates (220, 345)) was detected on the lower edge of the box instead of the corner. The reason being that this corner has been detected when the gaze was at corner number 7. We observe that the shadow of the top plane of the box on the bottom plane gave rise to another edge besides the edge of the box itself. These two edges joined together in the low resolution periphery of the log-polar image at a location closer to the fovea than the actual corner location. There is another erroneously detected corner near the left

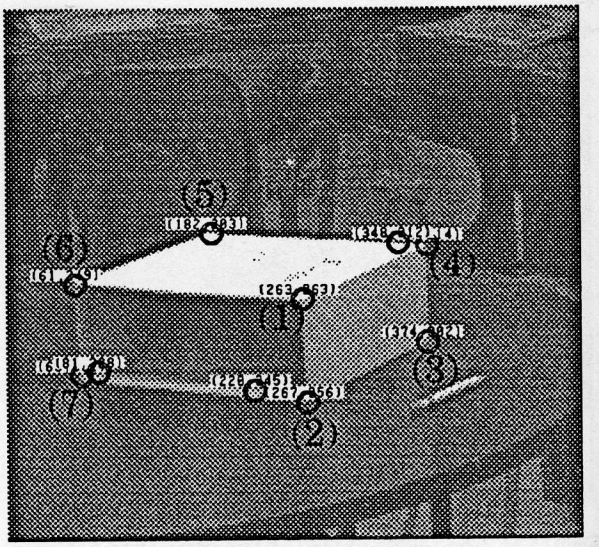

Figure 11: The left image of the hollow box

corner of this edge (corner 7) when gaze was applied to corner number 2 due to the same reason. Figures 13 and 14 show the result of applying the second approach to the triangular and trapezoidal polyhedrons. Looking at Figure 14 we observe that a corner was detected in the middle of an edge. The reason for that is that this was really a corner that was created by the change in the direction of the edge before and after this corner. The performance of the algorithm with the corners of these objects is shown in Table 1. The table entry "Error (pixels)" is the error, in pixels, between the detected location and the actual location of the corner. The detected location for a corner was $(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n})$, where n was the number of corners detected by our algorithm to be in the proximity of this corner. The entry "Shortest Edge (pixels)" is the length of the shortest edge connected to that corner. "Error (%)" is the percentage error for the corner and is given by:

$$Error(\%) = \frac{Error(pixels)}{ShortestEdge(pixels)} \times 100. \quad (15)$$

The "Average Error (%)" is the average percentage error for the object.

6 Advantages and Limitations

The proposed system has the following advantages: First, the algorithm is computationally efficient since the corners were detected through a series of

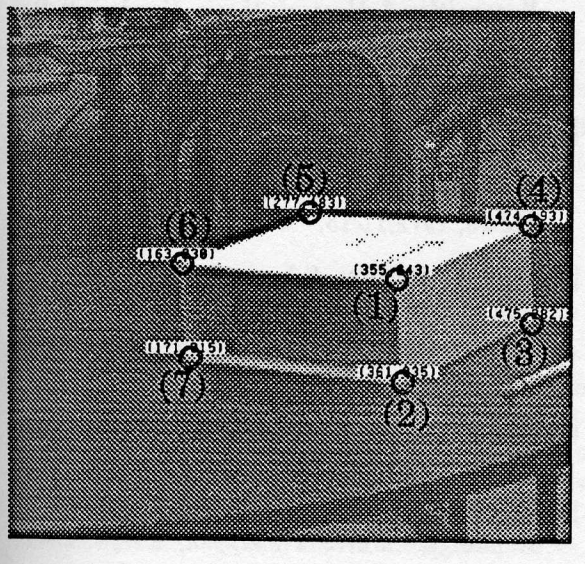

Figure 12: The right image of the hollow box

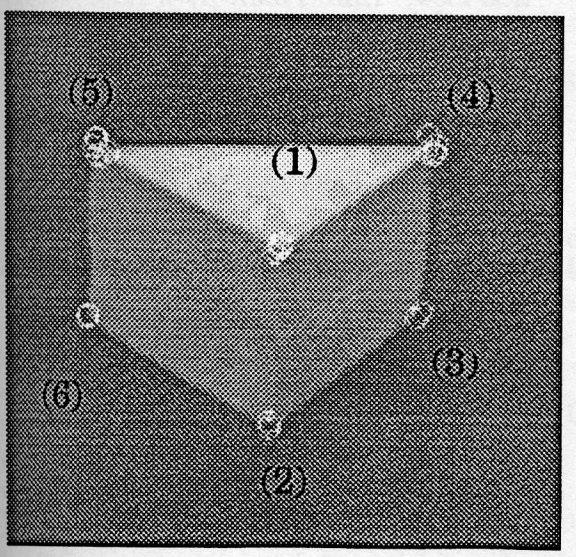

Figure 13: The Triangular polyhedron

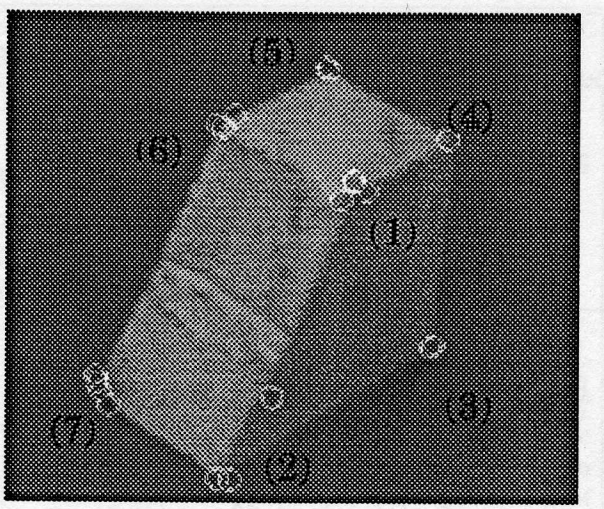

Figure 14: The Trapezoidal polyhedron

saccades without wandering in the image. Second, the algorithm is highly suited for parallel implementation, by distributing each corner in the "interest map" to a processor so that the fixations could be carried out simultaneously. This has already been implemented and will be presented in a future paper. Third, the system maintains the connectivity between the detected corners. Finally, the system simplifies the problem of matching between corners detected in two stereo images because it keeps the identity of each detected corner. The identity was defined as the number of edges meeting at the corner and their orientations. Corners of "similar" identity are matched together. "Similar identity" means that the corresponding corners have the same number of edges and the corresponding edges emanating from corresponding corners have approximately the same orientation.

On the other hand the presented system has the following limitations: First, the first corner of the object was detected in the retinal (image) plane. Second, the hough transform was used to detect the longest edge in the image and we assumed that that edge belonged to the object. Third, the foveal region occupied a big area of the LP domain and no processing was done on the foveal region.

7 Conclusions

We have presented here an algorithm for locating the salient features of objects through a series of explorative saccades and fixations on each of these features, one at a time. We have used the hough transform to locate the first salient feature. Then

Object	Corner	Error (pixel)	Shortest Edge (pixel)	Error (%)	Average Error (%)
Box (R)	1	2.0	94	2.12	2.63
	2	0.0	94	0.00	
	3	2.8	84	3.33	
	4	4.2	84	5.04	
	5	6.2	128	4.86	
	6	0.0	86.14	0.00	
	7	3.2	86.14	3.66	
Box (L)	1	2.0	91	2.19	3.97
	2	2.2	91	2.46	
	3	1.4	91	1.55	
	4	9.8	91	10.7	
	5	2.0	124	1.16	
	6	1.0	86.00	1.16	
	7	9.4	86	10.9	
Tri	1	5.8	150	3.68	5.12
	2	5.0	150	3.33	
	3	6.0	150	4.00	
	4	3.6	150	2.39	
	5	8.0	150	5.37	
	6	2.2	150	1.48	
Trapezoid	1	10.8	105.19	10	5.768
	2	7.2	127.00	5.6	
	3	7.3	221.8	3.2	
	4	1.4	105.00	1.34	
	5	3.16	105.00	3	
	6	6.4	105	6.1	
	7	4.1	127	3.25	

Table 1: The performance of the algorithm on the examples of polyhedral objects

through the gaze at this feature and through the processing the periphery the algorithm has located the corners that are going to be gazed at next. Therefore, in our system, "where to look next" is solely feature driven.

The system was implemented with the absence of a moving camera yet active vision concepts have been used, namely: The use of space-variant sensing (the emulation of it), gazing at prominent features, finding new gaze targets in the visual periphery, and performing saccades in order to fixate the new gaze targets. The algorithm has been tested on real images and the results were shown.

References

[1] Aloimonos, Y. "Purposive and Qualitative Active Vision," *Proc. of Image Understanding Workshop,* pp. 816 - 828, 1990.

[2] Ballard, D.H. "Animate Vision," *Artificial Intelligence,* Vol. 48, pp. 57 - 86, 1991.

[3] Bishay, M., "Controlled Foveation for Object Recognition," *Master's thesis,* Vanderbilt University, Department of Electrical Engineering, Nashville, TN, May, 1992.

[4] Duda, R.O. and P. E. Hart, "The Use of hough Transformation to Detect Lines and Curves in Pictures," Communications of the ACM, Vol.15, 1972,11-15.

[5] Jain, R., Bartlett, S.L., and O'Brien, N. "Motion Stereo Using Ego-Motion Complex Logarithmic Mapping," *IEEE PAMI,* Vol. 9, May, 1987.

[6] Kanatani, K., *Group-Theoretical Methods in Image Understanding,* Springer-Verlag, 1990.

[7] Kanatani, K., "Gazou Rikai (Image Understanding)," *Morikita Shuppan,* Tokyo, Japan, 1990.

[8] Marapane, S. B. and M. M. Trivedi, "Region-Based Stereo Analysis for Robotic Applications," *IEEE Trans. Systems. Man, and Cybernetics,* Vol. 19, No. 6, Nov/Dec. 1989.

[9] Massino, T. and Sandini, G. "On the Advantages of the Polar and Log-Polar Mapping for Direct Estimation of Time-to-impact from Optical Flow," *IEEE PAMI, April 1993, Vol 15, No. 5.*

[10] Schwartz, E. L. "Computational anatomy and functional architectural of striate cortex: A spatial mapping approach to perceptual coding," *Vision Research,* Vol. 20, pp.645-669, 1980.

[11] Swain, M. J. and M. Stricker, "Promising Directions in Active Vision," University of Chicago *Technical Report CS 91-27,* November, 1991.

[12] Von Seelen, W. and Janβen, H. "Structural principles in visually guided autonomous vehicles," 11^{th} *International Association for Pattern Recognition (IAPR),* The Hague. The Netherlands, August 30 - September 3, 1992.