Proceedings of the 1998 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Victoria, B.C. Canada October 1998 pp:1808-1813

[1]An Automated Inspection System With Biologically Inspired Vision

Hülya Yalçın & H.Işıl Bozma

Intelligent Systems Laboratory[2]
Department of Electric Electronic Engineering
Boğaziçi University, Bebek 80815 İstanbul, Turkey

Abstract

This paper describes BUVİS -an automated visual inspection system – which selectively processes the incoming image data and combines vision with behavior. For this, BUVİS is endowed with biologically motivated visual capabilites: oculomation, attention and spatio-temporal reasoning. The advantages of such a system are being real-time and robust while running on very simple hardware.

1. Introduction

The design of artificial systems has long been guided by disciplines like physiology, cognitive science and neurology [15, 16, 17]. Studies have revealed that such systems must have oculomotion, attention and spatio-temporal reasoning capabilities [8, 9, 18, 19, 20]. Vision researchers haven then proposed selective perception in pursuit of developing systems mimicking some of this behavior [11, 10, 4, 21, 22, 23]. Interestingly, few studies have focused on employing selective perception mechanisms in higher level tasks. One such example has been APES [3]. In our work, we extend these ideas to automated visual inspection – a task assumed to be simple, yet still posing problems in real-time and robust applicability in factory manufacturing settings - and investigate the possibility of overcoming these problems. The novelty of our vision system is that it can direct its attention to spatial points of interest and both the temporal and spatial nature of the visual information thus gathered is used in the accomplishment of its task.

The inspection setup is as shown in Figure 1. Metal car parts - odd shaped and having several holes and extrusions (hence are not necessarily planar) - are placed on an assembly line at an *arbitrary* position and orientation. A camera located exactly above the assembly line views the objects orthographically. Let us remark that despite orthographic viewing, since the parts are not planar and are arbitrarily positioned, there is significant perspective distortion on the image plane. The goal of the inspection is to determine whether all the holes on a given part are located correctly and of correct shape within acceptable tolerances.

1.1. Related Work: Object Recognition

There have been hundreds of articles describing various methods for 2D object recognition [24,25] - based on contour or region descriptors. Most of these papers – although having well-thought schemes of representation – have nevertheless problems with real-time applicability (one or two seconds at most!) and robustness at the same time running on simple hardware. This motivated us to explore the use of biologically motivated ideas in object recognition.

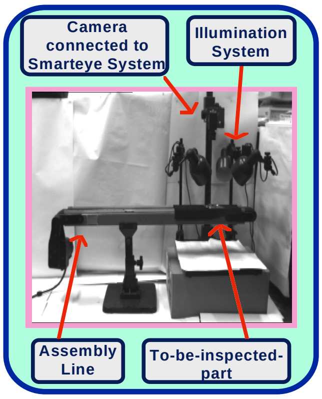

Figure 1: Visual inspection setup.

1.2. Related Work: Automated Visual Inspection

Due to the inherent problems with respect to real-timeness and robustness, automated visual inspection is one task within manufacturing that has been realizing at a comparatively slow pace [1,6]. Although such systems are available commercially for a wide variety of inspection tasks including automobile, electronics, metal industries, they have had nevertheless limited applicability in complex industrial environments. Furthermore, tasks requiring intensity based image processing have had problems regarding real-time operability. One common feature of such systems – also contributing to this problem – is that they usually process the whole image and use image subtraction or localized histogramming methods for defect detection [7].

[1] This project has been supported in part by Boğaziçi University Research Fund Project grants #AF 96A0236, and #AF 96HA0222 and TÜBİTAK grant Misag-65, 1995.

[2] http://www.isl.ee.boun.edu.tr

Proceedings of the 1998 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Victoria, B.C. Canada October 1998 pp:1808-1813

1.3. Related Work: Selective Vision

Physiological studies reveal two crucial features of biological vision: a visual field with a centered small region of high acuity (fovea) and a surrounding region of lower acuity (periphery) that together see only a limited part of the scene and oculomotion that enables the shift of the visual field to different regions of the scene [8,9]. Vision researchers - motivated by these findings - have then proposed selective vision where limited resources process only the most relevant parts of the incoming visual data [10,11]. Interestingly, studies focusing on integrating selective vision to actual tasks requiring recognition have been limited. One such work has been the integration of biologically motivated visual capabilities to robots engaged in physical motion [2,3]. In this paper, we present a real-time and robust automated inspection.

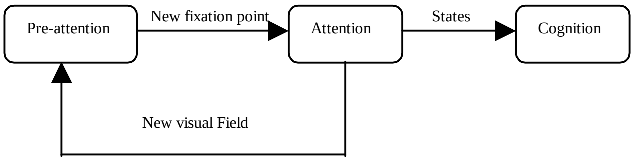

Figure 2. General flow of processing.

2. Visual Processing

The visual processing consists of three stages of operation: pre-attention, attention and cognition as shown in Figure 2. The aim of the pre-attention stage is to determine where to look next in the visual field. As a result of this behavior, visual resources are allocated to process only a small part of the whole scene. After the occurrence of physical attention, this region is subjected to further processing - in order to extract more complex features. These two stages of vision occur repeatedly – collecting data in space and time and generating an attentional sequence and thus contribute to the pool of information used by the cognition stage to accomplish the given inspection task. The attentional sequences thus collected are subjected to further processing to accomplish the given visual inspection task. In particular cognition two possible modes: 1.) Learning, where the system assumes that it is presented with an ideal part and forms a library model of the part; and 2.) Inspection, where the system compares the part-being-inspected to the library model and determines whether the part is faulty or not and if faulty, what the faults are.

2.1. Pre-attention: Where to Look Next

The aim of this stage is to determine where to look next in the image. In order to find next fixation point, simple computations are applied on the periphery region of the current fixation point by considering all candidate image points, computing their saliency - a measure of interest based on the presence of simple features with low computational requirements - and designating the image point with the greatest saliency as the potential fixation point. The periphery region is defined by a window of adaptive size. The window grows in four directions - up, down, right and left. The saccade direction - namely the direction from the previous fixation point to the current fixation point - determines which pixels in the periphery are subjected to further processing. In particular, those whose relative position wrt the current fixation point are in the direction of the saccade, are processed and are favored depending on their closeness to this direction. For the first fixation point of each contour segment - where a saccade direction is not yet defined - equal amount of growth in each direction occurs, as illustrated in Figure 3.

For each size of the window, a measure of saliency for each candidate next-fixation-point located on the window border is computed. In our case, saliency is defined as the weighted sum of the distance, d, and angular edge difference, $\Delta\theta$, between the current fixation point and point under consideration as well as the number of edge points in its 8-neighborhood. The first two measures assess the proximity of the two respective points with respect to nearness and edge orientation, while the third measure assesses the degree of centeredness of point under consideration. Window growth continues until either upper limit for the window is reached or a point with a saliency greater than a predefined threshold is detected.Once the window stops growing, the point with the maximum saliency measure is designated to be the next fixation point and is added to the current fixation point chain. It may happen that given a fixation point, the system cannot find any unlabeled edge points to take into consideration for computing saliency. In this case, the chain ends and an arbitrarily chosen nearby edge point is taken to be the initial fixation point of the next fixation point sequence with a new label. If such an initial point cannot be found, the process of fixation sequence generation ends.

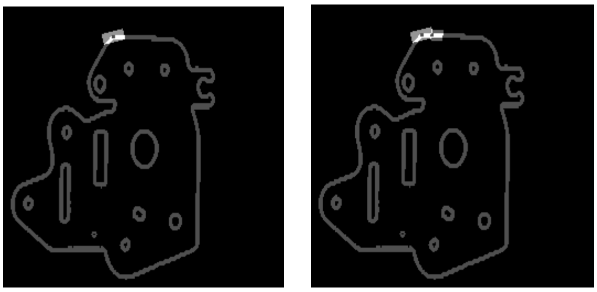

Figure 3. (Left) Detection of first fixation point and shift of visual field along the saccade direction; (Right) Detection of next fixation point.

2.2. Attention

In the attentive stage, the properties which charaterize the state of the fovea thus extracted. The feature vector obtained in this way is then added to the attentional sequence. The type of processing during attention is determined by the task at hand and is much more detailed in nature than that of the pre-attentive stage. In our case, we use edgetype as the feature. Edgetype is a valued entry consisting of edge magnitude and orientation. Let us remark that by choosing edgetype as feature, the saccades track the contour of the subparts. However, by choosing other types of features, other types of visual behavior can be realized.

2.3. Recognition Using Attentional Sequences

In the cognition stage, the attentional sequences thus generated are subjected to further processing. An attentional sequence corresponds to a spatio-temporal representation of the image – since it not only represents the locations of the features, but also the manner in which they were traversed.

2.3.1. Grouping Attentional Sequences

The first type of cognitive processing aims to group attentional sequences together. In particular, two segments are possibly to be grouped if their end-points are close to each other. However, this is not an easy task. Fortunately, the representation of the segment as a chain of fixation points also provides such information approximately. The initial and the end fixation points may be taken roughly to be the end points of the segment and assessment of closedness of two segments can be made based on their physical and feature proximity, as shown in Figure 4 (Left). In our case, we use distance as a measure.

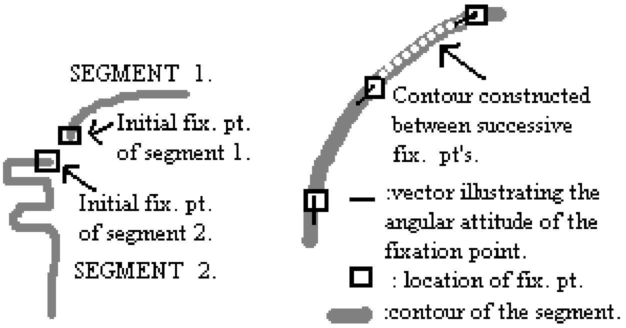
SEGMENT 1.
Initial fix. pt. of segment 1.
Initial fix. pt. of segment 2.
SEGMENT 2.
Contour constructed between successive fix. pt's.
:vector illustrating the angular attitude of the fixation point.
: location of fix. pt.
:contour of the segment.

Figure 4: (Left) Merging and (Right) contour construction.

2.3.2. Interpolation of Grouped Attentional Sequences: Contour Construction

Next, the grouped attentional sequences are used in an interpolation aimed at constructing the contours of all the subparts. As fixation points are sequentially ordered, this task is reduced to a series of - given two fixation points - determining a connected set of edge points starting from the first and ending at the second fixation point as illustrated in Figure 4 (Right). All those image points whose edges are within a predefined threshold proximity to that of the current fixation point and are connected to the current fixation point taken to be part of the contour. Two image points are connected to each other if there is an 8-connectivity path from one to the other and all the image points along the path have the same label. Let us remark that due to the adaptively growing window and connectivity-based interpolation, isolated image points do not pose any problems.

2.3.3. Shape Representation

The representation of shapes is based on elliptic Fourier descriptors [12,13]. Elliptic Fourier descriptors represent a shape weighted sum of ellipsoids. Using elliptic Fourier descriptors, each shape *i* is defined by a vector $p_i \in P \subseteq \Re^{4k+2}$ where *k* is the number of harmonics:

$$p_i = \left[a_{i0} b_{i0} a_{i1} b_{i1}, c_{i1} d_{i1} \ldots a_{ik} b_{ik}, c_{ik} d_{ik}\right]^T$$

The order of harmonics *k* represents the accuracy of the model. The set of *n* shapes is then described by $p \in P \subseteq \Re^{n(4k+2)}$ via concatenating p_i [14]. In our case, *n* = 5. The parameter set a_{ik}, b_{ik}, c_{ik} and d_{ik} can be used to extract geometric features such as major and minor axis length, orientation. Furthermore, geometric relations within and between shapes can be easily formulated mathematically. Finally, given a sequence of points forming a complete contour, a simple procedure can be used to compute the elliptic Fourier parameters [13].

2.3.4. Construction of Part Model

Once the shape descriptors are extracted, the next stage is to examine all the subparts on the part and then construct a part model that represents where all the subparts are located on the part. Note that as the part is randomly positioned and oriented, this is not an easy task. We use shape invariants for representing each sub-part [12] and then use a particular -Euclidean-invariant graph for constructing the complete part model. For shape invariants, we use the major and minor axis lengths of the ellipse (corresponding to second order moments) passing roughly through the contour of each subpart as computed based on elliptic Fourier descriptors. To construct our Euclidean-invariant graph, we first compute the vector anchored at the center point of the part in the direction of the center point of a subpart which is assumed to be always present, use this vector to generate a rotated x-y coordinate frame and calculate the radial coordinates (r_{pi}, θ_{pi}) of the center point of each subpart i=1,..,N where N is the number of subparts as shown in Figure 5.

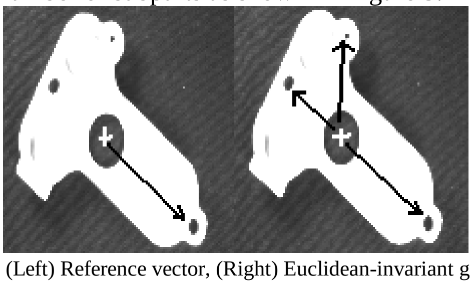

Figure 5. (Left) Reference vector, (Right) Euclidean-invariant graph.

2.3.5. Decision Stage

The decision stage has two modes. In the learning mode the system is presented with a non-defective part. Part's model is constructed and then stored in a library file to be later used. In the inspection mode, a "to-be-inspected" part is presented to the system and a decision regarding whether the part is defective or not is made. If found defective, the defective parts are listed. In order to accomplish this task, a model of the part is constructed and then compared with the model of the 'ideal' part. The comparison is based on identifying corresponding subparts on each respectively and then using a measure of proximity to determine whether the positioning and shape of each subpart on the inspected part is as it should be. In our Euclidean-invariant graph representation of the parts, a mere comparison in radial coordinates suffices to identify corresponding subparts. The j'th subpart on the model is matched with the i'th subpart on the part being inspected that minimizes the Spatial-Similarity-Criterion, SSC_{ji},

$$\mathrm{SSC}_{ji} = \left(\left(r_{pi} * \mathrm{Cos}(\theta_{pi}) - r_{mj} * \mathrm{Cos}(\theta_{mj}) \right)^2 + \left(r_{pi} * \mathrm{Sin}(\theta_{pi}) - r_{mj} * \mathrm{Sin}(\theta_{mj}) \right)^2 \right)^{0.5}$$

where r_{pi} and r_{mj} are the radial distance of i'th subpart of to-inspected-part and j'th subpart of model to the starting point of the reference vector respectively and θ_{pi} and θ_{mj} are the angular displacements of corresponding subparts relative to the reference vector. The average SSC_{ji} for all subparts, defined as follows,

$$\mathrm{ASSC} = \frac{1}{N} \sum_{j}^{N} \mathrm{SSC}_{j(\mathrm{matched-subpart-on-inspected-part})}$$

can be used as a feature to determine if a subpart is missing. The ASSC value is almost invariant for a part no matter what its orientation is. Due to a missing subpart (i.e. a missing hole), the value of ASSC will change considerably compared with the model's ASSC. Having determined that there exists a missing subpart, the missing subpart can easily be identified through SSC_{ji} values.

Having matched the segments on both images, our next objective is to determine whether the segments are properly shaped within some tolerance. The invariants computed out of first few harmonics successfully aids in rough inspection of the shape of the segments.

3. Experiments

We have developed an inspection system -BUVIS - based on TMS320C31 where visual processing is based on our approach. The system components are shown in Figure 6. Visual processing is done on the Smarteye Vision System which is designed around a high performance DSP chip TMS320C31PQL. Computationally intensive parts of the program are directly programmed in TI assembly language. The remaining parts are programmed in C and then cross-compiled to TI assembly code. The illumination system consists of four lamps located so as to minimize the shadowing effects of each lamp. An image of the 'to-be-inspected' part is taken and is subjected to further inspection to determine whether it has any deviations from its CAD specs.

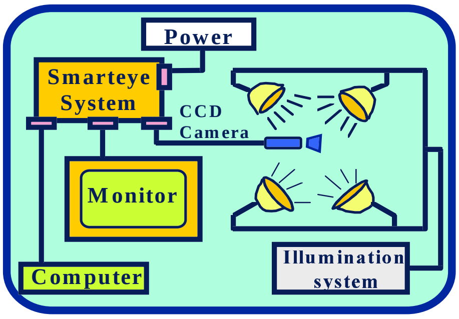

Figure 6 . Inspection system components .

Experiments were held on several industrial parts. Let us note that although these images look like high-contrast images due to the printing quality, in reality they are pretty noisy. In these experiments, the center of the boundary of the part is the reference point and the reference vector is directed to the center of a subpart (or hole) which is known to exist on every part whether the part is defective or not. The ASSC value, explained in section 2.6, is computed for different orientations of each part's model and defective samples of the part. Let us remark that the difference in ASSC value between normal and defective parts can be used to determine a threshold for deciding whether a given part is acceptable or not.

3.1 Experiments with Part 1

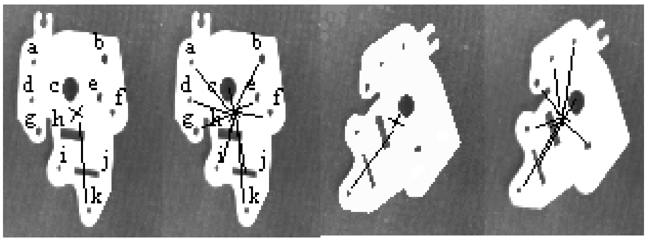

Figure 7. PART1 (a) Model and reference vector; (b) Euclidean-invariant graph of the model; (c) and (d) reference vector and Euclidean-invariant graph for faulty sample where holes b and f are missing.

The part in Figure 7.a, out of which the model is constructed is rotated around itself, to measure the variances of placements of subparts on a part at different orientations and the results are displayed in Figure 8.

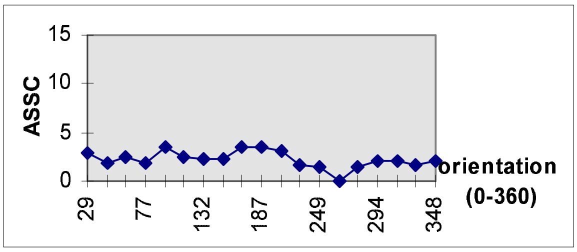

Figure 8. ASSC vs orientation graph for Model of Part1 .

The model of the industrial part was constructed with a randomly selected orientation at 270 degrees. The part is then rotated around itself and ASSC at every orientation is computed as shown in Figure 8. The average similarity measure varies between 0-4 pixels per subpart. Let us remark that ASSC does not turn out to be zero because from the perspective of the camera, the transformation of the part is not completely Euclidean-invariant. Rather, as our parts have extrusions and holes on them, foreshortening effects come into play and distort the image very slightly – thus causing variation of the ASSC. We can use the upperbound value 4 as the maximal tolerance for mismatching. The faulty sample shown in Figure 7.c is compared with the model shown in Figure 7.a . ASSC values for different orientations for the faulty part is shown in Figure 9. The ASSC value varies between 5-8, -all above the upper bound. Furthermore the variations (not shown in the graph) are much larger. In addition to detecting faulty parts, the system easily identifies the faulty subparts.

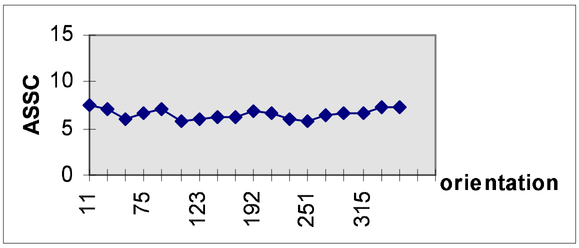

Figure 9. The ASSC vs orientation graph for an erroneous Part1.

It takes the system approximately 4.5 seconds to perform the whole process and the inspection duration does not change significantly for varying orientations of the part. The experiment is repeated for parts with many different types of faults. As expected, as the number of faults of type 'missing holes' increases, ASSC also increases.

3.2 Experiments with Part 2

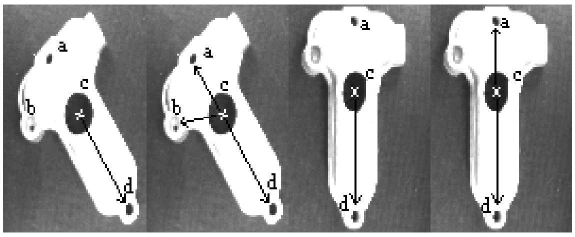

Figure 10. PART2 (a) Model and reference vector; (b) Euclidean-invariant graph of the model; (c) and (d) reference vector and Euclidean-invariant graph for faulty sample with hole b.

Similar experiments were held for the industrial part, Part2 - shown in Figure 10.a. The model of the industrial part was constructed with a randomly selected orientation at 300 degrees. The part is rotated around itself in order to measure the variances of placements of subparts on a part at different orientations and the results are displayed in Figure 11. It is observed that ASSC varies between 0-6. Let us remark that there is more variation as compared to Part 1 due to the increased foreshortening effects caused by the uneven surface of Part 2. Again, we can use the upperbound value 6 as the maximal tolerance for mismatching.

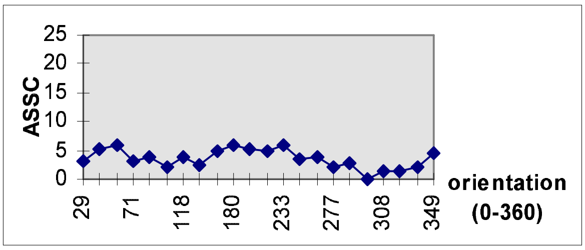

Figure 11. ASSC vs orientation graph for Model of Part2.

The faulty sample of Part2 shown in Figure 10.c, is compared with the model shown in Figure 10.a . The experiment is repeated for varying orientations of the faulty sample. ASSC values for different orientations of erroneous Part 2 object are shown in Figure 12. The ASSC value varies between 10-15, -all above the upper bound. Furthermore the variations (not shown in the graph) are much larger. Again, additionally the system easily identifies the faulty subparts of the part currently being inspected. Experiments with faulty parts reveal that the greater the number of faults, the greater ASSC value gets.

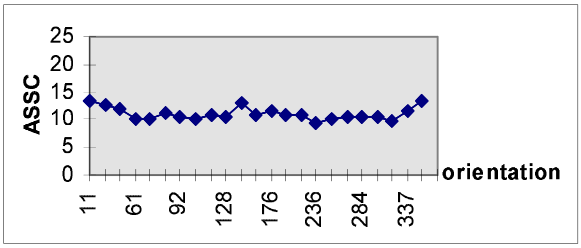

Figure 12. The ASSC vs orientation graph for an erroneous Part 2.

It takes the system approximately 2.5 seconds to perform the whole process and the performance duration does not change significantly for varying orientations of the part. Let us remark that using college students as inspectors, we tried to compare human inspection times with our system. The inspection time for first few parts is about 7-8 seconds for this part, Part 2. Due to the learning ability of human, the inspection duration reduces to 2 seconds after few parts. However, after a certain period of time, due to fatigue and tediousness of the task, the inspectors were observed to have a loss of concentration and the inspection time goes back to initial durations.

4. Summary

In this paper, BUVİS – an automated visual inspection system endowed with visual attention capability – is presented. The advantages of this system are that it is fast and robust while running on simple hardware. In this system, the visual processing consists of a continuum of pre-attentive and attentive stages, and generates an attentional sequence which represents the visual data spatio-temporally. This processing is occasionally followed by cognition, where attentional sequences are grouped together, the grouped attentional sequences are used as anchor points for contour interpolation, shape parameters of all subparts are extracted and compared with the library models for fault detection. We are currently working on reducing inspection times even further. Also, we are investigating the use of optimal control ideas in bringing a mathematical formulation to the problem of attentional sequence generation.

5. Acknowledgements

The TI assembly programming of BUVIS has been designed and realized by Mehmet Tezol (tezolmeh@boun.edu.tr).

6. References

1. Newman, T.S. and A. Jain. *A Survey of Automated Vİsual Inspection. Computer Vision and Image Understanding.* Vol 61, No.2, March pp:231-262,1995.

2. Çağatay Soyer, H. Işıl Bozma. *Further experiments in classification of attentional sequences: Combining instantaneous and temporal evidence*, Proceedings of ICAR'97, California, USA.

3. Ç. Soyer, H. I. Bozma, Y. İstefanopulos. *A mobile Robot With a Biologically Motivated Vision System,* Proceedings of IROS'96,JAPAN.

4. A. L. Abbott. *A survey of selective fixation control for machine vision.* IEEE Control Systems, 1992.

5. Carl Fredrik Westin. *Attention Control for Robot Vision*. Processing of CVPR' 96, pp 726.

6. Kennedy, C.W. and E.G. Hoffman and S.D. Bond. Inspection and Gaging. Industrial Press, New York, 1987.

7. Batchelor, B.G. and D.W. Braggins. Commercial vision systems, in *Computer Vision: Theory and Industrial Applications* (Torras, Ed.), pp.405-452, Springer-Verlag, New York, 1992.

8. P. Gouras. *Oculomotor System*. In Principles of Neural Science, editors J.H. Schwartz and E.R.Kandel. Elsevier, 1988.

9. Kelly, J.P. *Anatomy of central visual pathways*. In Principles of Neural Science, editors J.H. Schwartz and E.R.Kandel. Elsevier, 1988.

10. Ballard, D.H. and C.M. Brown. Principles of Animate Vision. CVIP: Image Understanding, 56(1), July 1992.

11. Ballard, D.H. Animate Vision. Artificial Intelligence,48:57-86, '91.

12. C. Lin, C. Hwang. *New Forms of Shape Invariants from Elliptic Fourier Descriptors*, Pattern Recognition. V.20, No. 5, pp. 535-545,'87.

13. Frank P. Kuhl. *Elliptic Fourier Features of a Closed Contour*, Computer Graphics and Image Processing 18, 236-258 (1982).

14. R. Safaee-Rad. *Application of Moment and Fourier Descriptors for the Accurate Estimation of Elliptical Shape Parameters.* IEEE Robotics and Automation 1991.

15. D.H. Ballard and C. Brown. *Computer Vision*. Prentice Hall Inc.' 82.

16. R. Haralick and L. Shapiro. *Computer Vision*. Prentice Hall Inc.' 85.

17. H. Barrow and J.M. Tenenbaum. *Computational Vision*. Proceedings of IEEE, 69:572-575, 1981.

18. D. Noton and L. Stark. *Eye movements and visual perception*. Scientific American, 224(6), 1971.

19. S. Zeki .*The visual image in mind and brain*. Scientific American, 267(3):69-76, 1992.

20. D.H. Hubel. *Eye, Brain and Vision*. Scientific American Library, 1988.

21. R. D. Rimey and C. M. Brown. Selective attention as sequential behavior: Modelling eye movements with an augmented hidden markov model. Technical report, the university of Rochester, Computer Science dept., February 1990.

22. L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), February 1989.

23. M. J. Swain and M. A. Stricker. Promising directions in active vision. International Journal of Computer Vision, 11(2): 1090126, 1993.

24. S. Ullman. High-Level Vision, MIT 1996.

25. R. Chellappa, A. Sawchuk, Digital Image Processing and Analysis. IEEE Catalog No. EHO232-9, 1985.