BUVIS - A Novel Approach to Real-Time Automated Visual Inspection

Hülya Yalçın

Mehmet Tezol

H. Işıl Bozma

Çağ Onganer

Intelligent Systems Laboratory

Department of Electrical&Electronics Engineering

Bogazici University, Istanbul, Turkey

October 1997

Contents

ABSTRACT 1

1. Introduction 1
    1.1 Issues That Need to be Resolved 2
    1.2 Related Work: Automated Visual Inspection 2
    1.3 Related Work: Selective Vision 2
    1.4 Inspection Setup 2
    1.5 Visual Processor 4
    1.6 Contributions 4

2. Visual Processing 5
    2.1 Segmentation as a Fixation Sequence Generation 6
    2.2 Merging 8
    2.3 Contour Construction 8
    2.4 Shape Representation: Elliptic Fourier Descriptors 9
    2.5 Construction of Part Model 10
    2.6 Learning Mode 10
    2.7 Inspection Mode 11
    2.8 Experiments 13
        2.8.1 A Complete Cycle of the Inspection Process 13
        2.8.2 Inspection of Randomly Oriented Parts 14

3. Summary 16

References 17

Appendix A Aspect Ratio Compensation 1

Appendix B TMS320 Implementations 1

Appendix C Glossary 1

Appendix D Visual Processing C Code ..... **Error! Bookmark not defined.**

List of Figures

1 Figure 1: Inspection setup. 3
2 Figure 2: Inspection of various parts. 3
3 Figure 3: Inspection system components. 4
4 Figure 4: General Flow Diagram. 5
5 Figure 5: (Left) An industrial object - door part manufactured for automobile industry; (Center) Image after low-pass filtering; (Right) Image after edge extraction. 5
6 Figure 6: (Left)Detection of first fixation point and shift of visual field along the saccade direction; (Right) Detection of next fixation point. 6
7 Figure 7: The selection of next fixation point 7
8 Figure 8: Results of segmentation process. Note that each segment is represented by a different gray value and fixation points detected are also displayed. 8
9 Figure 9: Merging example. 8
10 Figure 10: Contour construction. 9
11 Figure 11: Constructed contours. 9
12 Figure 12: (Left) The ‘ideal’ part; (Right) To-be-inspected part. 11
13 Figure 13: (Left) The reference vector. (Right) Euclidean-invariant part model. 11
14 Figure 14. Construction of Euclidean-invariant part model for the to-be-inspected part. 12
15 Figure 15 (Left) Another industrial object manufactured for automobile industry; (Center) Image after low-pass filtering; (Right) Image after edge extraction. 13
16 Figure 16. Results of segmentation process. Note that each segment is represented by a different gray value and fixation points detected are also displayed. 13
17 Figure 17 Contour constructed between successive fixation poin 14
18 Figure 18. Contour reconstructed out of efd coefficients using 15 harmonics. 14
19 Figure 19 (Left) The reference vector. (Right) Euclidean-invariant part model. 15
20 Figure 20 Construction of Euclidean-invariant part model for the to-be-inspected part. 15
21 Figure 21 . (Left) An ellipse with major to minor axis ratio of 4/3, (right) the ellipse is mapped like circle to videobuffer due to the aspect ratio problem. 1
22 Figure 22 (Left) An ellipse with major to minor axis ratio of 4/3, (Right) the ellipse is mapped thinner to videobuffer due to the aspect ratio problem. 1
23 Figure 23. Smarteye card components. 1

List of Tables

1 Table 1.Memory Map 4
2 Table 2.Euclidean-Invariant Part Model (Ideal Image). 12
3 Table 3: Relative Placement of subparts of incoming image wrt reference vector 12

ABSTRACT

This report presents a novel system - BUVIS - geared for the real-time automated inspection of industrial parts for post-manufacturing quality control. The aim of the visual inspection is to determine whether all the holes on the part are located correctly and of correct shape within acceptable tolerances while ensuring that the completion time is comparable to that of an human inspector. The novelty of the system is that it is based on an event-driven approach to visual processing and decision-making. In the Phase 1 of this project, an algorithm that selectively processes the incoming visual data - hence being event-driven - and finds all the holes on the given part has been developed. In the second phase of the project, this algorithm has been extended to infer the positions and shapes of all the holes detected. In the final phase of the project, two modes of operation has been added to the system: (i) The learning mode during which a model part is presented to the system which then forms an internal model of the holes and their locations, (ii) the inspection mode during which a 'to-be-quality-controlled' part is inspected and compared with the model part to determine its manufacturing quality.

1. Introduction

This report presents a novel method to real-time automated visual inspection of industrial parts for post-manufacturing quality control based on an event-driven approach to processing and decision-making. Inspection involves determining if a part[1] deviates from a given set of specifications [18]. In our case, the inspection scenario is as follows: Metal car parts - odd shaped and having several holes and extrusions on them - are placed on an assembly line at an arbitrary position and orientation and arrive to the inspection booth. 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. For automated inspection to be feasible, the inspection time should be comparable to that of a human inspector and be reliable [16].

[1] Also referred to as an object, product, both in this paper and literature.

The novelty of our system is that - in contrast to traditional systems which process the whole image - our visual processing is based on selective vision concepts where only areas around certain 'interesting' points are subjected to processing [1,2,3,4].

1.1 Issues That Need to be Resolved

An automated inspection system needs to address several issues effectively if it is to be applicable.

- The parts to be inspected may be odd-shaped.
- Parts may be of metal and have specular reflection.
- The parts' exact position and orientation are arbitrary.
- Real-time inspection with a time comparable to that of a human inspector.
- Processing simplicity due to limited hardware memory.

1.2 Related Work: Automated Visual Inspection

Automated visual inspection is one task within manufacturing that has been realizing at a comparatively slow pace [16]. The advantages of automation have been listed as: 100% inspection, consistency and higher accuracy. Commercial vision systems are available for a wide variety of inspection tasks including automobile, electronics, metal industries and have a taxanomy based on type of image used - binary, intensity, color and range. Intensity based systems - although of great potential - has had limited applicability in complex industrial environments. Such systems usually process the whole image and use image subtraction or localized histogramming methods for defect detection [17].

1.3 Related Work: Selective Vision

Physiological studies reveal biological vision has two crucial features distinguishing it from that of machine vision systems: 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 [12,13]. Selective vision systems - motivated by these findings - allocate their limited resources to process only the most relevant parts of the incoming visual data [14,15]. In this perspective, selective perception consists of a series of pre-attention and attention followed by cognition that enables the accomplishment of the given task. The use of these ideas in robotic system has been presented in [1,2]. In this work, we design and develop a new method to inspection that selectively processes the incoming image and yields reliable and fast inspection.

1.4 Inspection Setup

The inspection set is shown in Figures 1-2. The system components are shown in Figure 3. A camera located exactly above the assembly line views the assembly line orthographically. The illumination system, consists

of four lamps located 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.

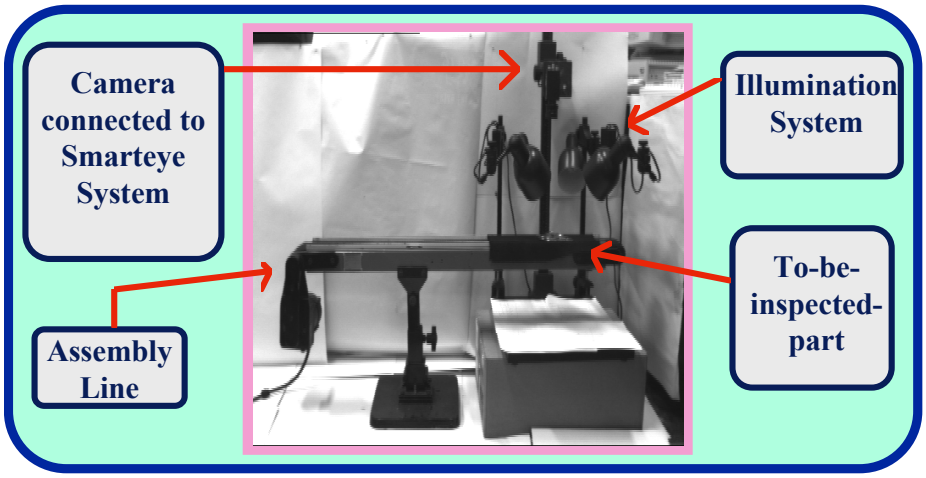


Figure 1: Inspection setup.

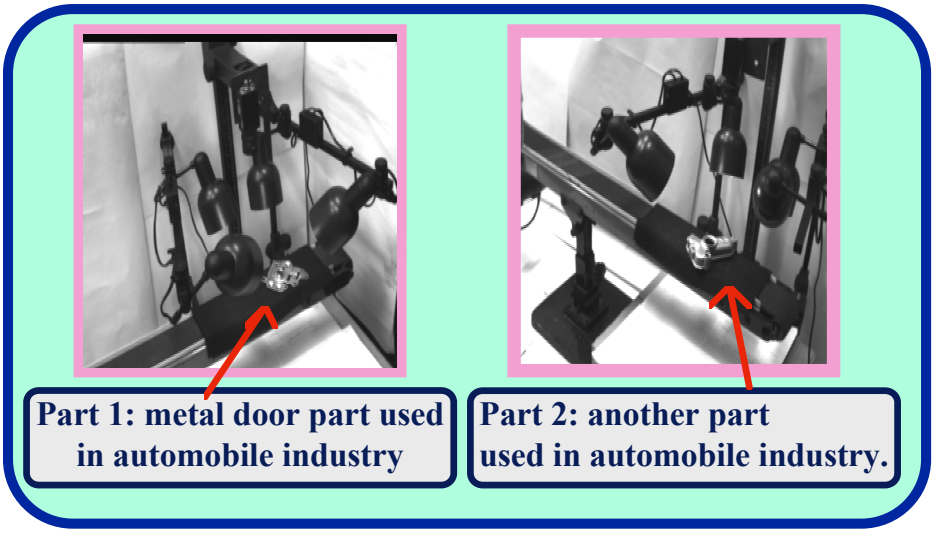


Figure 2: Inspection of various parts.

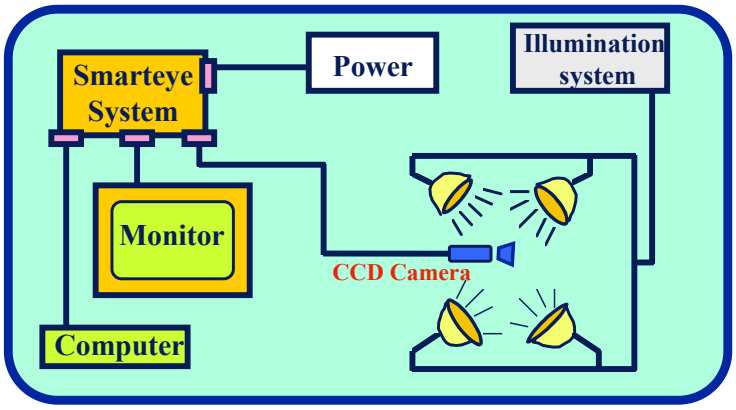


Figure 3: Inspection system components.

1.5 Visual Processor

Visual processing is done on the Smarteye Vision System which is designed around a high performance DSP chip TMS320C31PQL. The internal bus of the system is 24-bits with an address space of 16mwords. The internal data bus is 32-bits. The memory map is given in Table 1.

Table 1. Memory Map

ALLOCATION	START(HEX)	END(HEX)	WORD LENGTH (BITS)
Eprom	00 0000	00 7FFF	32
Program/Data Ram	10 000	10 7FFF	32
Video Buffer 0	20 0000	23 FFFF	8
Video Buffer 1	30 0000	33 FFFF	8
On-Chip Program/Data Ram	80 9800	80 9FFFF	32
Input Port	8F FFFF	8F FFFF	8
Image Display Memory	90 0000	93 FFFF	8
Graphics Display Memory	A0 0000	A3 FFFF	4
Framegrabber 0	B0 0000	B3 FFFF	8
Framegrabber 1	C0 0000	C3 FFFF	8

1.6 Contributions

The contributions of this project can be summarized as follows:

- A novel method - online and reliable - to image analysis of a complex part based on selective vision concepts for inspection purposes.
- Design and implementation of this system on TMS320C31PQL processor.

2. Visual Processing

The flow of processing is as shown in Figure 1.

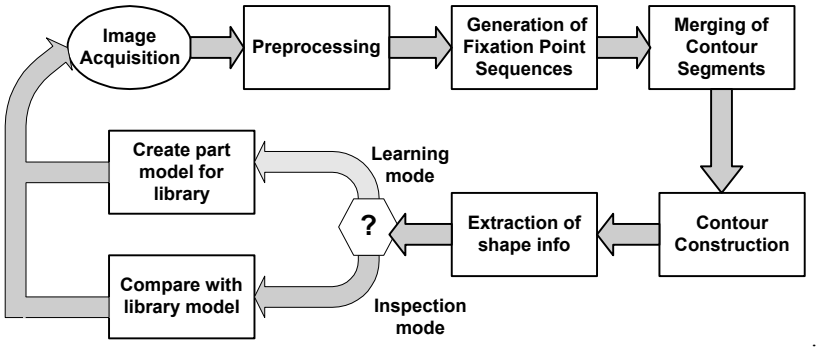


Figure 4: General Flow Diagram.

First, the incoming image is subjected to preprocessing with two purposes in mind: 1.) Removal of image noise via low-pass filtering and 2.) Extraction of prominent edges via high-pass filtering and thresholding. In our application, an averaging operator - box filter - is used for noise removal due to its computational simplicity and Sobel edge operator for edge extraction. Typical results are as shown in Figure 5.

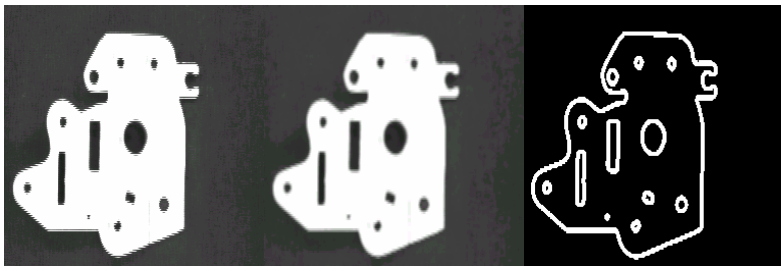


Figure 5: (Left) An industrial object - door part manufactured for automobile industry; (Center) Image after low-pass filtering; (Right) Image after edge extraction.

Next, the edge points are grouped into contour segments. The aim of this stage is to determine all the salient contour segments. Instead of having an explicit segmentation algorithm as traditional systems have, we have an implicit means of extracting contour segments via the generation of fixation point sequences. Each sequence coarsely represents a salient contour segment and can be used as anchor points for extracting the contour precisely. Due to illumination and viewing artifacts, the contour of a part or a sub-part may be seen as broken into several salient contour segments. To deal with such problems a merging algorithm is applied after segmentation.

Once merging is completed, the contour of each part or subpart is defined coarsely. Since we need accurate shape information for inspection purposes, each fixation point sequence can then be used to interpolate

the contour precisely. After this stage, we are ready to represent the shape of the part and all of its subparts for shape extraction. For this purpose, elliptic Fourier descriptors are used. The advantage of this representation is that given a shape, we can represent it to a degree of desired accuracy simply via deciding on the order of the parameters and furthermore extract useful geometric features such as center, area, etc.

The final stage has 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 Segmentation as a Fixation Sequence Generation

The aim of this stage is to determine all the salient contour segments - given the preprocessed image containing the edge information. A salient contour segment is defined coarsely by a chain of fixation points. Depending on the maximum size of the fovea, the actual location of these fixation points may be near or distant from each other.

To start visual processing, an arbitrarily chosen edge point - in our case the upper left most edge point - is taken to be initial fixation point of the first salient contour segment. To associate it with this contour segment, its label is changed. This is illustrated in Figure 6. Once an initial fixation point is determined, a find-next-fixation-point procedure is continually invoked to form a chain of fixation points until the chain is broken due to gaps or high-rate changes in certain contour properties.

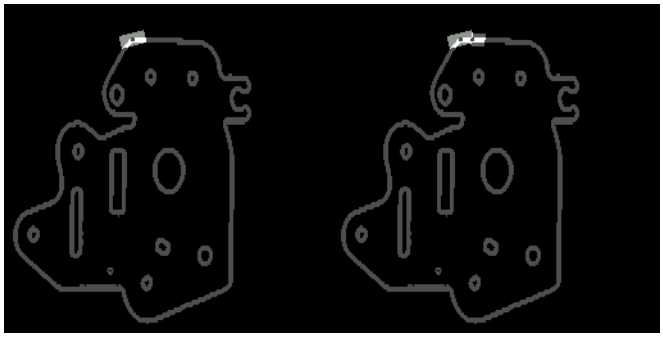


Figure 6: (Left)Detection of first fixation point and shift of visual field along the saccade direction; (Right) Detection of next fixation point.

In this stage, simple computations are applied on the periphery region of the current fixation point in order to find next point of interest by considering all candidate image points - namely edge points without any labels -, 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 as shown in Figure 7. 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.

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. Labeling of contour segments also occur during this process. 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 are assigned the label of the current fixation point. 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 this adaptively growing window and connectivity-based labeling, isolated image points are not labeled wrongly. 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. In Figure 8. the result of segmentation is illustrated.

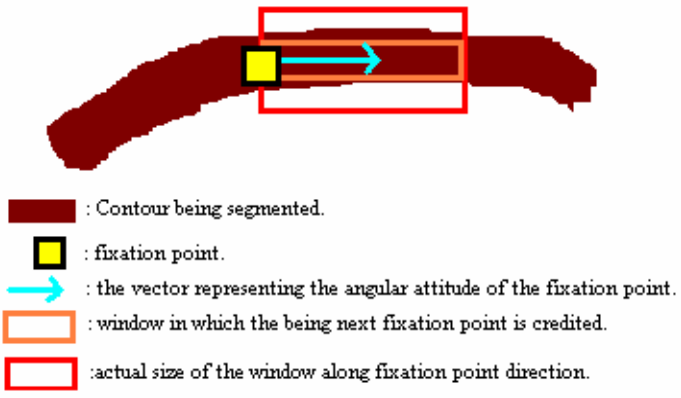


Figure 7: The selection of next fixation point .

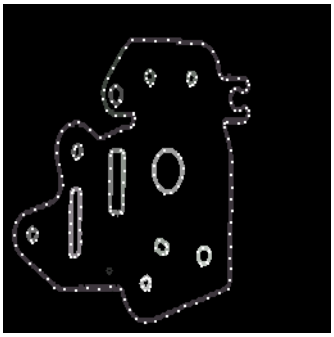


Figure 8: Results of segmentation process. Note that each segment is represented by a different gray value and fixation points detected are also displayed.

2.2 Merging

The merging process aims to group segments that of part of the same complete contour. In particular, two segments are possibly to be merged if their endpoints 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 endpoints of the segment and assessment of closedness of two segments can be made based on their physical and feature proximity. In our case, we use distance as a measure.

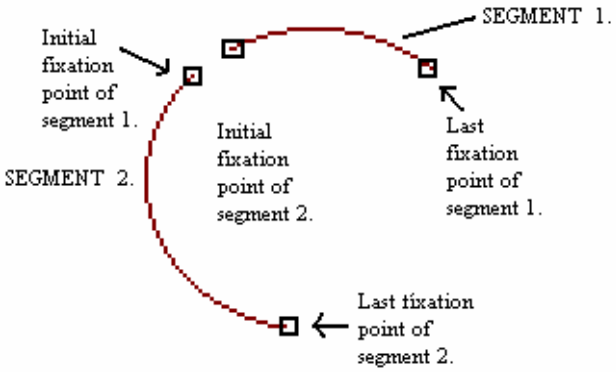


Figure 9: Merging example.

2.3 Contour Construction

The aim of contour construction is - given only the sequence of fixation points lying on a contour - to form a sequence of connected image points forming that contour. 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 shown in Figure 10., starting from the first fixation point, the next point of the contour is chosen among the neighbours of previous contour point. The new fixation point must be closer to the end fixation point while

preserving angular continuity to some measure. The final contours constructed are shown in Figure 11.

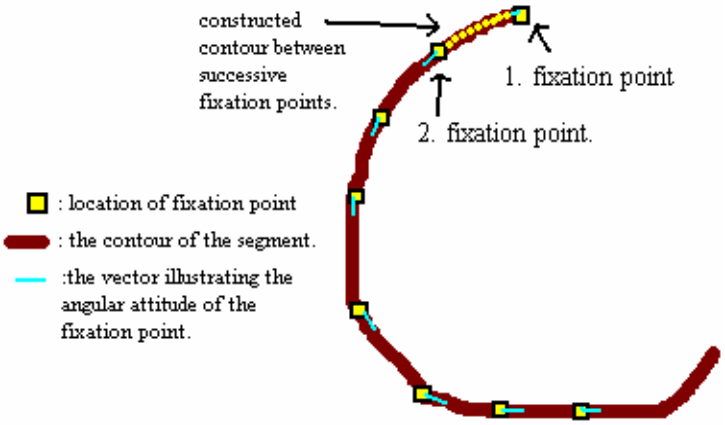


Figure 10: Contour construction.

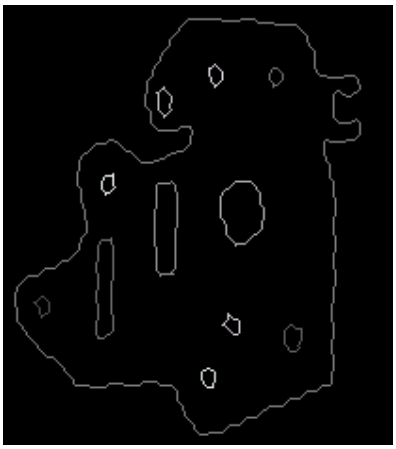


Figure 11: Constructed contours.

2.4 Shape Representation: Elliptic Fourier Descriptors

The representation of shapes is based on elliptic Fourier descriptors [10,11]. 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 [9]. 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 [11].

$$a_n=\frac{T}{2\pi^2n^2}\sum_{p=1}^{K}\frac{\Delta x_p}{\Delta t_p}\left[\cos\frac{2n\pi t_p}{T}-\cos\frac{2n\pi t_{p-1}}{T}\right] \quad (1.1)$$

$$b_n=\frac{T}{2\pi^2n^2}\sum_{p=1}^{K}\frac{\Delta x_p}{\Delta t_p}\left[\sin\frac{2n\pi t_p}{T}-\sin\frac{2n\pi t_{p-1}}{T}\right] \quad (1.2)$$

$$c_n=\frac{T}{2\pi^2n^2}\sum_{p=1}^{K}\frac{\Delta y_p}{\Delta t_p}\left[\cos\frac{2n\pi t_p}{T}-\cos\frac{2n\pi t_{p-1}}{T}\right] \quad (1.3)$$

$$d_n=\frac{T}{2\pi^2n^2}\sum_{p=1}^{K}\frac{\Delta y_p}{\Delta t_p}\left[\sin\frac{2n\pi t_p}{T}-\sin\frac{2n\pi t_{p-1}}{T}\right] \quad (1.4)$$

Δx_p and Δy_p are incremental changes of the contour in x and y directions during the time Δt_p . T is the period to trace the contour once. n is the harmonic number. For each part or subpart contour, shape coefficients using 15 harmonics are computed.

2.5 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 [10] and then use a particular - Euclidean-invariant subparts graph for constructing the complete part model. For shape invariants, we use the major and minor axis lengths of the ellipse passing roughly through the contour of each subpart as computed based on elliptic Fourier descriptors. To construct our Euclidean-invariant subparts graph, we first compute the vector anchored at the center point of the part in the direction of the average center points of the subparts, use this vector to generate a rotated x-y coordinate frame and calculate the radial coordinates of the center point of each subpart.

2.6 Learning Mode

In the learning mode, the system is presented with an non-defective part. Part model is constructed and then stored in a library file to be later used.

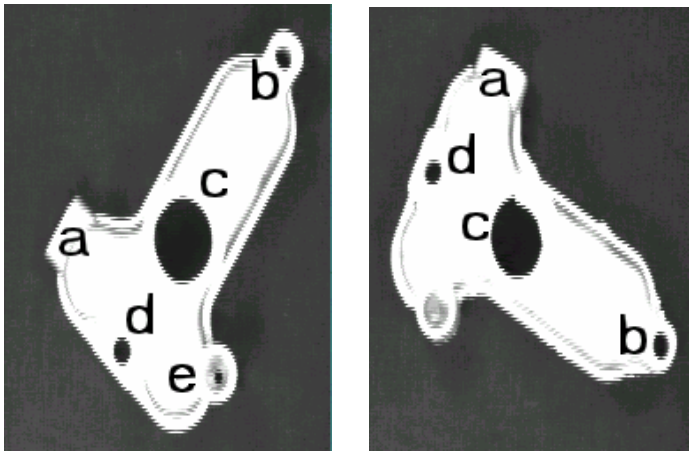


Figure 12: (Left) The 'ideal' part; (Right) To-be-inspected part.

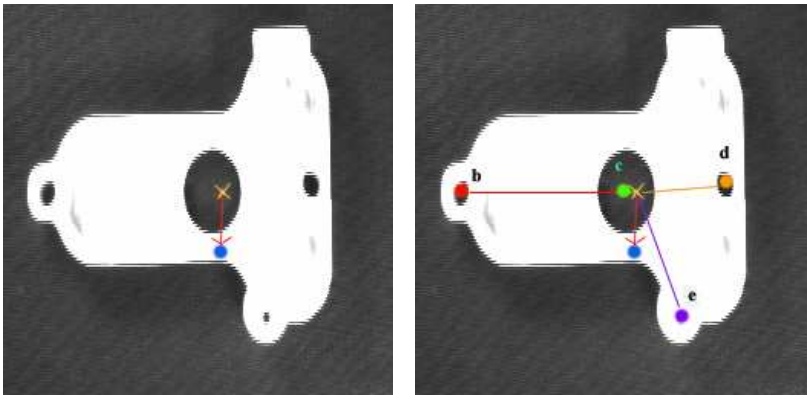


Figure 13: (Left) The reference vector. (Right) Euclidean-invariant part model.

2.7 Inspection Mode

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. It may be possible that due to a missing subpart (i.e. a missing hole), a match cannot be made. Once an identification is made, the shape invariants are compared to detect shape differences. Let us consider the case as shown in Figure 12. The letters indicate the labels of the probable segments on the object. Note that images are viewed at different orientations and "to be inspected" image has one of the holes missing, the one labeled as "e" in Figure 12(Right). The hole labeled as "c" in Figure 12(Right) is not properly shaped compared with that of the perfect model's in left Figure. The letter "a" is the label of outer boundary.

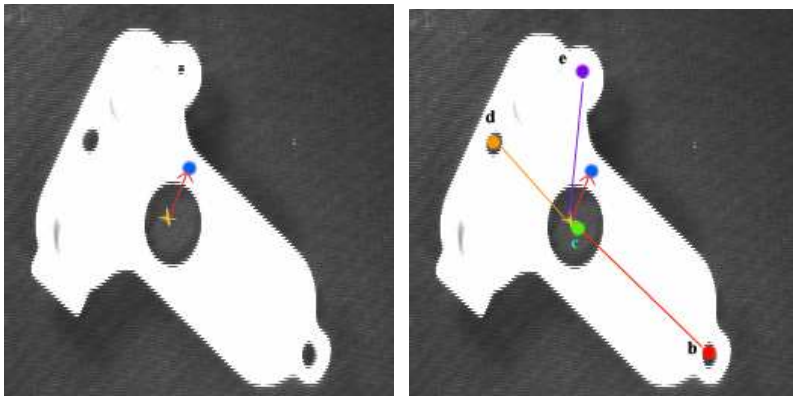


Figure 14. Construction of Euclidean-invariant part model for the to-be-inspected part.

Through Figures 13-14, the learning and the inspection mode of our system is illustrated. The part in Figure 13 is assigned to be the model part. The reference vector that originates from the center of outer part and directs to the center of subparts, shown in left Figure, and relative angular orientation and the distances of subparts with respect to the reference vector, shown in right Figure are stored in a library file to be used later. The elements of Euclidean-invariant part model and relative placement of subparts of to-be-inspected part is illustrated in Table 2 and Table 3 respectively.

Table 2. Euclidean-Invariant Part Model (Ideal Image).

LABELS OF SUBPARTS	RELATIVE ANGULAR ORIENTATION (0-360)	RELATIVE RADIAL DISTANCE (0-255)
a	0	23
b	77	155
c	341	20
d	277	102
e	213	82
length of reference vector	22	

Table 3: Relative Placement of subparts of incoming image wrt reference vector

LABELS OF SUBPARTS	RELATIVE ANGULAR ORIENTATION (0-360)	RELATIVE RADIAL DISTANCE (0-255)
a	359 (= -1)	22
b	82	154
c	355	20
d	283	101
e	219	84
length of reference vector	22	

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.

2.8 Experiments

The visual processing software has been developed using a combination of C and TMS320 Assembler on a Pentium 100 PC. The assembly code is presented in Appendix B. The code is cross-compiled, down-loaded to the visual processor.

2.8.1 A Complete Cycle of the Inspection Process

We experimented with a variety of parts. In this section, we will present typical results from the inspection of another part. The part is shown in Figure 15 (left) and has four holes (sub-parts). The samples to be inspected may possibly have missing or jaggedy holes. Results after low-pass filtering and edge finding are presented in the rest of the same figure. The elliptic Fourier coefficients for each of the subparts of the part upto 15th harmonic is computed. The contours reconstructed out of the elliptic Fourier coefficients are shown in Figure 18. The processing inspection time is 2.5 seconds. The inspection time of BUVIS for this industrial object is similar to that of an human inspector. The inspection time of a human inspector is about 7-8 seconds for the first few parts. Due to learning, the inspection time of a human inspector decreases to almost 2-seconds after being presented. However, this cannot be maintained for long and the performance of human inspector deteriorates rapidly after a period.

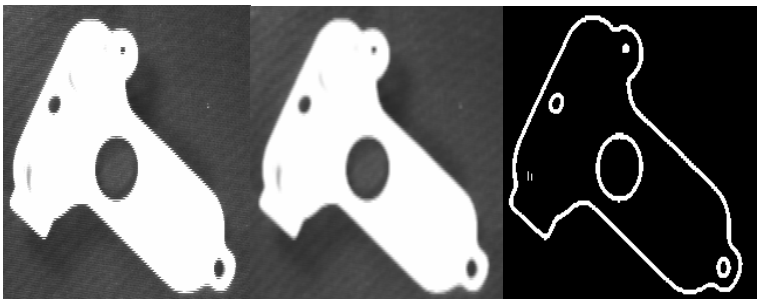


Figure 15 (Left) Another industrial object manufactured for automobile industry; (Center) Image after low-pass filtering; (Right) Image after edge extraction.

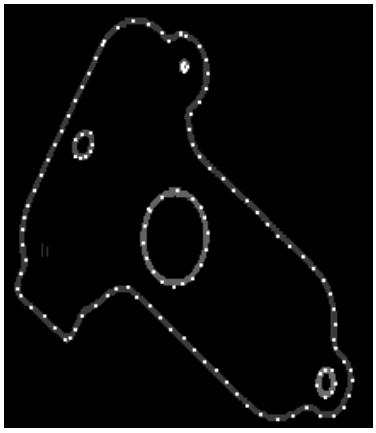


Figure 16. Results of segmentation process. Note that each segment is represented by a different gray value and fixation points detected are also displayed.

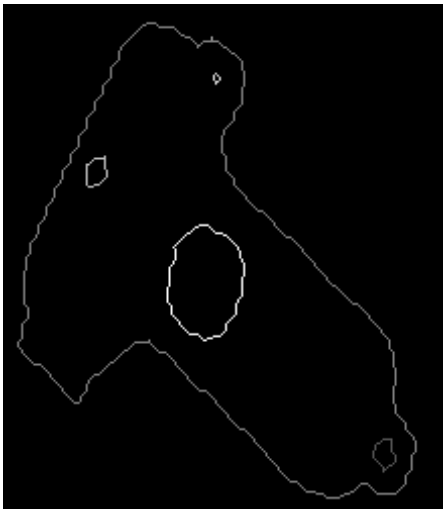


Figure 17 Contour constructed between successive fixation poin

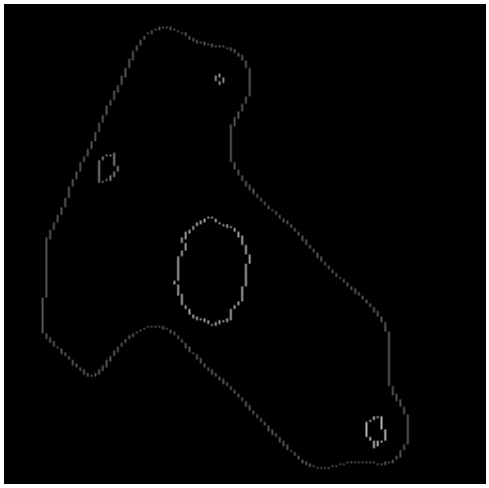


Figure 18. Contour reconstructed out of efd coefficients using 15 harmonics.

2.8.2 Inspection of Randomly Oriented Parts

Experiments were held on several industrial parts. The metal door part used in automobile industry is shown in Figure 19 (left). The construction of Euclidian-invariant graph of the model is illustrated in Figure 19. Relative angular orientations wrt reference vector and radial displacements of subparts from the initial point of reference vector are stored in memory. Another Euclidean-invariant graph, shown in Figure 20 (right), is constructed for the incoming image of a part to-be-inspected. The features of subparts of the part to-be -inspected are compared with those of the model and the subparts are matched successfully . The subparts with the points of same color in right of Figure 19 and 20 are matched. The inspection time varies between 4 and 4.5 seconds.

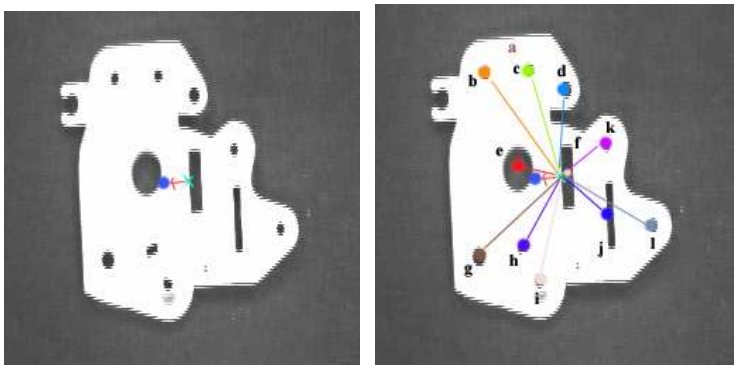


Figure 19 (Left) The reference vector. (Right) Euclidean-invariant part model.

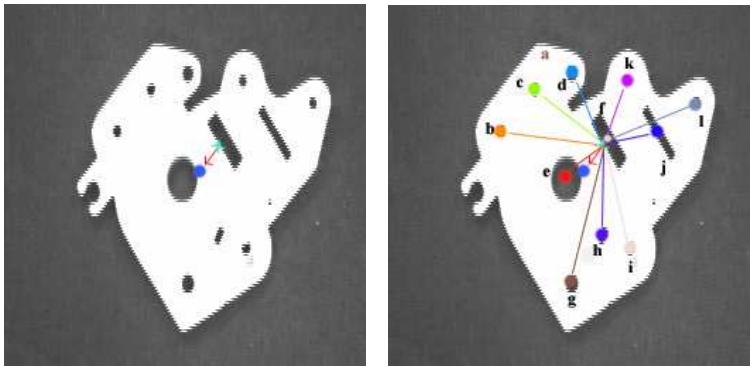


Figure 20 Construction of Euclidean-invariant part model for the to-be-inspected part.

3. Summary

In this project, we present a novel approach to automated vísual inspection of metal parts - based on selective vision - that proves to be both real-time and reliable and develop BUVIS - an inspection system based on TMS320C31. In this system, a key integral feature is that the image is selectively processed and the salient contour segments are implicitly identified as a sequence of fixation points. This not only reduces the required computation drastically, but also enables merging of the segments and interpolation of the contour in a natural manner. Once the complete contours of the subparts are estimated, their shape parameters - elliptic Fourier descriptors in our case - are easily computed. In order for the system to be insensitive to part position and orientation during the inspection, we then use an Euclidean-invariant graph representation of the part and its subparts. In the learning mode, the system records the model thus constructed. In the inspection mode, the model of the inspected part is constructed and compared with that recorded in the learning mode. A series of experiments have proven the feasibility of this system in industrial environments.

References

1. Çağatay Soyer, H. Işıl Bozma, Y. İstefanopulos. *A mobile Robot With a Biologically Motivated Vision System*, Proceedings of IROS'96, Osaka, JAPAN.
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. A. L. Abbott. *A survey of selective fixation control for machine vision.* IEEE Control Systems, 1992.
4. Hülya Yalçın, H. Işıl Bozma. *Detecting Circular Objects*. Bsc. Thesis, June 1996.
5. Carl-Fredrik Westin. *Attention Control for Robot Vision*, Proceedings of CVPR'96, pp 726.
6. Smart Eye I. User's Manual: *Complete Image Processing System on a Board.* Revision C, Visionex Inc. , February 1995.
7. TMS320C3x User's Guide. Digital Signal Processing Solutions. Texas Instruments, 1997.
8. M.R. Teague. *Image Analysis via the General Theory of Moments*, Opt. Soc. Am. , V.70, No.8, 1980.
9. R. Safaee-Rad. *Application of Moment and Fourier Descriptors for the Accurate Estimation of Elliptical Shape Parameters*. IEEE Robotics and Automation 1991.
10. Chun-Shin Lin, Chia-Lin Hwang. *New Forms of Shape Invariants from Elliptic Fourier Descriptors*, Pattern Recognition. Vol. 20, No. 5, pp. 535-545, 1987.
11. Frank P. Kuhl. *Elliptic Fourier Features of a Closed Contour*, Computer Graphics and Image Processing 18, 236-258 (1982).
12. P. Gouras. *Oculomotor System*. In Principles of Neural Science, editors J.H. Schwartz and E.R.Kandel. Elsevier, 1988.
13. Kelly, J.P. *Anatomy of central visual pathways*. In Principles of Neural Science, editors J.H. Schwartz and E.R.Kandel. Elsevier, 1988.
14. Ballard, D.H. and C.M. Brown. Principles of Animate Vision. CVIP: Image Understanding, 56(1), July 1992.
15. Ballard, D.H. Animate Vision. Artificial Intelligence, 48:57-86, 1991.
16. Newman, T.S. and A. Jain. A Survey of Automated VÝsual Inspection. Computer Vision and Image Understanding. Vol 61, No.2, March pp:231-262,1995.
17. 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.
18. Kennedy, C.W. and E.G. Hoffman and S.D. Bond. Inspection and Gaging. Industrial Press, New York, 1987.

Appendix A Aspect Ratio Compensation

Aspect Ratio Compensation

In order to construct a correct shape model of the part along with its subparts, a crucial practical issue was to compensate for the distortion caused by the 4:3 aspect ratio of the imagíng system. Consider the ellipse oriented at 90 degrees in Figure 21. With an aspect ratio of 4:3, it is mapped to a circle. Now consider the same ellipse - but this time oriented at 180 degrees. It is mapped to a squeezed ellipse and seen to be of a different shape as shown in Figure 22. As our parts are randomly oriented, shape distortion due to unequal aspect ratio must be compensated for. Our solution to this problem is the adaptation of the computation of the EFD coefficients via manipulating equations 1.1-1.4 to account for the aspect ratio.

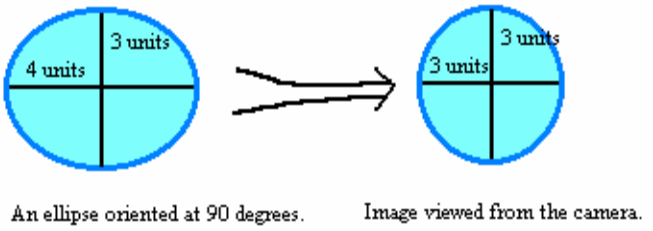


Figure 21 . (Left) An ellipse with major to minor axis ratio of 4/3, (right) the ellipse is mapped like circle to videobuffer due to the aspect ratio problem.

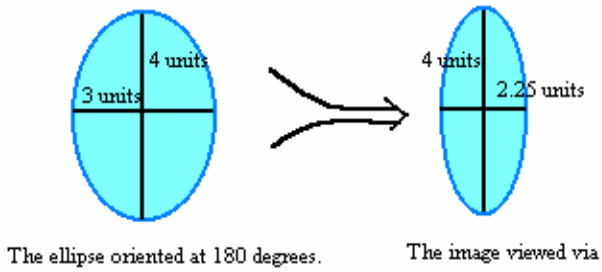


Figure 22 (Left) An ellipse with major to minor axis ratio of 4/3, (Right) the ellipse is mapped thinner to videobuffer due to the aspect ratio problem.

Appendix B TMS320 Implementations

B.1 DSP Card Components

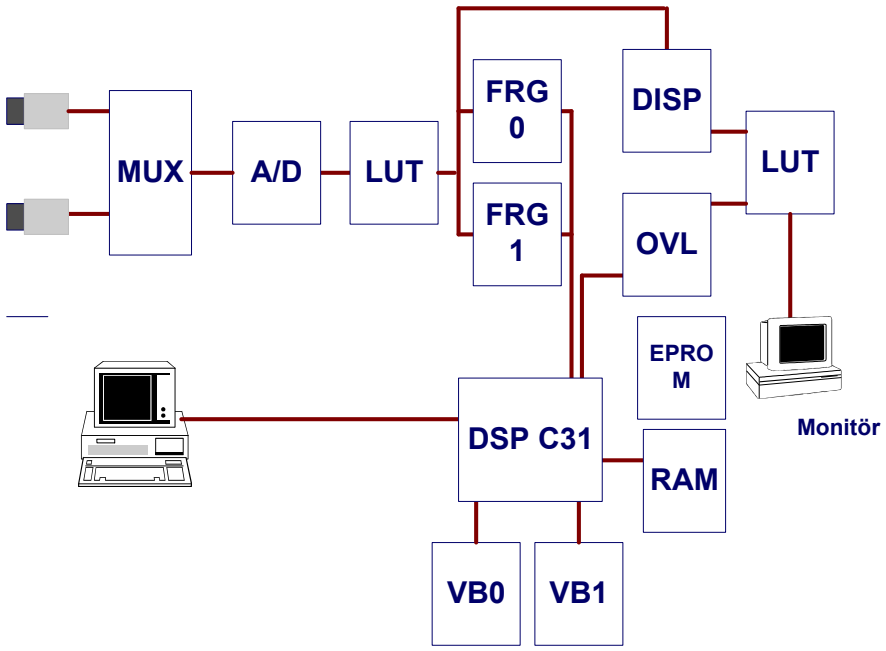


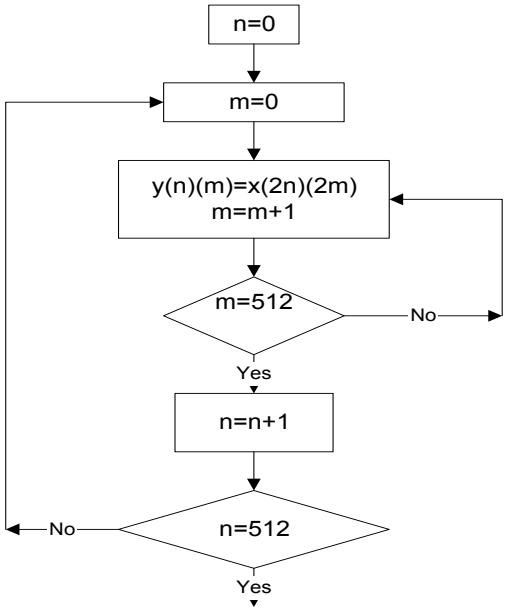
Figure 23. Smarteye card components.

B.2 Sub-Sampling of the Image

The whole 512x512 pixel data is saved to the framegrabber - a read-only part of the memory. We then subsample the image with are rate of 2:1 and store the resulting 256x256 image on the upper ¼ part of the video buffer in order to reduce buffer space. Mathematically, the sub-sampled two dimentional image(y(n)(m)) as a function of the original image can be expressed as follows:

$$y(n)(m) = x(2n)(2m)$$

where $n,m \in \{0,\ldots,255\}$. Here, x is the original image data on video grabber memory and y is the sub-sampled data on first ¼ of video buffer. The flow diagram is as follows:



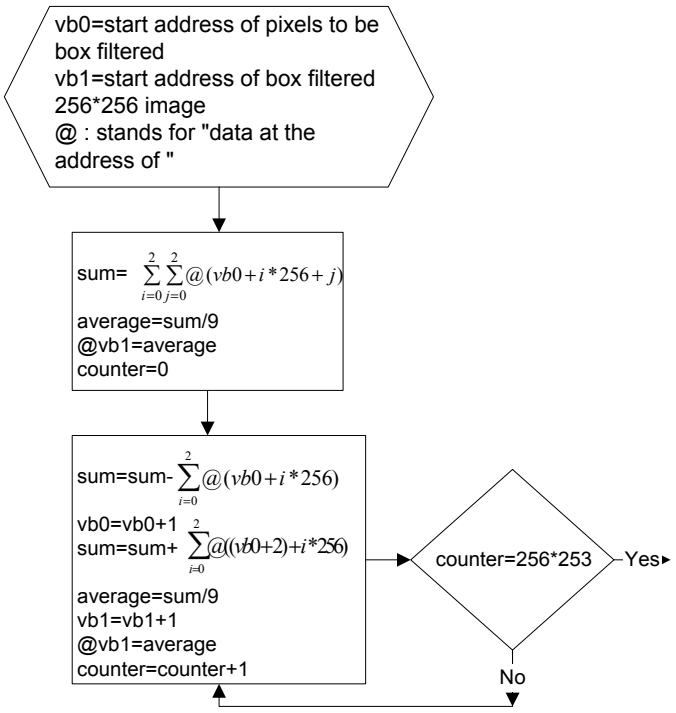
The assembly language code of the compression task is given below:

```
ldp AAvb0
ldi @AAvb0,ar7      ; first ¼ video buffer address
ldi @Aadigfr0,ar0   ;512*512 image start address
ldi 512,r5
ldi   255,rc
rptb   lp1
ldi     *ar0++(2),r4
```

```
sti     r4,*ar7++
addi   512,ar0
subi   1,r5
cmpi  0,r5
 bnz    lp2
```

B.3 Noise Removal

We eliminate the noise by a process called “box filtering “. Box filtering is simply taking the average of the data on each 3*3 pixel boxes on the whole image.The assembly code is as follows:




```
ldp   AAvb0
ldi   @AAvb0,ar7
addi  @kolon2,ar7          ;     ar7=
second ¼ video buffer address
ldi   @AAvb0,ar0      ;   ar0=  first ¼
video buffer address
ldi   @loopcount1,rc   ;
ldi     ar0,ar1
addi  256,ar1
```

```
ldi     ar0,ar2

addi  512,ar2

ldi    -16,r3

ldi    7282,r2

ldi       *ar0++,r0     ; preparation for
main filtering loop

addi  *ar1++,r0

addi  *ar2++,r0

addi  *ar0++,r0

addi  *ar1++,r0

addi  *ar2++,r0

addi  *ar0++,r0

addi  *ar1++,r0

addi  *ar2++,r0

mpyi  3 r2,r0,r1

lsh     r3,r1

sti      r1,*ar7++    ;  r0=r0/9

rptb   loop1              ;   main filtering
loop starts here

subi   *-ar0(3),r0

subi   *-ar1(3),r0

subi   *-ar2(3),r0

addi   *ar0++,r0

addi   *ar1++,r0

addi   *ar2++,r0

mpyi3  r2,r0,r1

lsh       r3,r1

sti r1,*ar7++
```

B.4 Finding Edges: High Frequency Filtering

Differentiation is achieved via discrete convolution of the image with a 3*3 template representing a high-pass filter. If a point $a_{0,0}$ and its neighborhood is represented as follows:

$a_{-1,-1}$ $a_{0,-1}$ $a_{1,-1}$

$a_{-1,0}$ $a_{0,0}$ $a_{1,0}$

$a_{-1,1}$ $a_{0,1}$ $a_{1,1}$

Using Sobel operator, the vertical and horizantal gradients can be computed as follows?

Vertical gradient $S1=(a_{-1,-1}-a_{-1,1})+2(a_{0,-1}-a_{0,1})+(a_{1,-1}-a_{1,1})$

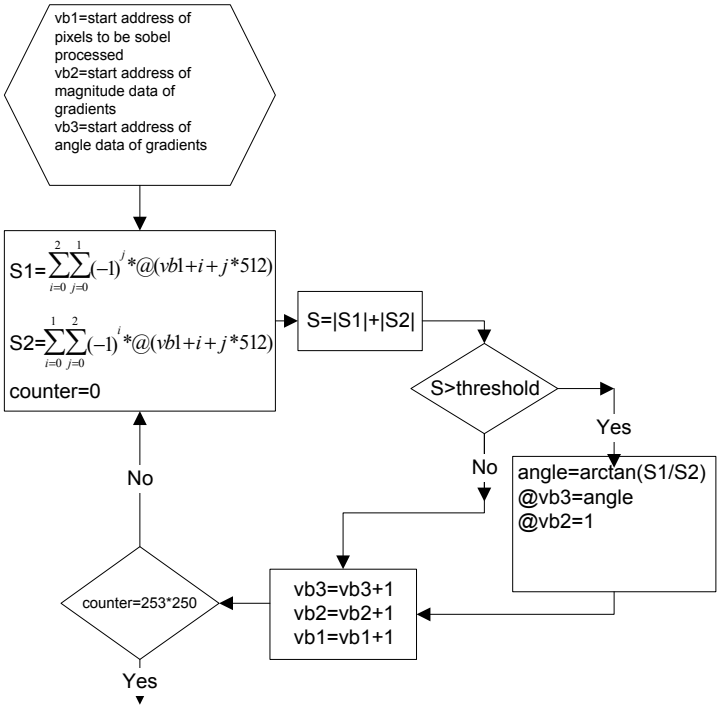
Horizontal gradient $S2=(a_{-1,-1}-a_{1,-1})+2(a_{-1,0}-a_{1,0})+(a_{-1,1}-a_{1,1})$

The magnitude of the gradient is S = |S1| + |S2|.

S1 stands for the vertical gradient of the 3*3 window of the image data, whereas S2 stands for horizontal gradient of the 3*3 window. |S| is the ultimate amplitude of the directional gradient that is the sum of vertical and horizontal gradients. Total gradient is compared to a threshold number. The ones greater than threshold value are taken to be edge pixels. Edge pixel data is written on the third 1/4 portion of the video buffer memory. A one loop of assembly code of edge identification given below :

```
        ldp    AAvb0
        ldi    @AAvb0,ar7
        ldi    ar7,ar0
        addi   @kolon2,ar7      ; ar7 apoints to second
¼ of video buffer
        addi   @kolon3,ar0      ; ar0 apoints to third ¼
of video buffer
        ldi    @loopcount1,rc
        ldi    2,r5
        ldi    1,r4             ; if a pixel is a n edge point,
it is given a label  1 in the third ¼ of the video
buffer
        ldi    0,r6             ; otherwise 0 is given as a
label
        ldi    ar7,ar4
        addi   514,ar4     ;  ar4→ a1,1
        ldi    ar7,ar1
        addi   256,ar1     ;  ar1→ a0,-1
        ldi    ar7,ar3
        addi   512,ar3     ;  ar4→ a1,-1
        ldi    ar7,ar5
```

```
        addi   258,ar5      ;  ar4→ a0,0

        rptb     exit       ; this label points to the end of
the arctan section

        subi3   *-ar4,*+ar7,r0

        subi3   *ar4++,*ar7++,r2

       ||mpyi   r5,r0

        subi3   *ar5++,*ar1++,r1

        subi3   *ar3++,*+ar7,r3

       ||mpyi   r5,r1

        addi    r2,r0        ; r0 = (a-1,1-a1,1)+2(a-1,0-a1,0)

        addi    r3,r0        ; r0 = S1

        addi    r2,r1        ; r1 = (a-1,-1-a1,1)+2(a1,-1-a0,1)

        subi    r3,r1        ; r1 =  S2

        ldi     r0,r3

        ldi     r1,r2

        absi    r3           ; r3=|S1|

        absi    r2           ; r2=|S2|

        addi    r3,r2        ;  r2 =S= |S1|+|S2|

        ldp     _thresh

        cmpi  @_thresh,r2

        sti       r6,*ar0++      ; if threshold criteria is
satisfied label 1 is given to that pixel

        blt     exit           ;  otherwise 0 is given and

        sti     r4,*-ar0       ;  exits goes to end of loop
```



B.5 Computing Edge Angles

The angle of the gradient vector is found from the simple formula given below :

angle = arctan(S1/S2)

Inverse tangent function is approximated by the first three components of the Taylor series expansion of the function. The resulting angle is then converted to degree equivalent in the $\{0,360\}$ range which are then scaled down to 0-255 range due to the fact that the memory buffer is 8-bit. The assembly code is given below:

```
ldi -1,r2
 mpyi r2,r1
 mpyi r2,r0
 ldi r0,r7
 ldi r1,r6
 ldf 40.5845,r3
 ldf 6.760874,r4
 ldf 0.3381,r5
 ldi r7,ar7
 ldi r6,ar6
```

```
        ldi 0,r1
        cmpi 0,r7
        bp  brnch1
        bn  brnch2
        cmpi 0,r6
        ldip  191,r1
        bp  out
        ldin 63,r1
        bn out
        bz out
brnch2   cmpi 0,r6
        ldiz 0,r1
        bz out
brnch1   cmpi 0,r6
        ldiz 127,r1
        bz out

        absi ar7
        absi ar6
        cmpi ar7,ar6  ; r6-r7
        ldihi r6,r0
        ldihi r7,ar1
        ldihi 63,ar5
        bhi brnch4
        ldieq 31,r1
        beq brnch7
        ldi r7,r0
        ldi r6,ar1
        ldi 0,ar5

brnch4
        push r7

        float r0
        ldf r0,r7
```

```
        pushf r7
        pop r7
        call invf
        pop r7
        ldi ar1,r1
        float r1
        mpyf r1,r0
        absf r0
        mpyf3 r0,r0,r1
        mpyf r5,r1
        subf r4,r1
        mpyf r0,r1
        mpyf r0,r1
        addf r3,r1
        mpyf r0,r1
        fix r1              ; angle in degrees
        cmpi ar7,ar6
        ldihi -1,r0
        ldilo 1,r0
        mpyi r0,r1
        addi ar5,r1

brnch7  ldi r1,r2
        ldi 0,r1
        cmpi 0,r6
        bn brnch5
        cmpi 0,r7
        ldip 127,r1
        addi r2,r1
        cmpi 0,r7
        bp out
        ldin 255,r1
        subi r2,r1
        bu out
brnch5  cmpi 0,r7
```

```
        ldip 127,r1
        subi r2,r1
        absi r1
out     nop
        ldp kolon44
        addi @kolon44,ar0
        addi 64,r1
        cmpi 255,r1
        blo store
        subi 255,r1
store   sti r1,*ar0
        subi @kolon44,ar0
```

Appendix C Glossary

aspect ratio: The horizontal/vertical ratio mapping from image coordinates to world coordinates.

event-driven approach: An approach where the visual processing is driven by the data coming from the image.

fixation point : The image point at which line of sight intersects the image plane.

merging: Grouping of segments that are part of the same contour into one entity.

periphery region: Low acuity region of visual field.

saccade direction: The direction of movement of fixation point.

segmentation: Decomposition of the image into parts that are meaningful with respect to a particular application.

selective fixation control: The process of identifying and selecting fixation points.

shape invariants: Shape descriptors invariant to Euclidean transformations.