AN ADAPTIVE PIPELINE PROCESSOR FOR REAL-TIME IMAGE PROCESSING

Neil Storey and Richard C. Staunton

Department of Engineering
University of Warwick, UK
Published: Proc SPIE Vol:1197 pp238-246, 1989

ABSTRACT

The need for low-cost, real-time, image processing for automated inspection and other industrial applications has led to considerable effort being directed at many novel computer architectures. Two main avenues of work have become apparent: the use of powerful parallel architectures and the use of pipelined processors. The former while offering great flexibility, have, to date, been associated with high cost. The latter, being essentially simpler processors can produce considerable cost savings but have generally been inflexible and provided a limited number of functions. Pipeline processors have generally found application as pre-processors for use with parallel machines or in image enhancement for human operators where relatively simple processing is required.

The paper describes the development of an adaptive pipeline processor which offers the potential cost efficiency of large scale integration together with the flexibility of an adaptive system. The processor concerned accepts digitised video information from any standard video source and performs real-time processing on it. Each identical processor element (PE) can be programmed by a separate controller over a high speed communications channel, and can perform a range of operations including filtering, averaging, edge-detection, line-thinning, thresholding, scaling and inversion. Synchronisation information is also processed by the PE allowing any number of elements to be cascaded without the need for a frame store.

A typical image processing arrangement would consist of a series of identical PEs connected in series and a separate control computer which is responsible for configuring the system. The absence of frame buffering within the system greatly reduces its cost and provides true real-time operation. Since each PE may be reconfigured in real-time the parameters and functions of each element can be modified to suit the image. This permits the processing to be adapted to cater for changing lighting conditions or a change of scene. Since each PE will be implemented as a single, identical, VLSI circuit, the arrangement is potentially very cost effective.

1. INTRODUCTION

To be used for industrial inspection tasks, machine vision systems must be reliable, cost effective and fast enough to process one object before the next is presented. The speed requirement obviously varies from one application to another but in many cases the speed of processing is a major constraint. In many applications a small amount of latency in the processing system is acceptable but a high update rate is essential. The ultimate goal of this pipeline system is to process data at the full video rate, this will be fast enough for most production line applications and all applications where a processed image is used by a human operator.

Over the last few years many pipeline processing systems have been developed. The majority of these systems use a number of frame stores and a single or small number of pipeline processor elements. The image data are cycled repeatedly through the short pipe of processor elements to perform complex operations. Many image processing applications require a great many separate passes through the pipeline (sometimes more than 100) and as each pass through the pipeline processor usually takes one frame time, the processing produced is slower than for systems without frame buffering. The inclusion of a number of frame buffers within the system allows some processing operations which would not otherwise be possible, but also increases the cost of the system.

This paper discusses the advantages of using a pipeline of processor elements (PEs) without frame buffering, one PE being provided for each processing function. Figure 1 shows the architecture of such a pipe.

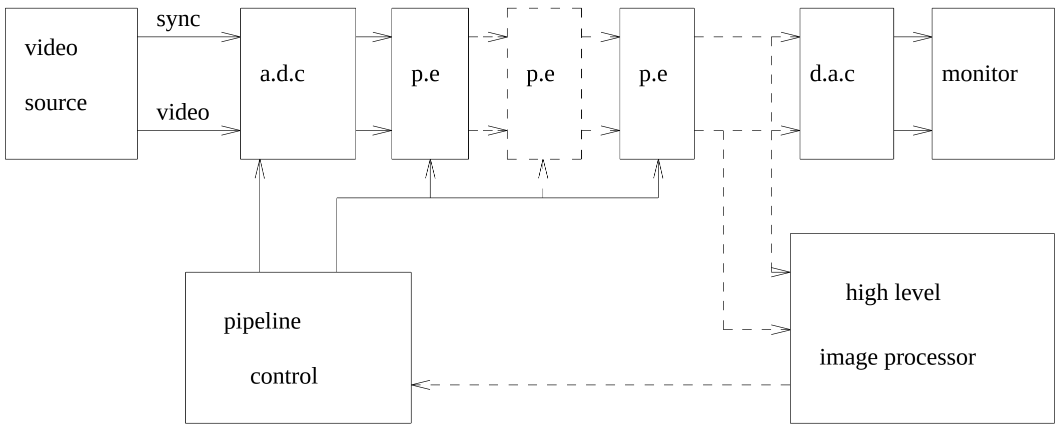

Figure 1. The pipelined image processor

The raster scanned analogue image data from the video source are digitised by the analogue to digital converter (a.d.c.) and together with the suitably buffered synchronisation signal (sync), is fed into the first P.E. in the pipeline. Within the pipeline, each PE stores and processes only a few TV lines of data before results begin to appear at its output, and thus several hundred PE stages may be cascaded before a delay of even one image frame time is accumulated. Pipeline output is either to a TV monitor via a digital to analogue converter (d.a.c.) or to a high level image processing computer. If a high level processor is used, the pipeline can be considered to be a pre-processor that performs low level functions and effectively reduces the data bandwidth for the high level system. The pipeline control unit programmes the individual PEs and provides continuous communication and synchronisation between them. The control unit can work autonomously or may collaborate with the high level processor.

To make this approach cost effective the cost of the individual PEs must be low. The use of VLSI offers the possibility of integrating the complete PE into a single chip which can be directly connected to other elements with a minimum of interfacing components.

Typical image processing applications will require a combination of many different types of processing operation. If each operation where performed by a single function PE many types of PE would be required and the development and manufacturing costs would be high. Such a system would also be inflexible as changes in the processing sequence would require a change to the physical layout of the system, or the use of complex switching hardware. A more attractive alternative is the use of programmable PEs which can be configured to perform one of a number of processing operations. This gives flexibility by allowing the system to be reconfigured for different applications and is also cost effective as it results in a single design which can be produced in higher volume.

This paper describes the design of a PE architecture, capable of performing any one of a set of functions. At present this set includes edge detection, filtering, line thinning, thresholding and inversion. The PE has been modelled in ‘C’ and complete pipes of several PEs have been simulated. The initial results, presented below, suggest that a useful PE design will emerge, and further studies indicate that it could readily be integrated into a single VLSI chip. This single chip implementation will be inexpensive, when mass produced, and will permit general-purpose, multi-element pipeline systems to be produced and programmed to suit individual applications.

The flexible nature of the pipeline will allow for dynamic re-programming of the individual PEs to allow for environmental changes such as variations in the ambient light level or direction of illumination. The pipeline simulation has been tested using a number of realistic, industrial images including those of an industrial sand core used in metal casting. These tests have demonstrated successful surface defect detection with the set of PE operators implemented. Results from some of these tests are presented below.

2. THE PIPELINE ARCHITECTURE

A typical industrial vision system might use a raster-scan TV camera and a converter to produce a 512 x 512 pixel picture, 50 or 60 times a second. Ideally, the image processing system should be able to process one frame of a picture before the next frame is available, giving maximum system throughput. Throughout this paper such a system is referred to as a 'real-time' system.

Throughout the world, the need for real-time image processing has led to considerable research effort being applied to the design of high-speed image processing architectures. Much of this effort has been directed at the use of two or multi-dimensional arrays of interconnected processors. However, the cost and complexity of 2D arrays of PEs for industrial systems are high. Many PEs are required for real-time processing, and large random access memories are needed at each PE to store image and programme data. Additionally, these systems must employ a frame store and data reformating logic, which introduces a delay of one video frame time, unless the problems of optical image input and output to each PE can be solved[1].

Many of the operations required during image processing do not require a knowledge of the complete frame of an image, but only of a group of adjacent pixels. If these operations are performed on a pipeline processor there is no need to store a complete image. Such a processor may operate directly on the serial data stream from the digitised output of a raster-scanned device. The PEs must operate in real-time, but since the operations are very simple only a few lines of the image need be stored in line length shift registers within each PE. Figure 2 shows a pipeline processing element which stores two lines of the video image.

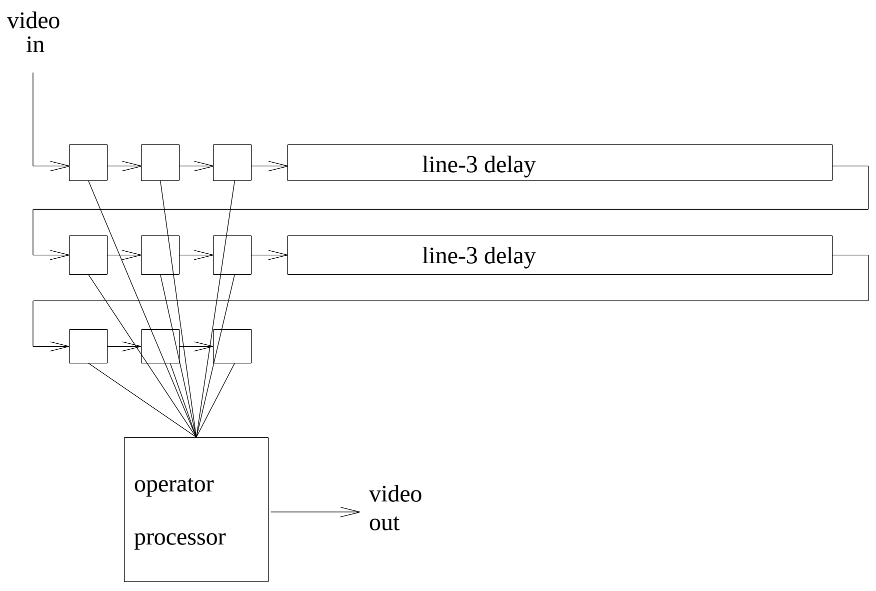

Figure 2. A pipeline processor element (PE)

Three bits of each of the two line storage registers, together with three bits from the previous line, form a 3x3 local image area on which processing is performed. Processing operations are performed on this array of elements and a new value for the center pixel is calculated and output, forming the output video stream. The overall system is much simpler and less expensive, than the parallel approach. Storing more lines of an image enables larger local areas to be used. For example a 5x5 pixel area could be used by storing four video lines. Processes of increased complexity can be achieved by cascading a number of PEs in a string.

Many pipelined image processing machines containing PE architectures based on that of figure 2 have been reported in the literature. Few are limited to contain just this single PE design but also have general purpose ALU and lookup table elements. Some of these systems are now briefly reviewed.

The Cytocomputer developed by McCubbrey and Lougheed[2] has a PE operator processor that is capable of performing arithmetic, Boolean and morphological operations. They have reported its use for morphological image analysis[3]. The pipeline contains up to three PEs and is used in conjunction with a frame store and a host microcomputer.

The PIPE system developed by Luck[4] is usually configured with from three to eight PEs. The direction of data flow within the pipeline is controlled by switches. Each PE can perform Boolean or arithmetic neighbourhood operators. Luck divides the vision task into three parts; Low level vision, Mid level vision and High level vision. Low level vision includes operations such as edge detection, mid level operations such as the Hough transform, and high level operations such as feature detection. He identifies low level vision as being a suitable task for pipeline processors and suggests that high level processes should be performed by a separate mini-computer or array processor.

Batchelor and Bowman et al[5] review developments in image processing for industrial inspection and again identify a division into low and high level vision tasks. They also state the necessity for control of the low level tasks by the high level processor. Bowman[6] has developed a system called Kiwivision. The standard system has a pipeline of three PEs, each performing a different set of operations. Input/output modules and two frame stores are employed. The modules are linked by five sixteen bit busses together with a VME bus which also communicates with a 68000 based computer. This computer provides system control and high level processing.

Valkenburg and Bowman[7] describe a new system, Kiwivision II, which consists of a pipeline section for low level operations and an array architecture section for high level operations. The array is comprised of Inmos transputers, and the pipeline section of commercially available pipeline module boards. They intend to interface the PEs to the standard transputer links to enable high level control.

The systems described above are characterised by having only short pipelines of PEs and by individual PEs in the line being of a different hardware construction. Frame stores are used to enable data to be re-circulated through the pipe and many video and system busses are employed for data communication and system control. Systems are comprised of many separate function boards. These systems are however capable of performing a wide variety of complicated image processes, some of which can be classified as being in the mid level vision range. For example the convex hull process has been performed by Bowman[8] using a pipeline processor. First data is scanned horizontally out of one frame store, processed and re-stored and then scanned vertically out of the second frame store and processed. Various single chip filters based on the architecture also exist such as the Inmos A110. This is a programmable device capable of a range of operations. However it is a general purpose device and is not able, as a single chip, to perform such operations as the Sobel edge detector.

We are aiming to produce a single PE design that can be programmed by a control computer to perform operations from a usefully large set of image processing operators and to integrate the PE on a single VLSI chip. At present we have simulated a basic design of PE and a short pipeline system. The design is described below and the simulation in Section 4.

To obtain the single PE design, some operator types will have to be excluded and the low/high level boundary, that determines which operations are implemented by the pipeline and which by the high level computer, shifted downwards.

The PE architecture is based on the model of figure 2, which precludes some operators such as median filters, although, at the price of extra silicon area, these could be provided and switched in if required. On the positive side, advantages will include low cost, the ability to connect large numbers of PEs in single board pipeline systems and a totally flexible pipe of identical PEs that can be programmed to perform a still large number of operations in any order that is required. This flexibility makes possible adaptive control of the pipeline. A control computer monitoring the pipeline output and other parameters, if programmed with a suitable algorithm, will be able to dynamically change parameters in the individual PEs or even change the operations being performed by each PE. If more PEs are present in the pipeline than are required, they will be programmed for null operation.

A further restriction imposed by a VLSI implementation is the number of edge pins available with standard chip packaging. To minimise production costs it is advisable to restrict the package to 40 or fewer pins. This restriction imposes some difficulties in implementing the re-circulation type system architecture, which requires several video and control busses, but other system advantages emerge to compensate for this. Our design limits the video input and output ports to 8 bits each. Video synchronisation is passed through each PE and a system clock signal is required. PE programming and control is via a serial link. In total, the PE could be packaged in a 28 pin d.i.p. or if two video input ports are allowed, a 36 pin d.i.p. Several PEs could be included within one of these packages. Long pipelines could be implemented on a single circuit board and the close proximity of the PEs will enable a high bandwidth serial control capability.

The types of pipeline architecture that can be implemented with these PEs are:-

(a) The standard system of figure 1. This has been simulated and results are reported in Section 5.

(b) A re-circulating pipeline that has a frame store at each end and a video bus connection between frame stores or a return pipeline.

(c) A multi-pipe system. If the delay in each PE remains constant, irrespective of the operation being performed, branching and merging of the pipeline is possible. For example figure 3 shows a system overlaying thinned edges on top of the original image.

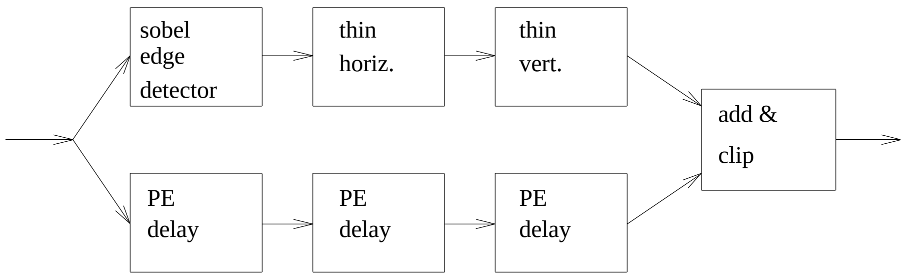

Figure 3. A multi-pipeline arrangement to produce an edge overlay on an image.

Figure 4 shows the elements of the PE. Pipeline synchronisation signals derived from the TV synchronisation signals, will be piped along with the image data to ensure image frame and line synchronisation as each PE delays the image data. A two line delay and a 3x3 local area is stored as in the basic PE model described earlier. The nine pixels of this local area are piped to two 3x3 multiplier arrays. Here the image data is multiplied by values in two separate and pre-stored 3x3 arrays. The elements of the resulting 3x3 arrays are summed and the resulting two values combined appropriately in the box labelled ‘programmable operator’. A series of other operators then process the data. It is possible to set all of the process boxes to null operations. The input and output video ports are 8 bit wide, although many of the interconnection paths have greater widths.

With our simulation we have implemented averaging, filtering, edge-detection, line-thinning, scaling, point enhancement, thresholding and inversion. The filter is a Gaussian approximation filter[9]. With this operation it is required to convolve the 3x3 local image area with a 3x3 constant array and to rescale the result. The 3x3 constant array is pre-loaded via the control input into one multiplier and a null array is loaded into the second. The first summer outputs a result and the second a null. The programmable operator passes just the output from the the first summer and the programmable scaler divides this by a pre-programmed constant value. The other process boxes are set to null. If a result greater than 8 bits (255) is going to result, the output will be limited to 255. Likewise limits are introduced to prevent negative output values.

The Sobel edge detector has been implemented and is an example of an operator that uses most of the process boxes within the processor. Horizontal and vertical edge detection masks are pre-loaded into the two multiplier arrays and convolved with the image data. The outputs are summed to produce horizontal and vertical vectors. The programmable operator squares these and then performs a square root approximation. This resultant vector magnitude can then be rescaled and compared with a pre-programmed threshold value. The thresholders binary output can then be inverted if required. Our final design will also incorporate two video input ports and an ALU to combine two input signals. A lookup table on the video output may be added.

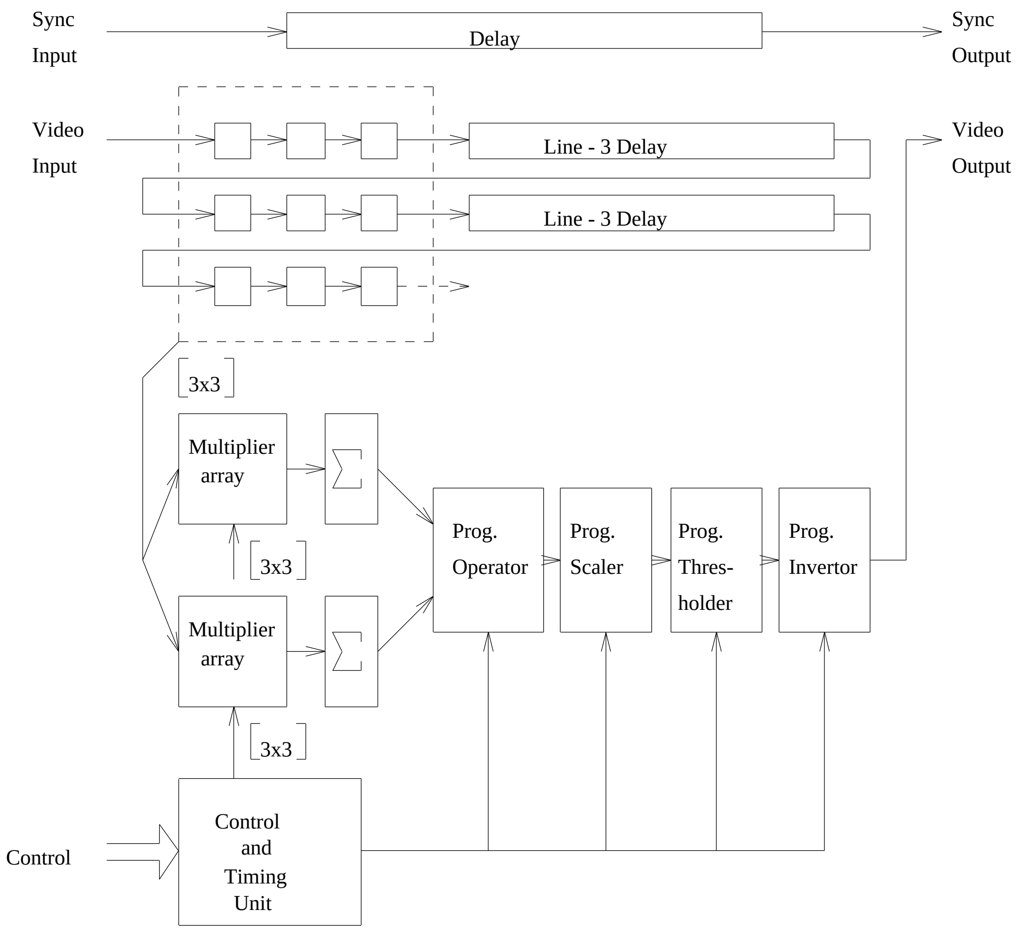

Figure 4. The structure of the pipeline processor

3. IMAGE SAMPLING

Before processing, a video signal must be digitised to produce a numerical representation of the image. Conventional image processing systems usually sample the signal to form a ‘square’ array of samples with equal vertical and horizontal spacing of the pixels. An alternative approach is to choose a regular hexagonal sampling technique. This has advantages with certain processes. A comparison between square and hexagonal sampling methods is presented in another paper at this conference, and will not be considered further here. The pipeline described in this paper can be designed to operate with either square or hexagonal image sampling, or with both. The simulations performed were carried out using both sampling methods.

4. PIPELINE SIMULATION

Preliminary simulations of the pipeline processor were performed in ‘C’. The purposes of these simulations were to investigate the characteristics of the individual PEs and to enable sampling techniques and local operators to be studied.

The simulation did not attempt to incorporate all the characteristics of the final design. Functions not simulated include the transmission of synchronisation signals within the PE and the ability to dynamically re-programme it.

The simulations were performed on a conventional workstation running UNIX, and the UNIX ‘pipe’ facility was used to construct multi-element pipelines to permit complex processing operations. Initial test data were provided using computer generated images, but later data from real images were used.

The results of the simulation confirmed the correctness of the algorithms used and provided useful data concerning edge effects. A more detailed simulation is now underway using the hardware description language HILO to provide more insight into the architectural requirements of the system.

It should be noted that the methods used for the preliminary simulations were very slow in execution. However the flexibility provided by this approach more than compensated for its lack of speed.

5. A SAMPLE APPLICATION - SURFACE INSPECTION

At an early stage in the pipeline and PE design, it was thought necessary to identify the types of operator that are necessary for general industrial image processing. An appraisal of the ease of implementation of these techniques in a single PE design was also required. A series of industrial applications was investigated, and some of the results from one such study are presented below. This application concerned the automatic identification of surface defects from images of a sand core. The core is used in the production of manifolds for I.C. engines.

Figure 5 shows a square sampled 256x256 image exhibiting gross defects such as a missing ‘finger’ and more subtle surface defects. Three surface defects are situated in the second ‘finger’ pair from the left. The object is not illuminated optimally, but the three defects, which are caused by scratches in the surface, are clearly visible in both images. There is one large circular defect, a long thin defect, and a small defect with dimensions comparable with the pixel size. The image was generated from an original 512x512 image captured by a frame grabber with a 4:3 image aspect ratio, and so Figure 4 is not a perfect square sampled image. Many of the image operators used are therefore not optimum.

Figure 5. The original image

Figure 6. Image after edge detection and thinning

Figure 6 is a thinned, edge image of the earlier image. This was formed by applying a Sobel edge detector, followed by magnitude comparison with a fixed global threshold value. This procedure is outlined by Gonzalez[10]. The image was then thinned by applying one horizontal and one vertical thinner. The detection is accomplished in one PE, and the thinning by two. The object outline has been calculated correctly and appears to be continuous. The defects have also been outlined. The images show the presence of small amounts of noise. The addition of a low pass filter as the first PE in the pipe will remove this noise. A 3x3 square Gaussian approximation filter as proposed by Davies[9] is suitable and easily realisable with the common PE architecture. A figure showing the effects of this filtering is not included for reasons of space.

When implemented using the proposed pipeline architecture the edge-detection, thinning and filtering operations described above require four PEs. Using a pipeline processor with repeated frame buffering these operations would typically take several frame periods (perhaps up to 80 ms), and could not produce continuous video output. The pipeline architecture proposed within this paper would generate continuous video output data with a latency of approximately 8 line periods (a delay of about 260 μs). For more complex processing operations the speed improvement would be equally dramatic.

6. CONCLUSIONS

The sample application illustrated above indicates that the proposed pipeline processor can perform many of the operations required for typical industrial projects. It also shows that these operations can be performed significantly faster using this approach than using pipeline processors which use repeated frame buffering.

The surface inspection results presented show that defect outlining is possible by the application of a series of simple, local, image processing operators. Higher level processes may then be used to separate defects from objects and normal surface features. Image processing projects must be partitioned to define tasks to be performed by the pipelined pre-processor, and those to be performed by a high level, non-pipelined machine. The absence of frame buffering requires that some tasks be performed in the high level machine which could be performed by a pre-processor equipped with such buffering. However, few operations come into this category and the speed advantage of the non-buffered pre-processor may compensate for the extra high-level processing required in these cases.

The present pipeline simulation is very slow, a 512x512 image processed by a moderate length pipeline taking about twenty minutes. The next stage in the development work will involve a complete hardware simulation of the PE using a hardware description language. This will be followed by the design of a full custom VLSI circuit. Estimates of chip area indicate that the present PE design could be integrated on a 25mm^2 die using a 2μm fabrication process.

7. REFERENCES

1. T.Fountain, *Processor Arrays Architecture and Applications,* Academic Press, London, 1987.
2. M.Lougheed, "A High Speed Recirculating Neighbourhood Processing Architecture," *Proc. SPIE,* Vol.534, pp.22-33, 1985.
3. D.L.McCubbrey and M.Lougheed, "Morphological Image Analysis Using a Raster Pipeline Processor," *IEEE Workshop on Computer architectures for pattern analysis and image database management*, 1985.
4. R.L.Luck, "Using PIPE for Inspection Applications," *Proc. SPIE,* Vol.730, pp.12-19, 1986.
5. B.G.Batchelor and C.C.Bowman et al, "Developments in Image Processing for Industrial Inspection," *Proc. SPIE,* Vol.730, pp.34-46, 1986.
6. C.C.Bowman and B.G.Batchelor "Kiwivision a High Speed Architecture for Machine vision," *Proc. SPIE,* Vol.849, pp.42-51, 1987.
7. R.J.Valkenburg and C.C.Bowman, "Kiwivision II a Hybrid Pipelined Multitransputer Architecture for Machine Vision," *Proc. SPIE,* Vol.1004, pp.91-96, 1988.
8. C.C.Bowman, "Getting the Most From your Pipelined Processor," *Proc. SPIE,* Vol.1004, pp.202-210, 1988.
9. E.R.Davies, "Design of optimal Gaussian operators in small neighbourhoods," *Image and Vision Computing,*Vol.15, no.3, pp 199-205, Aug. 1987.
10. R.C.Gonzalez and P.Wintz, *Digital Image Processing,* Addison Wesley, MA, USA. 1977.