Mapping Algorithms on to Platforms: An Approach to Algorithm and Hardware Co-Design

P. Courtney, R. B. Yates and P.A. Ivey
Dept. of Electronic and Electrical Engineering,
University of Sheffield,
PO Box 600, Mappin St.,
Sheffield, S1 3JD.
email R.Yates@sheffield.ac.uk

Abstract

The aim of this paper is to bring together aspects of our recent work in machine vision. We discuss the relative merits of various platforms for the implementation of machine vision algorithms with particular emphasis on the DIP chip, a custom device designed following an investigation into the computational requirements of a range of image manipulation algorithms.

We will show: (i) how certain operations may be made easily available in hardware; (ii) that algorithms may be redesigned to take this into account; (iii) that high performance processors other than workstations are the natural platforms for these algorithms; and that (iv) that such platforms will be necessary to convincingly demonstrate the potential scalability of these algorithms.

1 Algorithms for Machine Vision

The range and number of image manipulation algorithms developed over the years is vast. The requirements they attempt to satisfy and the constraints they encapsulate are equally diverse. They may be categorised according to their computational and communications requirements. The classical division is into three levels: an image-based low level; a high level manipulating symbols; and an intermediate level extracting high level symbols from the low level. One kind of operation which occurs again and again at the low level is that of convolution, where a block of data and a block of coefficients are first multiplied one with the other and the products summed together. A great many filtering and feature extraction tasks fall into this category. This repetitive and regular task offers the promise of easy parallelism and such operations are often worthy of specialised hardware.

During the course of a recent project concerned with the development of an image communication open architecture (ICOA) to support future remote image-based services, the need was recognised for a high performance image communication engine to carry out compute-intensive image operations. An analysis was

carried out of the requirement of image compression (coding, decoding & transcoding), image processing and machine vision. It considered patterns of operand usage and operations carried out [1]. The analysis was restricted to the low level and ruled out intermediate level operations likely to use pointer-based access or data with irregular structure. At the time of the analysis, no single chip fulfilled the requirements listed below without extensive additional support chips for program control and data formatting. It concluded that the principal requirements were to:

- carry out computations with variable bit accuracy
- deal with variable kernel sizes
- have a high I/O bandwidth
- provide a wide range of arithmetic functions

2 Platforms for Machine Vision

2.1 Workstations

General-purpose workstations based, for instance, on the SPARC processor, provide reasonable levels of performance with a steady increase - roughly doubling every 15-18 months. Development environments such as Sheffield's own TINA tool, provide the algorithm developer with the basic facilities to design complex vision algorithms without being concerned with the underlying platform. The ease with which users may move between machines is one of their great strengths.

2.2 The DIP chip

As a result of the algorithms analysis carried out and the lack of any suitable commercial device it was decided to design a chip architecture that would fulfill the requirements identified. A device, named the DIP chip (Digital Image Processor) was therefore produced [2] [3]. It has a hybrid architecture which combines an array accelerator with a scalar processor to meet the needs of both pixel-based and non-pixel based processing. See figure 1, overleaf. The scalar processor is a 16 bit RISC while the array is based on the idea of Single Instruction Multiple Data (SIMD) processing and is made up of 256 single bit processing elements.

2.2.1 RISC

The RISC controls the array. It is a conventional RISC using a 5 stage pipeline, an instruction set of 57 instructions and thirty-two 16 bit registers. It has separate code and data memories. In addition to the standard functions, the ALU includes a barrel shifter and a min/max generator to speed up list sorting operations.

2.2.2 Array

The array has its own instruction stream and can be programmed in one of two ways: as a conventional SIMD with each processor receiving the same instruction; or in a pipeline mode where each row of processing elements receives the same instruction thus acting as a 16 stage pipeline. The array can also be partitioned for applications where a complete 16 x 16 array is inappropriate, for instance, it can perform four 8 x 8 operations such as Discrete Cosine Transforms at a time.

Each processing element of the array consists of a one bit ALU and a set of memories partitioned as A (16 bits), B (32 bits) and C (32 bits). An alternate bank of A, B and C memories is also provided, permitting simultaneous computation and communication.

Figure 1 Floorplan of DIP chip and photograph of die.

Since one of the most important functions was identified as convolution, it was decided that the processor should be able to carry this out as efficiently as possible. Performing this operation on the DIP chip consists of the following pipelined stages:

- multiply data with corresponding coefficients on the array in SIMD mode.
- accumulate bits in row of processing elements on the array in pipeline mode.
- transfer partial sums to the RISC for final summation.

The array is therefore intended to be the main compute engine and carry out repetitive operations on blocks of pixels while the RISC carries out other processing such as final accumulation which would be inefficiently performed by the array.

Earlier SIMD array chips such as CLIP and GAPP could only manage a relatively small number of processors and could not accommodate a fully programmable controller but instead resorted to a tree adder to help speed up convolution operations.

2.2.3 Instruction Issue

The main advantage of using a large array of single bit processors is the potential to utilise the silicon area more efficiently by exploiting the maximum amount of parallelism. It also permits operations to be tuned to the appropriate level of bit accuracy by applying the same instruction a number of times. However, one of the potential problems with such arrays is that of delivering instructions quickly enough. For instance an N bit add will be a loop of N single bit adds; while a multiply will be a nested loop of N add loops. This is addressed in the DIP chip by the use of special loop instructions, loop counters and an instruction cache. This provides two levels of nested looping, reducing the instruction bandwidth, eliminating the need for large cache, and simplifying the task of programming the device. This cacheing scheme proves to be sufficient to keep the array supplied with instructions for single loops of greater than 6 and double loops of more than 5. A C-like compiler called DISPL has also been produced which includes constructs for specifying these loops.

2.2.4 Input/Output

Each processing element is connected to its four neighbours North, South, East and West, as well as to four busses which connect the processing elements to the I/O channels. These channels operate via corner turn line buffers (CTLB) which convert 256 bits values from the processing elements into 16 words of 16 bits. One channel is used to interface between the array and the RISC via two CTLBs (1 each for input and output); the other three operate as DMA channels between the array and the external memory. The RISC is responsible for controlling the DMA channels which is done by the use of DMA control four 16 bit microcode words.

These multiple DMA channels, double buffered memories and the instruction cache all help to maximise the internal processing rate while keeping the off chip frequency low thus easing system integration.

2.2.5 Implementation

The device is 14.4mm by 13.5mm and consists of a total of 850,000 transistors. It has been fabricated on 1 micron CMOS in two layer metal by ES2. The array runs at 40MHz, whilst the RISC and external interfaces run at 20MHz giving a total performance of 1.2Gips (assuming 8 bit addition/subtraction) and 1Gbits/sec I/O bandwidth. Second silicon with bugs identified in the first design eliminated, is currently being processed.

The design is intended to be scalable in that the number of processors in the array can be increased without a great deal of reworking, resulting in a linear increase in performance¹. This is in contrast to conventional designs using crossbar switches and/or techniques such as scoreboarding and superscalar where a great deal of design effort must be expended to ensure a similar increase in peak performance. The scalability of a design is an important consideration and future architectures for machine vision should be analysed in regard to their balance in terms of computation, communication and control of the processing resource.

The requirements identified earlier were therefore satisfied in the following way:

- variable bit accuracy - most efficient above 5-6 bits
- high I/O bandwidth - up to 1Gbits/sec
- wide range of arithmetic functions - programmable array and controller

2.3 Compression Engines

While the DIP chip was being designed, the number of companies engaged in producing devices for the general multimedia market has increased greatly. A series of image compression standards have emerged and this has led some suppliers to concentrate on designing programmable rather than dedicated chips. These devices can be seen as an extension of DSPs specialised for 1D signal processing tasks. Two devices of particular note have emerged.

2.3.1 IIT-VP Vision Processor

The IIT-VP² is a programmable image compression engine and features a number of interconnected functional blocks for H.261, etc. It is programmable via microcode and incorporates a powerful multiply-accumulate block which suggests that it would be suitable for convolution type operations.

2.3.2 MVP

The 4 million transistor MVP³ consists of four 32 bit integer DSP cores; a RISC controller with floating point unit; a set of twenty-five 2k memory banks; and a crossbar switch to permit any-to-any connection. The DSP cores contain additional hardware for zero overhead loops to support XY processing. Their ALUs

¹ With improved process geometries, the performance will go up as a cube of the scale

² Integrated Information Technology, CA, USA

³ Multimedia Video Processor from Texas Instruments, Bedford, UK

are splittable so they can process four 8 bit operand pairs per cycle. This feature makes these devices well suited to pixel processing and yields an aggregate processing rate of about 640M multiplies/sec at 40MHz.

3 Mapping Algorithms onto Platforms

Real applications place severe requirements on machine vision in terms of robustness, reliability and throughput. Certain applications involving realtime control have hard deadlines and also demand predictability of throughput. To date, platform constraints have rarely been employed in the design of algorithms. The principal constraint is that repetitive low level operations such as multiply-accumulators are easy to provide. Low level processing operates with known low levels of bit accuracy when compared to higher level operations, allowing integer rather than floating point units to be used. Such operations use regular data structures making predictability of throughput easier to ensure. Hardware supporting these operations may be easily assembled from commercially available boards, chips or gate arrays libraries. It is therefore worthwhile examining to what extent some of the computational effort in carrying out higher level operations can be transformed into convolution type operations and in this section we consider how they can be applied to two major tasks in machine vision.

3.1 Stereo Reconstruction

PMF [4] remains one of the most well-known stereo reconstruction algorithms. This edge-based matching algorithm employs a set of constraints in part inspired by work in human psychophysics. It makes extensive use of linked list structures and has been successfully ported to a transputer platform to carry out pick and place operations in 10 seconds [5].

An alternative approach has been to apply the same constraints to a matching algorithm using modified area-correlation [6]. This method has been shown to have a similar performance to the original PMF algorithm in terms of location accuracy and outlier rejection. This algorithm however has a very regular dataflow and is much more amenable to speeding up by low level convolution subsystems. A device capable of carrying out this, and other tasks employing variable kernel sizes, has been designed as part of a PhD [6]. This is a companion device to the DIP chip and a system using both is expected to attain a throughput of 10 images per second.

3.2 Object recognition

Geometric Invariances (GIs) are a popular technique based upon measures which are invariant to image viewpoint, particularly perspective distortions [7]. A number of these, based on groups of line and curve features have been discovered for planar objects. The steps involved are: (a) compute invariances based on line/curve features; (b) use these to index into model base; and (c) verify candidates by comparison of edgel strings in a suitable coordinate frame.

Pairwise Geometric Histograms (PGHs) are representations where co-occurrences of object lines are encoded in terms of relative angle and distance in a histogram [8].

The steps involved are: (a) histogram construction - one for each line in the scene; (b) search object database - matching each histogram with every other using dot-product correlation (histograms of size 32 by 32 with 8 or fewer bits per bin have proved adequate); and (c) object location - by voting in a hough space.

These two methods are invariant to absolute translation and rotation and are candidates for the basis of practically realisable object recognition systems. They have both been shown to be robust in the presence of noise due to clutter & occlusion in the scene, and sensor error. A further aim is that of scalability to realistically large number of objects. This is particularly important for use in 3D recognition since large numbers of 2D objects, corresponding to the various characteristic views, will be needed.

In computational terms GIs seem to be preferable. They rely upon the computation of a small number of highly accurate measures and the comparison of a few candidate edgel strings for verification; whereas PGH methods require repeated correlation. However from the point of view of constructing an architecture to carry these methods out, the use of floating point operations at an early stage and the manipulation of irregular data structures is not helpful whereas matching large numbers of small low bit accuracy histograms is and this will execute rapidly on multiply-accumulate rich architectures.

Scalability is a difficult requirement to satisfy as it requires large amounts of time and effort to study. To construct the database with 64 objects currently may take several hours and to test scalability it will be necessary to run tests with 10 to 100 times more objects. In an effort to address this issue, the practical effect of trying to implement PGH on a range of platforms was investigated.

3.2.1 Mapping PGH onto a range of platforms

A study was carried out to assess the levels of performance likely to be observed for a PGH-based object recognition system implemented on a number of platforms. The matching stage was selected as the most work intensive stage since for a system of a moderate size, this likely to be far greater than the PGH construction time. In all cases typical histograms of size 32 by 32 bins were used.

The DIP chip: In mapping the PGH matching to the DIP chip the process was broken down into three main stages: the multiply and initial add on the array; the RISC I/O control and 1D accumulation; and the off-chip I/O via the DMA channels [9]. These can all be performed in parallel with the system speed being limited by the slowest stage. An examination was made of the effect of varying the bit resolution of the histogram bins. The following table shows the time taken for each stage using 16 by 16 histograms:

Bits	Array computation	RISC	I/O	Total time	Bottleneck
6 bit	(54+31=) 86	108	96	108 cycles	RISC
7 bit	(70+33=) 104	108	112	112 cycles	I/O
8 bit	(88+35=) 124	108	128	128 cycles	I/O
9 bit	(108+37=) 145	108	144	145 cycles	Array

This demonstrates that in contrast with a workstation, performance can be usefully tuned to the desired bit resolution. It also shows that the DIP chip is fairly well balanced for this operation. The cost of the multiplication stage increases with the square of the number of bits and dominates the time above 9 bits.

Below 6 bits, the time is dominated by the RISC and array instruction issue.

IIT-VP Vision Processor: The IIT-VP evaluation resulted in a 5 stage pipeline capable of matching an estimated 125,000 histograms per second [10]. This is limited by the speed of the multiply-accumulate block. The on-chip run-length coder, intended for use on JPEG/MPEG blocks, can be usefully employed to reduce the I/O and storage requirements by a factor of typically 8 since PGHs are very sparse. This increases the potential matching throughput, eliminating what would otherwise have been an I/O limited task.

The results of the evaluation are summarised in the table below for a SPARC 1 workstation⁴; the DIP chip⁵; the IIT-VP⁶; and the newer MVP⁷. Throughput is given in histograms/second (32 x 32 bins) as well as the likely throughput in terms 2D objects/second, assuming 100 histograms/scene and 32 histograms/object.

These figures are biased against the workstation in that the other figures are estimates rather than measured values and the workstation used was not a top-of-the-range model but gains in the range of 2-3 orders of magnitude seem feasible. The differences in performance of the chips is consistent with the number of transistors.

Platform used	Histograms per second	2D objects per second	Speedup factor
SPARC 1 Workstation	700	0.22	1
DIP chip (estimated)	78,125	24	110
IIT-VP (estimated)	125,000	40	175
MVP (estimated)	312,500	97	445

4 Discussion and Conclusions

The development of new algorithms has typically been carried out in isolation from knowledge about the underlying platforms. In this paper we have shown how hardware constraints may be used in the design of machine vision algorithms. If machine vision is to become commonplace algorithms will have to be mapped onto architectures which can be mass produced. As has been shown, the algorithms which are most appropriate for these components are based on convolution-type operations, not only low level algorithms, but also redesigned high level algorithms will fit on the same platform.

Most vision applications seem to require a binary decision or control output from a scene image of millions of bits and conventional computers have not so far been powerful enough to process dense pixel data at a sufficiently high rate. From this, and the classical division of vision tasks into three levels, it has been inferred that the amount of data must be monotonically reduced as we move from the input to the output and that machine vision algorithms should in effect act

⁴programmed in C using cc compiler, optimiser turned on

⁵estimated for 40MHz device using 8 bit histograms; improved figures with fewer bits

⁶estimated for 40MHz device; 80MHz device now available

⁷rough estimate for 40MHz device; 100MHz device expected

as a "data funnel". PGHs are unusual in that they do not follow this model. The hundred or so scene lines (a few kbytes) are each expanded out into 1024 byte histograms (a few 100kbytes). Such algorithms are also easy to port to various platforms since they have well defined requirements in terms of numerical precision although ease of porting has always been one of the points in favour of workstations. As we have shown, large performance improvements are available when silicon area is given over to specialised multiply-accumulators; loop counters; and DMA controllers; rather than more general purpose units such as large caches and floating point units. The indications are that performance improvements in the range of 2-3 orders of magnitude over a workstation would be possible with an add-in card. These are the kinds of speed-up necessary to carry out realistic assessments of scalability. The use of platforms allowing scenes to be analysed at around framerate (above 10Hz) may give additional advantages. It can be argued that over this timescale the world changes little - this would explain the cutoff frequency of the human visual system. Analysis of the world at this rate will be easier and more reliable and we hope to examine this question in the near future.

It has recently been suggested⁸ that the design of image-oriented hardware occupies a marginal position in machine vision research. The problem seems to centre on high development cost and short useful life of the (application specific) hardware developed. However new opportunities which can help meet the needs of machine vision are becoming available thanks to image compression with its realtime demands, and highly flexible FPGAs (Field Programmable Gate Arrays)⁹. Novel computer architectures are clearly an enabling technology for future remote image-based services, such as image indexing by content, and we would suggest that this is a valid sub-field within machine vision.

Acknowledgements

This work was funded by the EEC RACE initiative on project 2056 AMICS (Advanced Multimedia Image Communication Services). The DIP chip was designed by Steve Evans and Simon Walker and fabricated by European Silicon Structures (ES2) Ltd. Thanks also to Chris Brown, Peter Riocreux and Neil Thacker for their assistance in carrying out some of the work reported here.

References

- [1] N.A. Thacker, J.D. Morris, S.J. Evans, R.B. Yates and P.A. Ivey, (1993), An Assessment of Image Processing and Computer Vision Algorithms Suitable for VLSI Implementation, Dept E&EE-ESG Memo 93/3.

⁸eg the IMV programme

⁹eg the XC4008 from Xilinx has 400 CLBs can hold twenty-five 20MHz 4 bit multiplier blocks, equivalent to 500,000 histograms/second. Even allowing a large amount for accumulators, data flow control and for routing, this is a substantial amount of processing power. SPLASH, a reconfigurable FPGA-based system designed for string-matching yielded performance increases of 300 times over a supercomputer for the same task [11]

- [2] S.J.Evans, N.A.Thacker, R.B.Yates and P.A.Ivey, (1993), A Massively Parallel Vector Processor for Image Communications, Proc. Image Com. 93, Bordeaux, pp303-308.
- [3] S.J.Evans, S.N.Walker, N.A.Thacker, R.B.Yates and P.A.Ivey, (1994), A 1.2 GIP General Purpose Digital Image Processor, Proc. IEEE Custom Integrated Circuits Conference, San Diego.
- [4] S.B. Pollard, J.E.W. Mayhew and J.P. Frisby, (1985), PMF: A Stereo Correspondance Algorithm using a Disparity Gradient Limit, Perception, 14, pp449-470.
- [5] M. Rygol, S. Pollard and C. Brown, (1991), MARVIN and TINA: A multi-processor 3D Vision System, Concurrency: Practice and Experience, 3(4), pp333-356.
- [6] R.A. Lane, N.A. Thacker and N.L. Seed, (1993), Stretch-correlation as a realtime alternative to feature-based stereo matching algorithms, Image and Vision Computing (in press).; R.A. Lane, N.A. Thacker and P.A. Ivey, (1994), A Correlation Chip for Stereo Vision, BMVC94 (submitted)
- [7] C.A. Rothwell, A. Zisserman, D.A. Forsyth and J.L. Mundy, (1992), Canonical Frames for Planar Object Recognition, ECCV92, pp757-772.
- [8] A.C. Evans, N.A. Thacker and J.E.W. Mayhew, (1993), The Use of Geometric Histograms for Model-Based Object Recognition, Proc. 4th BMVC93, Guildford, 21-23 Sept., pp429-438.
- [9] C.I. Brown, P. Courtney and R.B. Yates, (1994), Assessment of the DIP chip for Object Recognition using Pairwise Geometric Histograms, Dept E&EE-ESG Memo 94/6.
- [10] P. Courtney and P. Riocreux, (1994), Evaluation of the IIT Processor as a platform for Object Recognition using Geometric Histograms, Dept E&EE-ESG Memo 94/5.
- [11] M. Gokhale, *et al.*, (1991), Building and Using a Highly Parallel Programmable Logic Array, IEEE Computer, pp81-89.