Very coarse granularity parallelism: implementing 3-D vision with transputers

P Healy and D Vernon
Department of Computer Science
Trinity College, Dublin
Ireland

One of the strategic issues of industrial vision concerns the implementation and exploitation of emerging passive 3-D techniques, such as stereopsis, camera motion, and shading analysis. Since dedicated hardware, in the form of custom-designed integrated circuits or array processors, is not likely to be flexible enough to cater for the varying requirements of these disparate techniques, real-time processing may be achieved through the use of loosely-coupled concurrent modules in the form of re-configurable transputer arrays. This paper describes the design and implementation of such a Re-configurable Vision Processor (RVP). A paradigm of very coarse granularity parallelism is employed with each node on the system running an identical 3-D vision software package. The vision package includes an interpreter for a vision programming language and, inter-node communication and parallelism is achieved by allowing each node to construct vision programs and request their execution on a neighbouring node. Examples of the vision capabilities of the system are presented and the paper includes a comparison of the system's speed with a more conventional architecture for which the software was originally developed.

***Mr Healy** is a research assistant in the Vision Group at Trinity College, Dublin. He is currently working on the development of parallel architectures for vision systems, on a project funded by the Irish Science and Technology Agency (EOLAS).*

***Dr Vernon** is a lecturer in the Department of Computer Science in Trinity College, Dublin. He is involved in several research projects in the area of computer vision, including projects funded by the European Strategic Program for Research and Development in Information Technology (ESPRIT) and the Irish Science and Technology Agency. He is a director of the Vision and Sensor Research Unit in the Irish Advanced Manufacturing Technology programme.*

Presented at Image Processing : Blenheim Online Publications, Pinner, Middx, UK. 1988

Introduction

It seems most surprising, but it is true nonetheless, that the vision techniques used by industry are still predominantly based on the 2-D binary vision techniques of the late 1970s [Luh and Klassen 1985; Pau 1984]. Certainly, there are exceptions to this as some research results find application in isolated industrial environments, either because the research has been specially commissioned in the first place or because the research institute has close ties with a commercial vision company (e.g. [Ray and Wilder 1984; Ray and Zeuch 1983]). For the most part, and this is particularly true of the small and medium size enterprises which endeavour to use vision in industrial inspection and simple part manipulation, they rely on simple segmentation, utilizing either binary thresholding or primitive edge detection such as gradient-based Sobel operator [Duda and Hart 1973], followed by the derivation of a number of simple features (e.g. position, orientation, area, first and second moments of area, radii signatures, minimum and maximum bounding rectangles [Berman et al. 1985; de Coulon et al 1983; Juvin and de Cosnac 1984]) or by local template matching (e.g. [Persoon 1978/79]) and, where appropriate, classification. Even the classification techniques tend to be primitive; it would appear that nearest-neighbour classifiers are more often used than Bayesian estimates. When it is necessary to make explicit the 3-D structure of the scene being analysed, industrial vision systems resort, almost without exception, to active sensing, depending to a large extent on light-striping and on laser range finding Bogaert and Ledoux 1983; El Hakim 1985; Horaud and Bolles 1984].

The implementation of popularly-accepted techniques has significant ramifications for the current philosophy of industrial vision: either the technique is implemented in software on a host computer, with severe limitations on the complexity and speed of potential algorithms, or available hardware is utilised. This hardware falls into three categories, none of which represent particularly suitable approaches for the requirements of complex vision: special purpose devices (e.g the Plessey chip for edge detection [Beedie 1986]), augmented frame-grabbers, and array-processors.

The second category, referred to here as augmented framegrabbers, are image acquisition devices which offer additional front-end processing. Quite often, these facilities are offered, not because there is an intrinsic demand for it, but because the technology is available. The type of processing offered includes the implementation of Look-Up Tables in real-time (both for in-coming and out-going data), finite-impulse response filtering facilitating, at best, 8x8 convolutions, and very restricted feature extraction (merely providing a vector of coordinates of blobs in a binary image) [Imaging Technology 1987, Datacube 1987]. Fortunately, this situation is maturing somewhat and more useful techniques, e.g. Difference of Gaussian (DoG) approximations to the Laplacian of Gaussian edge detector are now being offered [Datacube 1987].

The third category of hardware which is sometimes used in configuring industrial vision systems includes vector and array processor boards. These can be difficult to program and have quite a narrow domain of application; SIMD architectures do indeed have potential but they are not a panacea for vision implementations.

In 1981, Bolles drew a distinction between image understanding systems and industrial automation (vision) system [Bolles 1981], in that image understanding systems endeavour to deal with the complexity of real environments, addressing occlusion, shading, texture, deformation, while the industrial automation approach relies heavily on engineering the environment to facilitate image processing and simple image analysis. *This distinction, unfortunately, seems to be as true today as it was then.* There has been no general dissemination and exploitation of more advanced techniques for segmentation and for 3-D sensing, based on passive vision, though there are certainly many isolated examples of advanced vision being used for industrial applications, e.g. [Besl et al. 1985; Baylou et al. 1984; Ben Rhouma et al. 1983 ; Bolles and Cain 1982; Ikeuchi et al 1986; Fu 1982; Eshera and Fu 1986].

It is our contention that the low-level vision technology, such as Marr-Hildreth edge detection [Marr 1976; Marr and Hildreth 1980], stereo [Nishihara 1983], and camera motion [Sandini et al. 1986, Sandini and Tistarelli 1986, Horn and Schunck 1981]) are potentially very useful and applicable in industrial environments[Vernon and Tistarelli 1987].

However, the established user-base for such techniques is **not** large: the general situation is that applications engineers find themselves with a limited repertoire of simple, though useful, techniques and they have not ventured beyond for the past ten years. This is not wholly because the vision technology has not been developed but because it has not been made widely available in a sufficiently-useful general-purpose form. There are, of course, valid reasons for this lack of exploitation. Perhaps the main one is the computational complexity of the techniques and the inadequacy of existing hardware in dealing with this complexity. The problem is compounded by the disparity of computational paradigms: a complete vision system requires both low-level processing and high-level contextual analysis. While image processing techniques can, and have, exploited parallel architectures to obtain significantly increased speeds, the current level of development of parallel algorithms implementing the more sophisticated analysis functions in not advanced. For the present at least it would appear that sophisticated vision systems require hybrid architectures incorporating SIMD and MIMD approaches. There is one important additional reason why this is unlikely to change in the near future.

Recall the earlier definition of a tool as an integrated system, comprising hardware and software, such that the information it makes explicit is presented in a general, flexible, and tunable manner, allowing the imposition of application-dependent constraints. Most software which is provided with imaging systems tends to be predominantly of the image-processing type, i.e. it is capable of effecting image to image transformations, and little useful information is made explicit. The image analysis software, if provided, reflects the type of techniques that were discussed in the previous section and, as mentioned before, is largely binary in nature, being based significantly on the much-disseminated work performed at SRI toward the end of the 1970s. The information which is made explicit is, typically, a set of features, describing position, orientation, and shape in a trivial manner: *sophisticated data-structures, which might be used to integrate several redundant sources of visual information, are not supported.* Neither is there any facility for "system tuning", in the sense of being able to impose constraints on the derivation of information. However, sophisticated 3-D vision systems depend heavily on such integrated complex data-structures; structures which requires conventional Von Neumann architectures.

The sophisticated 3-D vision system, then, requires (at the very least):

- Fast low-level image processing (typically in the SIMD paradigm);
- Non-trivial image analysis;
- explicit representations of visual cues (requiring complex data-structures);
- explicit 3-D world models.

An industrial system also requires standard interfaces to facilitated integration in FMS and CIM system; however, such issues are outside the scope of this paper.

The approach we have adopted to satisfying the above requirements is to marry a 3-D computer vision software environment to a single, but powerful, processor node (comprising a T800 transputer, two INMOS A100 digital signal processing chips, and 2 Mbytes of memory); each node is replicated (four times in our present system) and each node can communicate with **and control** any other node in the system. The arbitrary connection topology is achieved using a cross-bar switching device which multiplexes the communications links emanating from each transputer.

The next section of the paper briefly describes the vision software environment and this is followed by a discussion of the design and implementation of the transputer hardware. Following this, we describe the integration of several of these nodes using the vision programming language and its virtual image structures (its image data-structure system). Details of the performance of the system are also included. Finally, the paper concludes with an eye to the future and we consider the next stages of the development of the 3-D vision system.

VIS: A 3-D Vision Software System

Resulting from research funded under the ESPRIT programme, a software system has been developed to facilitate research of low-level vision, with emphasis on the overall cohesion of multiple data-structures, and the perusal and manipulation of the information made explicit by the system, together with an awareness of the needs of sophisticated users and the type of machine interaction which might enhance efficiency [Sandini and Vernon 1987, Vernon and Sandini 1988]. The result of this research is a software system called VIS. One of the most important points to note is that VIS is a complete vision software environment, originally conceived for a single user on a conventional Von Neumann machine. It provides key features to allow a user to build systems of image structures, to archive them (or part of them), to dearchive them, and to process and analyse them. The main emphasis of the processing and analysis is "low-level" vision, i.e. Marr-Hildreth (Laplacian of Gaussian) edge detection [Marr and Hildreth 1980], contour and region analysis [Sandini and Vernon 1987], the generation of the "Raw Primal Sketch" [Marr 1982], and the inference of 3-D structure using stereopsis and motion

[Hildreth 1983; Horn and Schunck 1981; Sandini and Tistarelli 1985; Sandini et al 1986; Morasso et al. 1986; Frigatto et al. 1987; Sandini et al 1987].This acronym derives from the phrase Virtual Image System, so called as it allows a user to specify interactively the number, resolution, type, and the organisation of the images at run time; the number of images you can add is only constrained by the availability of memory. The atomic unit in VIS is the pyramid, i.e. a set of hierarchically related images, images are "added" by inserting them at specified level in the pyramid. Currently, pyramids can have up to ten levels corresponding to image resolutions of 1024x1024 pixels to 64x64 pixels, respectively.

Each pyramid (or image) is identified with one of three classes of representation: iconic, regional, and boundary. Within each classification there are further sub-categories so that each pyramid (or image) is tagged with an unambiguous and unique data type. This typing of images means that image processing is accomplished in a particularly simple manner: by transferring an image from one pyramid to another pyramid. All processing is effected implicitly. In addition, the system allows extensive windowing on both source and destination pyramids so that, in addition to being transformed, the destination image may be a scaled version of the source window.

Iconic types are: framestore, intensity, stereo disparity, motion, convolution (result of convolving an intensity image with a Laplacian of Gaussian mask), zero-crossing (this is derived from the convolution image and represents the location of the intensity discontinuities - edges - in the image), and range. The regional type represents explicitly the hierarchical nesting of regions on uniform sign in the convolution image. The boundary types represent different measures computed at the zero-crossings of Laplacian of Gaussian filtered images. The measures include zero-crossing position, strength, direction, disparity, velocity, and depth.

Once a system has been configured, and, possibly after the user has done some processing and analysis, he or she may wish to suspend the session and resume work later. This can be accomplished by saving the system status and later restoring it. This system status, represented by the complete Virtual Image Structure, is pivotal in the exploitation of multiple concurrent VIS machines in that is provides a user with a particularly elegant method to exchange visual information (and data-structures) between machines.

Users can communicate with VIS in two ways: via a menu system or using an in-built programming language called VISICL (VIS Interpretive Control Language). For the purposes of this paper, VISICL is the most important medium for controlling VIS since not only can a user configure and control an image processing session with a VISICL program but so to can another VIS system. We shall discuss this in more detail in the section on integrating the software and the hardware.

Much of low-level computer vision, and in turn, high-level vision, depends on one initial and key step: the detection of intensity discontinuities in the image. Normally, this step is referred to as edge detection. As we have noted, VIS utilises one of the most popular and advanced techniques for detection of intensity discontinuities: the Marr-Hildreth theory of edge detection which uses Laplacian of Gaussian filters. This theory is based on two premises: that discontinuities in image intensity give rise to zero-crossings (i.e. points were the image function change sign when differentiated twice) and that: the detail of these zero-crossings can be controlled by first smoothing the image with a Gaussian function. VIS provides for a complete set of image types corresponding to different level and stages of low-level processing. Since each of these image types are inter-related, VIS also links them together allows the user to exploit these linkages. There are several VISICL primitives to compute various statistics of the contour and region properties and to select and deselect contours and regions on the bases of those statistics.

Although the extraction of intensity discontinuities is fundamentally important, it is just a first step. Subsequent stages include the computation of depth (i.e. the recovery of 3-D structure) of the scene using, e.g., stereopsis or motion and the grouping of the image contours and their description using higher-level primitives such as line segments.

Low-level early vision is not, of course, an end in itself. The information generated at this stage must necessarily feed the higher level cognitive modelling sub-systems and, indeed, these cognitive systems must control the low-level processing. Here too VISICL facilitates this two-way flow of control and information. Cognitive modelling systems, typically running an alien host computer generate appropriate VISICL programs, transfer them to the VIS host where they are automatically executed, and the resultant information is then transferred back to the alien cognitive modelling system. At present, the information which is transferred is a modified version of Marr's Raw Primal Sketch, i.e. a description of the scene in terms of lines (comprising many straight edge segments and delimited by terminations). Bar and blob primitives are also generated but, as yet, these are not communicated to the alien system. The edge segments in this instance are identified by an initium and a terminus; two points specified, not just in 2-D, but by 3-D coordinates. Since these edge segments are generated from the zero-crossing contours, the third dimension can be directly extracted from the depth map derived from the stereo and motion ranging process.

Configuring the Hardware

The hardware characteristics have from the outset been driven by the requirements of the VIS algorithms. These manifest themselves in a need for a fast 1-D convolution and a hardware floating point unit coupled to a fast sequential processor. The first requirement is met by the inclusion of the INMOS A100 [INMOS 1987a], a 32 stage transversal filter chip, and the requirement by using the INMOS T800, a 10MIPS / 1.5 MFLOPS 32 bit processor [INMOS 1987b].

The system then is intended to act as a VIS work platform for industrial applications, where high performance is required at less than the cost of conventional solutions. As part of this philosophy, a transputer controlled frame grabber is being built, in conjunction with the system described here, to provide the frame capture and display function. The system is also designed to provide for ease of development work, without conflicting with its primary industrial application role.

The principal design criteria were based on consideration of the number of useful processing units that could be fitted into the area of an industry standard VME-bus board. In addition it was felt that more than one transputer node would be needed per board to most easily exploit the potential of transputer based parallel processing system. As regards the computationally expensive convolution filtering of 512x512 images, the practical minimum number of stages in the transversal filter was in the order of fifty. This requirement was met by including two A100 devices in each node.

In order to keep the volume of communication, in a relatively small scale vision system, to within acceptable limits, and bearing in mind the finite bandwidth of the inter-node communication using links, with only four links per transputer, the benefits of keeping a realistic number (greater than one) of 512x512 images in memory at one time are important. This is especially true with regards to memory hungry 3-D systems. High capacity dynamic RAM meets this requirement with little cost in terms of physical space. The memory interface of present 32 bit transputers is at any rate limited to a minimum cycle time of three processor cycles for a read or write to memory.

The transputer to A100 interface is kept simple in order to facilitate the placing of the maximum number of processing units rather than interface circuitry, and also to keep the software device driver reasonably flexible to use. The standard memory interface of the A100 is used to provide for both data input and output, as well for control functions. Using transputer DMA links to feed data into and take data out the A100 cascade was considered unfeasible within the space constraints. While it would have in effect allowed reconfigurable A100 devices to be allocated over DMA links, the conclusion came to was that, in any moderately sized system the DMA link resources of a network would be better allocated to conventional internode communication.

Internode communication is performed solely over transputer links, a proportion of which are switched through a crossbar device under the control of the host. Links connecting to the dual transputer framestore do not go though the crossbar switch in systems of four or eight processor nodes. The result is that any processor node is capable of obtaining an image immediately without the need for a reconfiguration phase affecting other parts of the network. With T800 transputer links running at 20Mbits/sec, the effective throughput of a single link is in the order of five 512x512 images per second. This routing simplification has considerable implications on software complexity required for a node to acquire or display an image, reducing it significantly.

One of the goals of the software implementation is to keep communication between the transputer network and conventional host system to minimum. This in practice means the archiving of a database at the end of a session and the dearchiving at the beginning of the next session. The traffic between host and network during a session is handled by low bandwidth transputer serial links mapped onto the VME bus using INMOS link adaptors.

Experience gained from working with an INMOS B004 on a PC bus development board suggests that transputer to host communication would benefit from the support of a DMA block move to and from the network, on the host side of the bus. Significant improvements in throughput however were made by simply buffering commands and data on the transputer side of the system before passing them to conventional hardware devices residing on the host bus. The result was that even a polled transfer implementation worked well in practice with little in the way of noticeable overhead, compared with the same software running native code on the host PC. This was especially evident in accessing a conventional framestore on the PC bus, where unexpectedly good results were obtained.

Crossbar reconfiguration under software control is an area under going study at the moment. Initially it seems that it may not be practical to use more than a very limited number of configurations for any particular application. With relatively small systems, which exhibit a high connectivity, the emphasis is more on identifying a small optimum number of configurations in the development stage of an application.

For convenience the prototype node implementation followed the INMOS TRAM architecture specification [Walker 1987]. This choice allowed for a wider range of potential hosts without sacrificing anything in the way of functionality. A lower density host mother board is another valuable consequence of taking this path. Expansion paths will considerably eased through the vertical stacking of TRAM compatible modules. A simplified block diagram of a single node appears in figure 1.

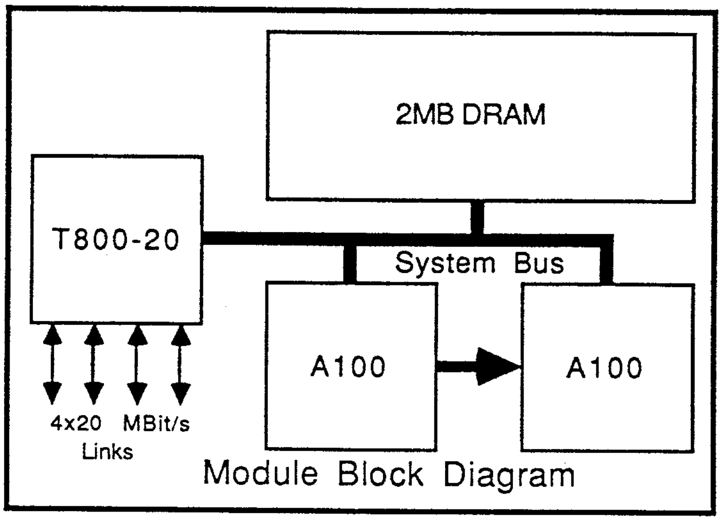

Figure 1.

Marrying the Software and Hardware: Coarse Granularity Parallelism

So far, we have described the vision system VIS which effects the 3-D visual sensing and a four-transputer VME-bus board comprising four modular transputer nodes in a standard TRAM format. VIS runs on each node and, in its own right, an individual node represents quite a powerful vision system. However, the true power of the system only emerges when one exploits the facility to link the nodes together. To link quasi-independent nodes requires both a physical medium, supplied in this instance by the 20MHz transputer links (four per transputer), and a communication protocol. As VIS supports its own programming language VISICL, the natural protocol is VISICL itself; a VISICL program represents the control aspect of the protocol while the visual data is represented in the most general case by the Virtual Image Structure, i.e. the VIS data-structure. Thus, rather than a user restoring some system from disk, invoking a VISICL program, and saving the processed system back on disk (as would be the case in a normal interactive session), the system is "restored" from a remote node, a VISICL program supplied from the same remote node and executed on the local node, and the processed system is "saved" back to the remote node. In cases where the information to be processed is iconic in nature, the VIRTUAL FRAMESTORE image type can be exploited to effect the exchange of information. The VIRTUAL FRAMESTORE is a logical device rather than a physical one, normally implemented as a disk file, and images can be read from and written to these logical devices or files. Thus, instead of saving and restoring (remote) systems, one transfers to and from (remote) VIRTUAL FRAMESTORES.

The elegance of this approach is apparent when one recognises that the only additions to the existing single computer VIS which had to be implemented to effect concurrency in this manner were the ability to recognise a link specification prefix on filenames (and redirecting data appropriately) and the presence of some VISICL primitive to "remotely execute" a VISICL program. This minimal overhead on development (and deployment) is one of the main advantages of the coarse granularity. The following VISICL (pseudo-)code will serve to illustrate the approach.

```
PROCEDURE process_1(sigma,  VFS_filename, edge_filename)

   /* procedure to process image data in a VFS and generate Raw */
   /* Primal Sketch edge information at a given spatial scale   */
   /* determined by the value of sigma.                         */
   /* The edge information is written to file.                  */

   configure_system;
   transfer_image(VFS_filename, 1);
   request_edge_information(edge_filename, sigma);

ENDPROC

/* remotely execute process_1 on the node connected by link 1;  */
/* it will expect to receive information from a VFS on its link */
/* 0 and it will write the edge data to a file on link 0.       */

remote_execute(link_1,process_1(6,"link0:test.vfs",
                                 "link0:test.dat"));

/* send the VFS image data to the remote VIS */

transfer_image(test.vfs, link_1:test.vfs);

/* now proceed to do some work on the local node */

process_1(12, test.vfs, test12.dat);

/* now collect the results from the remote */

copy_file (link_1:test.dat, test6.dat);
```

It should be noted that the remote execution of procedures can be effected recursively. Thus, a single program can be propagated through a list or tree of nodes. The onus is on the vision programmer, however, to ensure an orderly distribution of processing and collection of results. This is a significant deficiency which is presently being addressed.

To give the reader some indication of the capabilities of the vision system, diagrams 1 and 2 depict two stereo images with the disparity computed in local patches being shown in diagram 3. Diagram 4 illustrates four snapshots in a camera motion sequence as the camera moves around the object; the resultant velocity vectors and a "side elevation" of the corresponding depth image (in which depth is inversely proportional to the distance from the bottom of the image) are shown in diagrams 5 and 6 respectively.

The Laplacian of Gaussian convolution used within VIS operates by decomposing the operation into a number of 1-D convolutions. In this form the convolutions are particularly suited to operation with the A100, giving rise to significant increases in performance, as compared to a processor working on its own. The relative power of the system is summarised in Table 1.

Description of System	Image Size	
	512x512	256x256
IBM PC-AT, 80286 8MHz, SCO Xenix	33 Min 45s	7 Min 35s
Apple Mac II, 68020 16MHz, Lightspeed C	13 Min 15s	3 Min 5s
DEC µVAXII	11 Min 45s	2 Min 50s
T800 20MHz, INMOS/3L C compiler V1.3	7 Min 25s	1 Min 40s
T800 (as above) with 2 x A100	36s	-
T800 (as above) with 2 x A100 (projected)	<10s	-

Table 1. Timings taken for VIS to accomplish a Laplacian of Gaussian convolution with a mask size set at 51.

A software communication protocol for multi-node processing based on the INMOS alien file server protocol [INMOS 1988a] has been written to facilitate communication between VIS nodes in a general and automatic manner. Possible modifications include the addition of low level support for image transfer operations to make maximum use of the transputer link bandwidth.

The number of virtual VIS processes actually executing on a physical node depends on the memory resources available to the node, but in general only a small number of processes shall be active in any one time period. The exceptions are the transputers controlling the framestore, where there shall be at least four VIS processes per transputer and probably five. The four processes shall act as servers for the rest of the network handling requests for input and output of image data. The fifth process will act as an arbitrator between input and output requests from VIS processes running on the framestore transputer.

Toward the Future

Experience gained performing complex 3-D image processing with VIS running on transputer nodes with 2Mbytes of memory has shown the need for some nodes with more than this minimum amount of memory. As a first step, a node with twice the amount of memory, 4Mbytes, has been designed and it is likely that bigger nodes will be constructed will the availability of larger capacity dynamic RAM. Advances in VLSI technology have been noted as well, and it is likely that a transputer node incorporating the INMOS A110 DSP device will be made [INMOS 1988b]. The A110 is a 2-D convolver with on-chip programmable line delays. Work is also being done at the moment looking at the possibilities of interfacing a transputer node directly to a video bandwidth parallel data path, to supplement the existing serial link channels. This option is being undertaken to exploit the availability of the off the shelf third party vision hardware.

A software system which exists at one level higher than that of the low level VIS routines and which functions in a loosely-coupled feed-back loop with the lower routines is envisaged. This entity would be responsible for the following:

- Automatic allocation of work to the network through load balancing the work distributed to the VIS nodes. This parceling out of work will make use of the hybrid algorithmic and geometric paradigms. In the first instance it is envisaged that a processor farm type implementation will be evaluated.

- As part of the load balancing effort it is envisaged that automatic reconfiguration will be required as conditions warrant within the network

Research work is currently being migrated to a NS32332 based VME system running GENIX, which will serve as a host for the next generation hardware and software development. This system has an INMOS link adaptor wired into its DMA controller, which should achieve data transfer rates of over 300Kbytes/sec from the transputer system to the host processor.

Conclusions

Vision research is entering a new age, with strategic developments being undertaken in the critical areas of dynamic active visual sensing (processing as many images as is necessary without cost and controlling the image acquisition to enable the system to intrude into and interact with its environment); broadly-based early processing (incorporating texture and colour); expositions of shape which address form, function, and reasoning; cue integration; robustness and autonomy; knowledge and reasoning; and hybrid computing paradigms. Existing research programs such as ESPRIT and ALVEY are responsible in major part for realising the critical mass of work which has brought this situation about and there is great cause for optimism in the potential of these advanced research initiatives.

However, from an industrial perspective, the present position is not quite so bright and the reconciliation of the requirements of industry and the current and expected capability of research laboratories is pivotal. Existing vision tools which industrial engineers have to work with are wholly inadequate: there is an urgent need of much more enhanced visual capabilities for automation than are currently available or than seem to be likely to be available in the very near future. It is imperative that advanced vision tools be immediately developed and deployed in industry. It is our contention that the system described in this paper represents a good starting point from which such tools can be constructed.

Acknowledgments

The work described in this paper was funded partially by the European Strategic Program for Research and Development in Information Technology (ESPRIT) and by EOLAS, the Irish Science and Technology Agency. The authors would like to acknowledge the collaborative efforts of our colleagues Prof. G. Sandini and M. Tistarelli at the University of Genova and of M. Flanagan at Trinity College, Dublin, in designing and implementing VIS. We would also like to recognise the the work of Dr. B. Coghlan in the Department of Computer Science, Trinity College Dublin, for his part in the design of the TRAM-format transputer node.

References

Baylou, P., El Hadj Amor, B., Monfion, N, Bouvet, C., and Bousseau, G., 1984. "Detection and 3-D localisation by stereoscopic visual sensor and its application to a robot for picking asparagus", Pattern Recognition, Vol. 17, No. 4, pp.377-384.

Beedie, M. 1986. "Image IC detects edges in real time", Electronic Design, May 15th, pp.57-58.

Ben Rhouma, K., Peralta, L., and Osorio, A., 1983. "A 'K2D' perception approach for assembly robots", Signal Processing II: Theory and Application, Schurrler, H.W. (Ed.), Elsevier Science Publishers B.V., (North-Holland), pp.629-632.

Berman, S., Parikh, P., and Lee, C.S.G. 1985. "Computer Recognition of two overlapping parts using a single camera", IEEE Computer, pp.70-80.

Besl, P.J. and Jain, R. 1985. "Three-Dimensioal Object Recognition", ACM Computing Surveys, Vol. 17, No. 1, pp.75-145.

Bogaert, M. and Ledoux, O. 1983. "3-D perception in industrial environment", Proc. SPIE, Vol. 449, pp.373-380.

Bolles, R.C. 1981. "An overview of applications of image understanding to industrial automation", SRI International, Tech. Note No. 242.

Bolles, R.C. and Cain, R.A. 1982. "Recognising and Locating Partially-Visible Objects: the Local Feature Focus Method", The Int. Journal of Robotics Research, Vol. 1, No. 3, 1982, pp.57-92.

Datacube Inc., MaxVideo Product Spec, PO Box 527, Reading, Mass. USA.

de Coulon, F., Dessimoz, J-D, Kammenos, P., Unser, M., and Zurcher, J-M, 1983. "Conception de Systemes de Vision Artificielle pour la Localisation, l'Identification, et l'Inspection de produits Industriels", in "Point en Robotique", (Coordonnateur: J.C. Pruvost), Vol. 1, pp.84-104.

Duda, R.O., and Hart, P.E. 1973. "Pattern Classification and Scene Analysis".

El-Hakim, S.F. 1985. "A photogrammetric vision system for robots", Photogrammetric Engineering and Remote Sensing, Vol. 51, No. 5, pp.545-552.

Eshera, M.A. and Fu, K. 1986. "An image understanding system using attributed symbolic representation and inexact graph matching", IEEE Trans.

Fu, K. 1982. "Pattern recognition for automatic visual inspection", Computer, Vol. 15, No. 12, pp.34-40.

Frigato, C., Grosso, E. and Sandini, G. 1987. "Integration of Edge Stereo Information", ESPRIT Project 419 - Technical Report TK1-WP1-DI3.

Hildreth, E.C. 1983. The Measurement of Visual Motion, MIT Press, Cambridge, USA.

Horaud, P. and Bolles, R.C. 1984. "3DPO's strategy for matching 3D objects in range data", International Conf. on Robotics, Atlanta, GA, USA, pp.78-85.

Horn B.K.P. and Schunck, B.G. 1981. "Determining Optical Flow", Artificial Intelligence 17, no. 1-3, pp.185-204.

Horn, B.K.P. and Ikeuchi, K. 1983. "Picking Parts out of a Bin", A.I. Memo No.746, MIT A.I. Lab.

Ikeuchi, K., Nishihara, H.K., Horn, B.K., Sobalvarro, P., and Nagata, S. 1986. "Determining Grasp Configurations using Photometric Stereo and the PRISM Binocular Stereo System", The International Journal of Robotics Research, Vol. 5, No. 1, pp.46-65.

Imaging Technology Inc., Series 200 Product Brief, High-Performance Image Processing Subsystem, Part #47-B20000-00, 600 West Cummings Park, Woburn, Mass. USA.

INMOS Ltd. 1988a, Occam toolset User manual, 1000 Aztec West, Almondsbury, Bristol, UK.

INMOS Ltd. 1988b, IMS A110 Image and Signal Processing Sub-system, INMOS Ltd., UK.

INMOS Ltd. 1987a, IMS A100 Cascadable Signal Processor, INMOS Ltd., UK.

INMOS Ltd. 1987b, IMS T800 tranputer, Engineering data, INMOS Ltd., UK.

Iversen, W. 1982. "Robots pick parts out of a bin", Electonics, Nov. 30th, p. 50.

Juvin, D. and de Cosnac, B. 1984. "ANIMA 2: Un system generale de vision pour la robotique", Proc. Priemier Colloque Image, Biarritz, pp.165-169.

Luh, J.Y.S., and Klassen, J.A. 1985. "A three-dimensional vision by off-shelf system with multi-cameras", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No. 1, pp.35-45.

Marr, D. 1976. "Early Processing of Visual Information", Philosophical Transactions of the Royal Society of London, B275, pp.483-524.

Marr, D. and Hildreth, E. 1980. "Theory of Edge Detection", Proc.of the Royal Society of London, B207, pp.187-217.

Marr, D. 1982. Vision, Freeman and Co. San Francisco.

Morasso, P.,Sandini, M. and Tistarelli, M. 1986. "Active Vision: Integration of Fixed and Mobile Cameras", NATO ARW on Sensors and Sensory Systems for Advanced Robots, Berlin, Heidelberg, Springer-Verlag.

Nishihara, H.K. 1983. "PRISM: A Practical Realtime Imaging Stereo Matcher", A.I. Memo 780, MIT A.I. Laboratory, Boston, Massachusetts, USA.

Pau, L.F. 1984. "Approaches to imdustrial image processing and their limitations", Electronics and Power, February, pp.135-140.

Persoon, E.H.J. 1978/79 "A System that can learn to recognise two-dimensional shapes", Philips Technical Review, Vol. 38, No. 11/12, pp.356-363.

Ray, R. and Zeuch, N. 1984. "Vision guided robotic arm control for part acquisition", Proc. of the Robotic Intelligence and Productivity Conference, MI, USA, pp.30-34.

Ray, R. and Wilder, J. 1984. "Visual and tactile sensing system for robotic acquisition of jumbled parts", Proc. of the Robotic Intelligence and Productivity Conference, Detroit, MI, USA, pp.30-34.

Sandini, G., Morasso, P. and Tistarelli, M. 1987. "Motor and Spatial Aspects in Artificial Vision", Proc. of 4th. Intl. Symposium of Robotics Research, Santa Cruz, SRI International.

Sandini, G., Tagliasco, V., and Tistarelli, M. 1986. "Analysis of Object Motion and Camera Motion in Real Scenes", Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, pp. 627-632.

Sandini, G. and Tistarelli, M. 1985. "Analysis of Image Sequences", Proc. IFAC Symposium on Robot Control, pp.289-292.

Sandini, G. and Tistarelli, M. 1986. "Analysis of Camera Motion through Image Sequences", Advances in Image Processing and Pattern Recognition, V. Cappellini ad R. Marcone (Editors), Elsevier Science Publishers B.V. (North-Holland), pp. 100-106.

Sandini, G. and Vernon, D. 1987. "Tools for Integration of Perceptual Data", in "ESPRIT '86: Results and Achievements", Directorate General XIII (Eds.) Elsevier Science Publishers B.V. (North Holland), pp.855-865.

Vernon, D. and Tistarelli, M. 1987. "Range Estimation of Parts in Bins using Camera Motion", Proceedings of SPIE's 31st. Annual International Technical Symposium on Optical and Optoelectronic Applied Science and Engineering, in press.

Vernon, D. and Sandini, G. 1988. "VIS: A Virtual Image System for Image Understanding", Software Practice and Experience, Vol. 18, No. 5, pp. 395-414, 1988.

Walker, P. 1987, Dual Inline Transputer Modules, Technical Note 29, INMOS Ltd.

Yassaie, H. 1986, Correlation and convolution with the IMS A100, IMS A100 Application Note 3, INMOS Ltd.

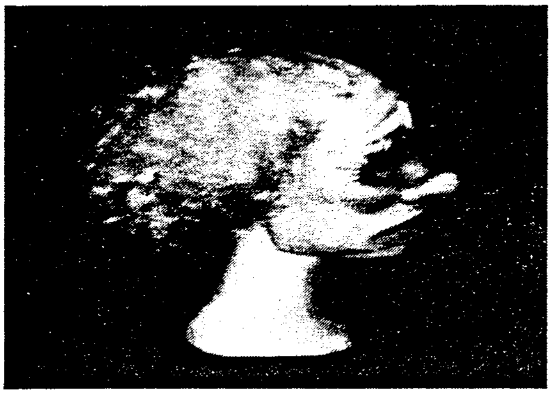

Diagram 1: Left Image in a Stereo Pair

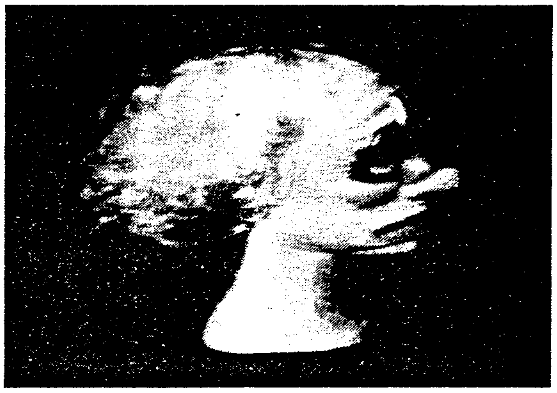

Diagram 2: Right Image in a Stereo Pair

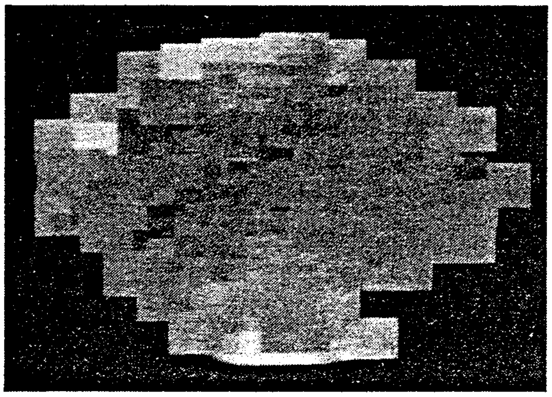

Diagram 3: Stereo Disparity

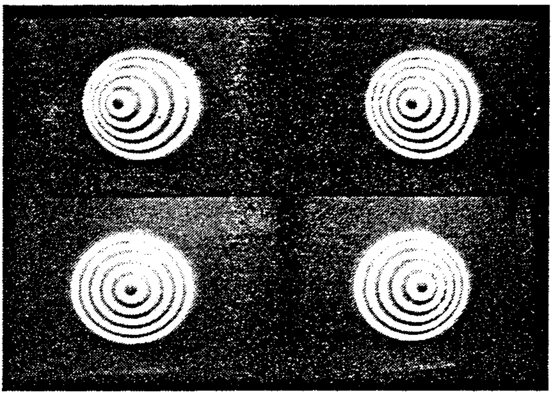

Diagram 4: Motion Sequence

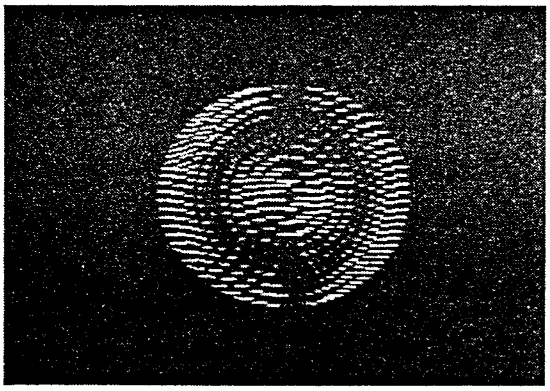

Diagram 5: Velocity Vectors

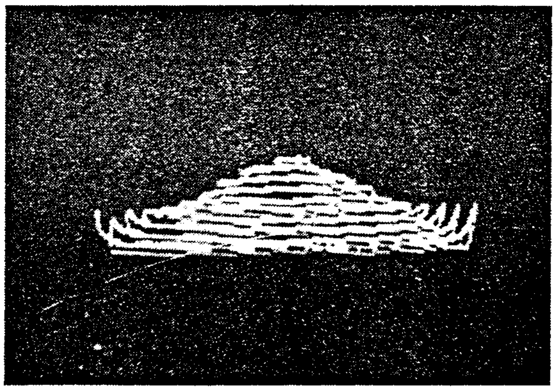

Diagram 6: Depth Image (side elevation of contours)

British Library Cataloguing in Publication Data

Image Processing 88
Proceedings of the conference held in London,
December 1988
1. Image processing
621.38'0414

ISBN 0-86353-157-1

ISBN 0 86353 157 1

Printed in the UK

The papers in this book are presented by the individual authors; the views expressed are their own and do not necessarily represent the views of the Publisher. Whilst every effort has been made to ensure the accuracy of the information contained in this book, the Publisher cannot be held liable for any errors or omissions however caused.