ARCHITECTURES FOR COMPUTER VISION

David Milford*

This paper discusses the problems underlying the design of hardware for the task of computer vision and describes an architecture which has been developed at the University of Bristol for use in a variety of applications. A modular heterogeneous architecture is shown to provide an effective and flexible solution to the problem of algorithmic diversity which is characteristic of computer vision applications.

1. Introduction

Computer vision is currently a very active research area and seems to provide a range of particularly intractable problems. Some workers have concentrated on a study of the human psycho-visual system and this approach has yielded some important guidelines to the implementation of a computer-based vision system. Nevertheless, radically new architectures are slow to emerge and most mainstream research remains tied to a range of well-established machine types. The emphasis in this case is on the development of new algorithms for extracting information from images. Table 1 illustrates the range of operations which are typical of image processing algorithms, organised very roughly according to their position in the hierarchy.

Table 1. The image processing hierarchy.

high level	OBJECTS	image interpretation
	PRIMITIVES	symbolic manipulation
		segmentation
		texture analysis
	FEATURES	edge extraction
		histogramming
		edge enhancement
		thresholding
		filtering
low level	PIXELS	convolution

* Deptartment of Electrical and Electronic Engineering, University of Bristol, University Walk, Bristol BS8 1TR, UK

At the lowest level, large amounts of pixel data are processed by operations which, in general, are reasonably simple and localised to small regions of the image. These low-level processes form the basis for further processing with the aim of extracting features from the image and, from these, deducing elementary shapes and finally identifying entire objects. As we move up through the hierarchy, the algorithms which are required become more global in nature, more complex and unfortunately less well-understood. It is clear that the operations at all levels are likely to be computationally expensive, either by virtue of the large quantity of data or because of the algorithmic complexity. In a computer vision application the real-time requirement demands extensive use of parallel processing. However, the algorithmic diversity suggests that no single architecture or programming model can be effective at every level.

Figure 1 illustrates the problem, shown as a funnel which concentrates the massive data input into a concise interpretation at the output.

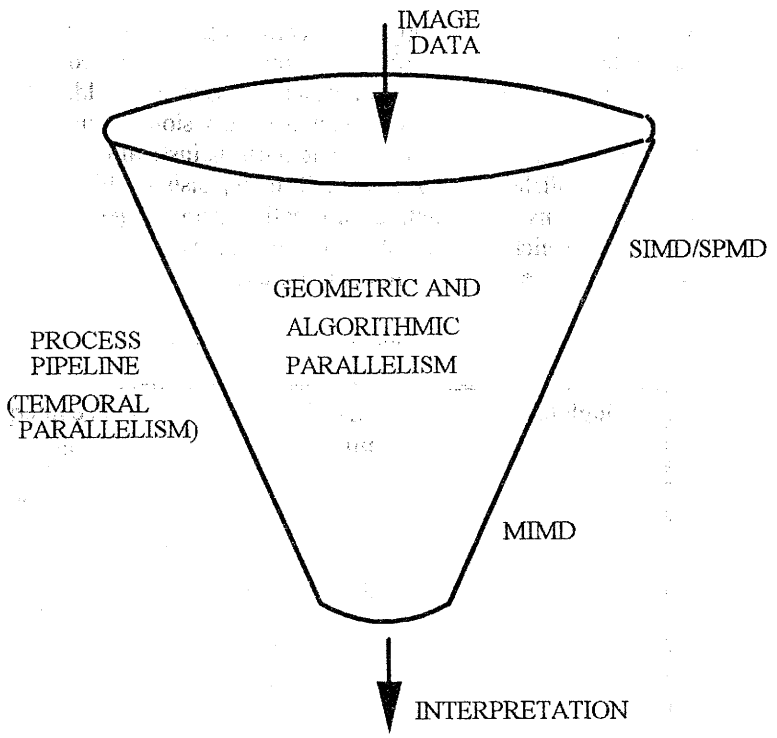

Fig. 1. The image data processing funnel.

Where localised processing is applied to large amounts of data SIMD or SPMD (single program, multiple data) architectures are appropriate but many of these processes can also be performed by application specific hardware. At the higher algorithmic levels a MIMD architecture is generally most suitable. The use of geometric

and algorithmic parallelism at different levels should be pipelined to achieve greater throughput.

2. The Bristol Vision System

In the Advanced Computing Research Centre at the University of Bristol we are developing a modular heterogeneous architecture which has been used in a variety of applications. Each module (Fig. 2) has a standard interface facilitating a pipelined architecture which is tightly coupled to a MIMD processor array. A MAXbus (Datacube) interface provides the means to construct a pipeline of digital video processors with pixel rates up to 20 MHz. The MAXbus connection system (which is essentially single source, multiple destination) allows modules to be joined in a variety of branching pipeline configurations according to the application. Each module includes an Inmos transputer which communicates with others in the system by passing messages through high-speed serial links. The actual transputer employed depends on its processing burden. The modules in use and under development are described below.

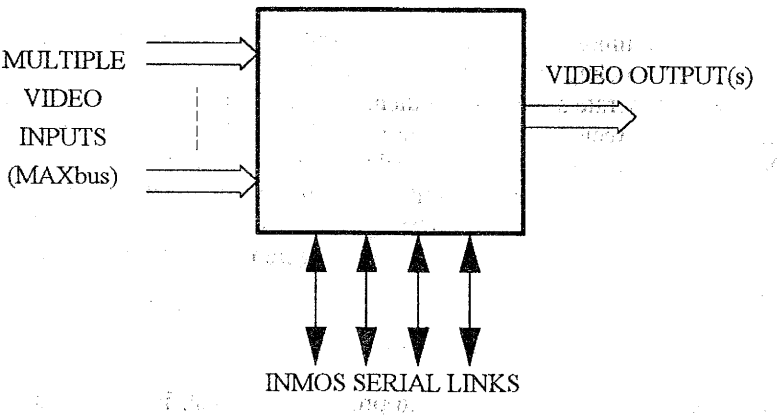

Fig. 2. The generic module.

2.1. Video I/O

Analogue video, selected from one of four input channels, is digitised to provide an image resolution of 768 x 576 pixels using interlace. The digital video data is transmitted with alternate pixels on each of two MAXbus channels allowing subsequent processing to be performed at frame rate with full resolution or at field rate with half resolution. The MAXbus timing signals can be synchronised with any one of the four input channels including signals originating from a VCR. A filter is provided as a programmable option for removing colour sub-carrier from the video input prior to digitisation.

MAXbus input is converted to analogue video for display on a RGB colour monitor. Full and half resolution display modes are supported with an overlay facility to enable use of a programmable cross-wire cursor. A 256 x 3 x 8 bit colour look-up table is used to map the input pixel values to pseudocolour.

All programmable options within the module are configured by an on-board T222 transputer which responds to commands from a master unit.

2.2. Image Filtering

The filter module is based on four SGS-Thomson A110 chips and provides a range of configurations and capabilities. It can operate with 10 and 20 Mbyte/sec data rates but contains no storage other than the inherent storage due to the pipeline delay. The A110 chips can be configured to give 28*3, 14*6 and 7*12 filter arrays or operated as two independent 14*3 or 7*6 filter arrays. A T2 transputer provides the system configuration interface and has access to all the hardware resources on the board.

2.3. General Function

This is a delay and function module that can produce any 8-bit function of two video inputs (Fig. 3). The lookup tables (LUT) can be used to provide two independent output functions (e.g. to generate angle and gradient from vertical and horizontal Sobel filters) or, by switching between LUTs, generate a single output function which varies with time. One of the video streams can be delayed before the function is applied or the output of the function unit can be delayed and combined with another input. The former provides processing of consecutive frames while the latter allows a temporal smoothing function to be performed. A T425 is used to configure the module and to compute new LUT functions as they are required.

2.4. Region-Of-Interest (ROI) Frame Store

The objective of this module (Fig. 4) is to provide a low cost, high performance frame processing module with region-of-interest capability. It can simultaneously capture and output video streams using a double-buffered video memory. ROI capability is achieved by the design of memory controllers (using FPGAs) which allow complete independence in the timing of the two streams. The module has 2 Mbytes of image store, all of which is continuously accessible by the on-board T800 transputer. There is, in addition, up to 4 Mbytes of general purpose DRAM. These modules satisfy the need for global access to image data at the pixel level and, having extracted feature data, provide an efficient interface to an array of transputers which can perform the higher level algorithms.

2.5. GLiTCH Processor

This module (under development) is based on the GLiTCH chip (Storer et al, 1992), an associative array processor designed at Bristol to implement a variety of image processing functions. The GLiTCH module is able to process small regions of the image

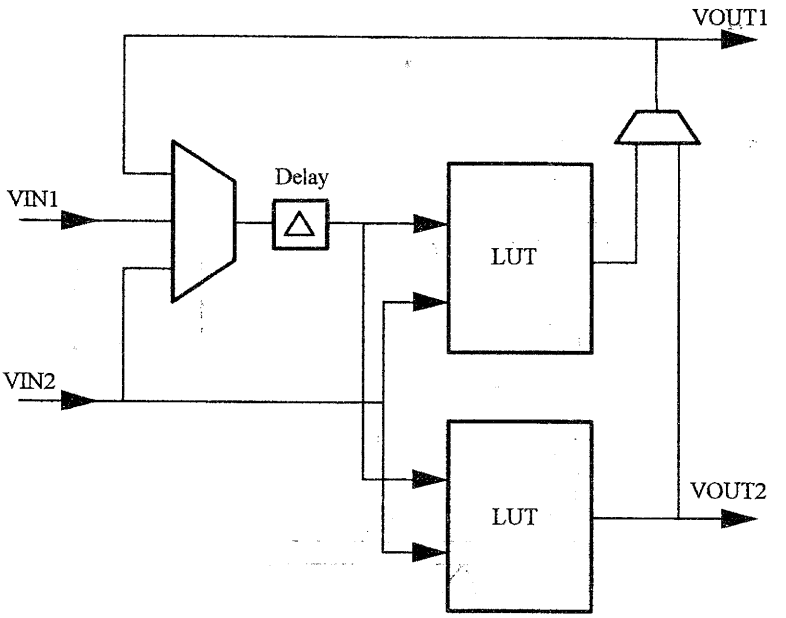

Fig. 3. Structure of general function module.

very quickly so it will be used in conjunction with framestore modules using their ROI capability as described below. A complete image is captured by a framestore then transmitted region by region to the GLiTCH module. The processed "patches" can then be received by a second framestore and pieced together for subsequent processing.

3. Conclusion

The system described in this paper has been successfully used in a variety of computer vision applications including:

- autonomous vehicle control (Thomas et al, 1991),
- vehicle number plate recognition,
- automated engine inspection.

The heterogenous modular architecture allows units to be connected together in an application-specific configuration unified by the transputer interconnection network. A software library supports each module so the program development for a customised application is relatively straightforward.

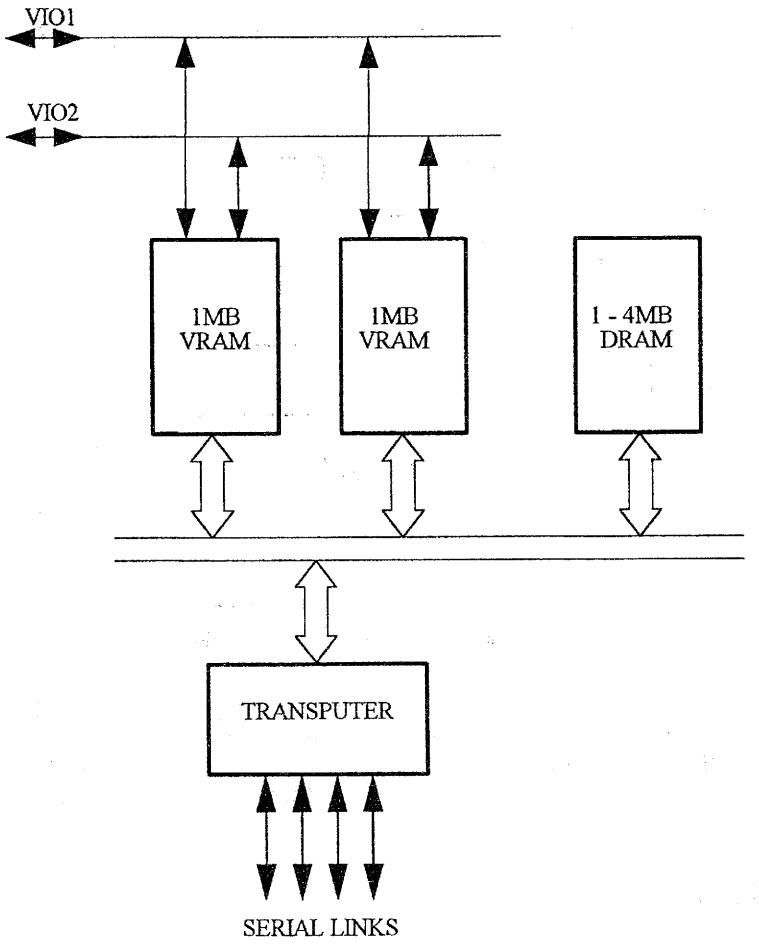

Fig. 4.

References

Datacube Inc: *MAXbus Specification and Users Manual.*

Storer R., Pout M., Thomson A.R., Dagless E.L., Duller A.W.G., Marriott A.P. and Hicks P.J. (1992): *An associative processing module for a heterogeneous vision architecture.*- IEEE Micro, v.12, No.3, pp.42-55.

Thomas B.T., Dagless E.L., Milford D.J., Morgan A.D. and Schaaser L.T. (1991): *Real-time vision guided navigation.*- IJEAAI, Part 4, v.4, pp. 287-300.