91.08

BIRA - Belgian Institute for Automatic Control
Working Party Digital Techniques and Computer Systems (DTCS)

Image Generation and Image Analysis on Transputers

Ir. W. HUISKAMP, TNO Physics and Electronics Laboratory, NL

Antwerp, 21 November 1991

Image generation and Image analysis on Transputers

Ir. W. Huiskamp

TNO Physics and Electronics Laboratory
Parallel Processing
P.O. Box 96864
2509 JG The Hague
The Netherlands

phone : +31 70 3264221
fax : +31 70 3280961

Abstract

The TNO Physics and Electronics Laboratory (TNO-FEL) in The Hague is part of the TNO Defence Research (TNO-DO) of the Netherlands Organisation for Applied Scientific Research. The activities of TNO-FEL focus primarily on operational research, information processing, communication and sensor systems. To support the fast data-processing often required in sensor system applications, research was started on parallel processing. The two main areas of interest are transputers [1] and neural network technology [2]. This research has now resulted in two major application areas : real-time computer generated imagery for simulators and 3D image analysis for computer vision systems. This presentation gives an overview of the development and implementation of these systems. The impact of the next Transputer generation on these applications is also discussed.

INTRODUCTION

Computer applications tend to need increasing amounts of processing capacities. Single processor systems are reaching the limits of performance improvements. It is obvious that using more processors running in parallel could (theoretically) provide unlimited power. Many existing sequential programs could benefit from being able to perform more than one action at a time. The main difficulty with parallel processing is to find an efficient way of decomposing the problem in a number of processes that can run concurrently. There are two basic approaches to this problem : data parallelism and algorithmic parallelism. In data parallelism the data is split up in independent parts. Each processor in the network will essentially perform the same operation on a different part of the data set. This option implies that the original (sequential) algorithms can often be re-used. With algorithmic parallelism the sequential algorithm is split up in parts and assigned to different processors. This option will often force you to re-design the algorithm, since automatic extraction of parallelism in program code is (at this moment) practically impossible.

TNO-FEL has gained considerable experience with transputer based parallel processing over the past five years. Also neural network technology has been identified as an important form of parallel processing. Neural network technology uses massive parallelism to emulate the functioning of neurons as they are known from the human brain. Parallel processing systems have been developed in the field of image processing, signal processing (RADAR), computer generated imaging (trainers & simulators), data visualisation (Voxel processor) and Computer Vision. Two of these applications will be presented in this paper.

VISUAL SIMULATION

Simulators

There is an increasing demand in simulator systems for training purposes. The most important reason to use a simulator is to prevent physical injuries or material damages when training for hazardous situations. For example : engine problems, calamities or combat training. There are however more advantages to using a simulator instead of the 'real thing' :

-) Cost reduction
   Training hours on systems (that could otherwise earn money) are very expensive in terms of maintenance and energy consumption. Also, in many cases the complete system is not required to train personnel for a specific task.

-) Environmental reasons
   Many outdoor (military) training areas will be closed down for environmental reasons. Also, realistic training may be difficult in areas that are being used intensively (e.g. airports).

-) Better instructor supervision
   The instructor has complete control over the training conditions. Also the trainee's responses can easily be recorded and evaluated afterwards.

The simulator market is already large and will continue to grow, especially if new applications become feasible when the cost factor can be reduced :

-) driving simulators (cars, trucks, trains, tanks etc.)
-) naval simulators (ship bridge, submarines)
-) flight simulators (military, commercial)
-) design (seeing a house before it is built)
-) entertainment (arcade games)
-) education (virtual reality)

A simulator exists in general of three parts : a mock-up of the real system (e.g. the cabin of a truck or the cockpit of an airplane), the system simulation computer and the environment simulation. The mock-up has all the important controls of the real system. In some systems the mock-up resides on an hydraulic platform to simulate motion. User input from the mock-up is handled by the system simulation computer. This simulation computer runs models of the flight- or driving dynamics of all the real systems in the training scene. The environment simulation provides data from the outside world. This can be sounds, motion and images. Given the fact that the human operators derive so much information about their environment from visual input it is not surprising that visual simulation is the most important part for many trainers. The visual system computes an 'out of the window' view of the outside world. The input to the visual system is the trainee's position and orientation in the artificial world. Additional parameters are : weather conditions (sunny, cloudy or foggy), time of day and sensor type (daylight or IR).

Computer generated visuals

Visual simulation by computer has a much higher flexibility than previous solutions based for example on camera's moving over modelled environments. Artificial worlds (landscapes, roads, buildings and objects) are modelled with a polygonal description of the real world. Modifications are relatively easy and new objects can be added quickly. The visual system of a simulator must provide several features :

-) Good realism

The images must resemble the 'real world' sufficiently to provide a trainee's acceptance of the artificial world and to give him the necessary clues. For example objects should be recognised correctly and speed should be accurately estimated from motion. Image quality depends on : resolution, available colours, rendering technique and 'special effects' (weather conditions, flashes).

-) Sufficient image update rate

An impression of smooth movement must be given. This is expressed in the number of frames per second that must be computed. This number depends on the type of simulation, i.e. the speed of the objects in the scene. A flight simulator requires more frames/sec. than a driving simulator. Representative values for a driving simulator are 25 Hz.

-) Low response time

The system must quickly respond to all the trainee's actions. The delay between trainee input and system response may not exceed 100 ms to prevent the effect of 'lagging'.

The visual simulation system is very computational intensive and until recently most CGI (Computer Generated Imagery) manufacturers used dedicated hardware for their systems. This is one of the reasons that these systems are very expensive and used mainly in areas like the flight simulator and military market.

The graphics pipeline

The visual simulation process is usualy described as a pipeline of several tasks [3] (Fig. 1). The (polygonal) database of the world model is processed into an internal representation for the system by the Tree Traversal stage. The Transformation stage converts the world model coordinates into the coordinate system of the observer (i.e. the trainee). The position of moving objects is computed before they are added to the world model. Clipping removes (parts of) objects that are outside the observer's field of view. Hidden Surface Removal provides a correct visualization of objects that are (partly) obscured by others. The operation involves a sorting process of all data that is shown on the screen. The sorting can take place at an early stage in the pipeline (using for example a Binary Space Partitioning on the polygon data). Sorting may also be achieved at the very end of the process, just before drawing pixels on the screen (Z-buffering). Transformed polygonal data is Scan Converted into edge data. Edge data describes, in an iterative way, where polygons cross the display's scanlines. This data is used to 'fill in' the pixels of a scanline with the appropriate colour. The Pixel Fill stage can use several techniques to 'shade' polygon pixels on the screen, they are presented here in order of increasing realism and computational requirements : flat shading, Gouroud shading or texture mapping. Flat shading adds a single colour to each polygon surface, based on the object's colour and the average sun intensity. Smoother looking objects are provided by Gouroud shading, where polygon colours are linearly interpolated from one edge to the other. Texturing maps digital photographic material on top of polygons. This results in a very realistic rendering of certain objects (e.g. sky, trees and ground surfaces).

PARALLEL PROCESSING FOR VISUAL SYSTEMS

A parallel graphics pipeline

The arrival of inexpensive, modular, off-the-shelf parallel hardware like transputers opened up new possibilities for CGI system implementations, since these parallel systems turn CGI development from a hardware problem into a software problem. The advantages of this solution are : reduced cost, a flexible design and a scalable performance. The graphics pipeline is a natural candidate for algorithmic parallelism : the pipeline stages are directly assigned to several processors. However, data parallelism can be implemented additionally. Two basic approaches are possible : display- and object space parallelism. Display space parallelism implies that processors are assigned to a certain area of the resulting image (e.g. a number of scanlines). The load-balancing problem can be tackled by implementing a processor farm. In this construction a controller process "farms out" a new piece of work (i.e. a part of the display) to each processor in the network as soon as it has finished work on a previous part. A second approach to load-balancing is the use of scanline interleaving. In this way, processors will get an equal mix of heavily occupied and empty scanlines. Object space parallelism is based on distributing a limited part of the input data (i.e. object descriptions). This implies that each node is assigned to a section of the (visible) object database. A node will produce the contribution of the local data to the result. The complete result will be available after combining (merging) all the contributions. The advantages of this method over the previous one are : fast access to the (local) data and good load-balancing since all contributions will need the same computation time, when the data sizes are equal. Disadvantages are the overhead of distributing object data and merging the partial results.

TNO-FEL CGI systems

For about 3 years TNO-FEL has been working on image generation for simulators. TNO-FEL has designed, developed and delivered operational image generation systems based on parallel technology (Fig. 2 and 3). The work has resulted in a family of systems, based on a modular approach to transputer hardware and software (Occam [4]). The members of this family have the following features :

-) Low-end system.
This first generation system visualizes targets only, backgrounds are not computed but projected from slides. Object shading is grey-scale only. This fully transputer based system can produce 25 frames per seconds for objects of 300 polygons with ca. 20 nodes. The display hardware in this system uses only multiple link inputs (4 or 8).

-) Medium-end system
The second generation system is currently in the realization phase. It visualizes fully textured backgrounds (computed off-line) and moving targets from a stationary location with 24 bit RGB values. The systems uses T800 transputers and a dedicated communication bus for image transfer.

-) High-end system
In the near future, the arrival of the next transputer generation (T9 series) will enable an upgrade of the Medium-end system into a system where real-time motion is possible through a fully textured background.

The first TNO-FEL implementations of the graphics pipeline on a transputer system had the topology shown in Fig. 2. Timing showed that the last stage of this pipeline (pixel fill stage) is the most computationally intensive one. This explains the large number of processors for this stage. Notice that

the design uses both algorithmic- (pipeline) and data-parallelism (display space). Bearing in mind that the throughput of a pipeline is limited by the slowest element, it becomes clear that a bottle-neck exist in the transfer of images (scanlines) from the rendering- to the display- processor across four links. Furthermore, it is not possible to significantly extend the length of the pipeline, since this would involve a continuing increase of the system's response time. Later designs have used a different approach : objects are processed in parallel on several pipelines, each one operating on a roughly equal amount of polygons. This way the pipeline as a whole can achieve a better load balance.
Several problems were encountered in the application of the current generation of transputers (T8 series) for visual systems. The link bandwidth is insufficient to transfer high resolution images (1280 x 1024 pixels of 24 bit) at framerates of 25 Hz. A high speed communication system (bus) is required or alternatively the scanline data in the render-processor should be directly accessible by the display-processor (e.g. dual-port memory). Adding more processors also implies that more data (polygons etc.) must be distributed amongst them. This requires a part of the limited link bandwidth and costs significant routing overhead. Finally, the computational performance of a single transputer should be increased to the state-of-the-art, reducing the total number of processors, which will result in a smaller system and reduced overhead.
The performance, link bandwidth and message routing problems are tackled by the T9 series and therefore it will be possible to use standard T9 hardware for visual systems of higher quality and performance than is possible today. However, the need for a high performance communication bus still exists for high-end systems. New developments of visual simulation systems are therefore undertaken on the basis of the new T9000 Transputer, combined with high performance video bus systems (Fig. 3). The bus will transfer image data and the links will mainly be used for polygon distribution and control tasks. The hardware support for message routing will enable an efficient use of the links for data distribution and will simplify software development. The hardware for CGI systems is directly applicable to the (real-time) computer vision field, where the same demands are found : high computing performance and high data throughput of images.

COMPUTER VISION

Computer vision systems

Computer vision will become a very important area in the near future, both in the scientific and in the commercial sense of the word. Many applications can be envisaged if we are able to develop high performance systems at a relatively low cost. Application areas for computer vision are : industrial inspection (production and quality control), surveillance tasks (security and military) and autonomous vehicles (hazardous environments, military). Until now computer vision applications have had only limited success because of two reasons. The first reason is that no general algorithms have been developed sofar for most object identification tasks. This is partly due to a lack of understanding of the biological perception system. Secondly, the need for 'real-time' performance and the complexity of the algorithms have required expensive, dedicated hardware for vision systems.

Autonomous systems

The objective of TNO-FEL in the computer vision research is to acquire knowledge of and to develop tools for (real-time) autonomous computer vision systems. The aim is to locate, identify and track moving objects by a moving sensor platform (robot). Computer vision is used for two different but related tasks in autonomous systems :

-) Vehicle navigation and orientation

-) Object identification

Autonomous systems should be able to execute pre-programmed tasks without human interference. The system must be able to operate in unknown environments and respond to unexpected situations.

Computer vision process

The main task of a computer vision system is to analyse an image of a scene (low level representation) and translate it into a symbolic description (high level represention). This translation is a data reduction process. The extracted symbolic description (or world model) must provide all the necessary information to successfully execute the required tasks. This implies that the symbolic description must be selected to fit the application : for navigation a simple description based on geometric shapes (cubes, cylinders etc.) may be sufficient, whereas more sophisticated structures are required for identification purposes.

Several stages are distinguished in the Computer Vision process [5] (Fig. 4). The Sensor stage provides the input to the system. Many types of passive and active sensors are suitable for vision applications : visible light or IR camera's, acoustic sensors or LASER range finders. CCD camera's are the most popular sensor for computer vision since they are inexpensive and easily available. Camera's are sometimes used in pairs to attempt to resolve depth information. However, we are not convinced that this is the best approach for 3D vision. Acoustic sensors are often used for obstacle detection but can also provide depth images. A possible advantage of an acoustic sensors is that it can see through opaque media (e.g. smoke or troubled water). LASER range finders can be applied to scan an environment and directly provide a 'depth map' of the scene based on 'time-of-flight' or phase information. A depth map is also known as '2.5D image', since it can only provide distance information for visible parts of the scene. Range scanners are much more expensive than camera's but the matching and lighting problems of stereo camera's are avoided. The direct supply of depth data is the reason for our selection of range finders as the preferred sensor system.

Sensor data is mainly used for further processing by the higher levels of the vision process, however the data is also used to do some low level control. This can be compared to human reflexes, e.g. directly stop the vehicle when an obstacle is hit.

The Preprocess stage often applies standard image processing algorithms on the sensor data to improve contrast, suppress noise or correct sensor characteristics. In many cases the use of dedicated hardware is required to achieve sufficient performance for this stage. This is mainly due to the large amount of data (pixels) that must be processed.

Suitable Features like edges, lines or planes are extracted from the sensor data. These features are the primitives from which a world model is derived. Features are therefore selected depending on the type of application. Feature extraction often uses well known image processing techniques, for 2.5D sensors however further research is still required.

The Analysis stage translates the extracted features into a symbolic description or world model of the environment. The translation involves finding relations between detected features : joining edges, combining polygons. The world model is the actual output of the vision system. All information required by the supervisor (e.g. for route planning or manipulator control) is extracted from this (hierarchical) world model. The symbolic description must be robust and suitable for the required tasks. Examples of descriptors are : wireframes, polygons or voxels.

The Supervisor interpretes the world model and decides which actions must be taken next to execute the systems task. This involves : route planning, motion control and manipulator control. If insufficient data is available, the supervisor may decide to collect more information from the sensors, for example look at the scene again from another angle or with a different sensor.

Notice that CGI is in many ways symetrical to the vision process. In vision systems images are reduced to symbolic descriptions (e.g. polygons), restoring hidden surfaces and converting objects from image space to world model space. With parallel graphics systems we found a communication bottleneck at the end of the process pipeline when transfering computed image data towards the display. In computer vision the bottleneck arises at the start of the process when sensor images must be distributed among the processor nodes. The computer vision field could therefore use much of the experience build up in CGI development.

PARALLEL PROCESSING FOR COMPUTER VISION

Parallelism in computer vision

TNO-FEL are investigating the use of parallel processing technology for the computationaly intensive vision tasks. We consider parallel processing the only way to achieve real-time computer vision performance. Our research concentrates in particular on the architectural requirements for transputer based machines to be successfully applied in (semi-) autonomous vision systems. Parallelism in computer vision can be implemented in very much the same way as with CGI systems. Algorithmic parallelism can be used to implement the different stages mentioned in the previous paragraph as a pipeline of processes. The preprocessing stages will be implemented with available dedicated hardware. Data parallelism will be used for the feature extraction stages, operating on the distributed 'images'. In the last stages object processing (e.g identification and world model updating) can be parallelized.

Neural Networks

The parallel processing group at TNO-FEL has put a considerable research effort into the application of Neural Networks. The main advantages of these systems are in our opinion fault tolerance and trained behaviour. Several projects were successfully undertaken, for example aircraft identification through JEM (Jet Engine Modulation) and garbled text string recognition. Neural network techniques are considered very suitable for object identification also, since it can deal with incomplete and noisy data. It is clear that neural network systems need a high degree of connectivety (massive parallelism). The next transputer generation (T9000) with its hardware routing support is therefore expected to provide much more efficient implementations of neural networks than the T8 series.

TNO-FEL vision projects

Some of the current work in the computer vision area is part of the EC TELEMAN project. TELEMAN's strategic objective is to develop advanced tele-operators that respond to the needs of nuclear industry users and manufacturers. The aim of the 'TELEMAN 17 - UKIS' project is to design an intelligent observation and imaging system for computer assisted tele-operators. TNO-FEL participate in the data-fusion and object recognition workpackages. In this project several sensor systems are applied : stereo camera's, acoustic arrays and laser range finders (Fig. 5). TNO-FEL are developing algorithms to segment 2.5D range images into polygonal regions (Fig. 6). Local normal vectors are computed from range data and the resulting information is used to construct regions of constant orientation. A first version of the polygon extraction algorithm on artificial range data (128 x 128 pixels), requires ca. 20 sec. computation on 16 transputers. The segmented regions, described with polygons, should be merged to recombine incorrectly disconnected segments. Currently, research is conducted into an optimal (graph) representation of the relations between polygons. This symbolic description will then be used to perform object identification based on neural network techniques.

The development of computer vision algorithms is (partly) performed on a transputer system with 64 T800 processors. The system provides software configurability and runs the HELIOS Operating System [6]. HELIOS is a UNIX-like OS designed to run fully distributed on a network of processors. The distributed nature is transparent to both the user and the programs running within it, as they never need to know where a system service is located exactly. HELIOS supports C and therefore provides an easy portability of existing software. HELIOS is based on a client/server model. Available servers include : X-Windows support, Ethernet servers and file servers. The performance of a system running HELIOS is lower than what is possible with OCCAM but the T9 generation with its message routing support will probably reduce this disadavantage.
In the near future a second transputers based platform will be used, providing a high speed communication system for the distribution of sensor data to processor nodes. This hardware is also applied for the CGI work.

In our opinion, the key to success in computer vision work is a combination of basic research in the field of high-level vision (notably Neural Networks) and state-of-the-art parallel systems (transputers). Particular consideration is therefore given to the identification and exploitation of parallelism within the vision process and the problems associated with mapping suitable algorithms on a multi-processor architecture.

CONCLUSIONS

The developed systems are a successful demonstration of the performance improvement and flexibility that parallel processing can deliver. Transputers have proved to be a very powerful tool, both for research and commercial applications. Software development was greatly simplified by the clear representation and support of parallelism that CSP (as implemented in Occam and HELIOS) offers. Among other improvements, the new transputer generation will provide a better hardware implementation of the CSP model, allowing an easier software development. CGI and computer vision development should benefit more from the strong relationship between them, both in an algorithmic and an architectural sense.

REFERENCES

[1] The Transputer Databook; INMOS publication, 1989.

[2] Lippmann, R.P.; An Introduction to Computing with Neural Nets. IEEE ASSP Magazine, April 1987, pp 4-22.

[3] Foley, J.D., Van Dam, A.; Fundamentals of Interactive Computer Graphics. Addison-Wesley, 1983.

[4] C.A.R. Hoare (Ed); OCCAM 2 Reference Manual. Prentice Hall, 1988.

[5] Marr D; Vision. Freeman Press, 1982.

[6] The HELIOS Operating System. Perihelion Software, Prentice Hall, 1989.

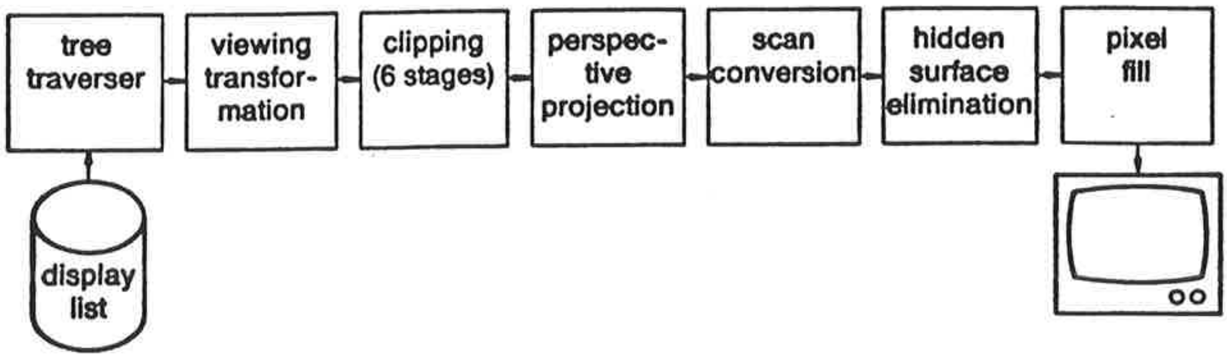

Fig. 1 The Graphics Pipeline

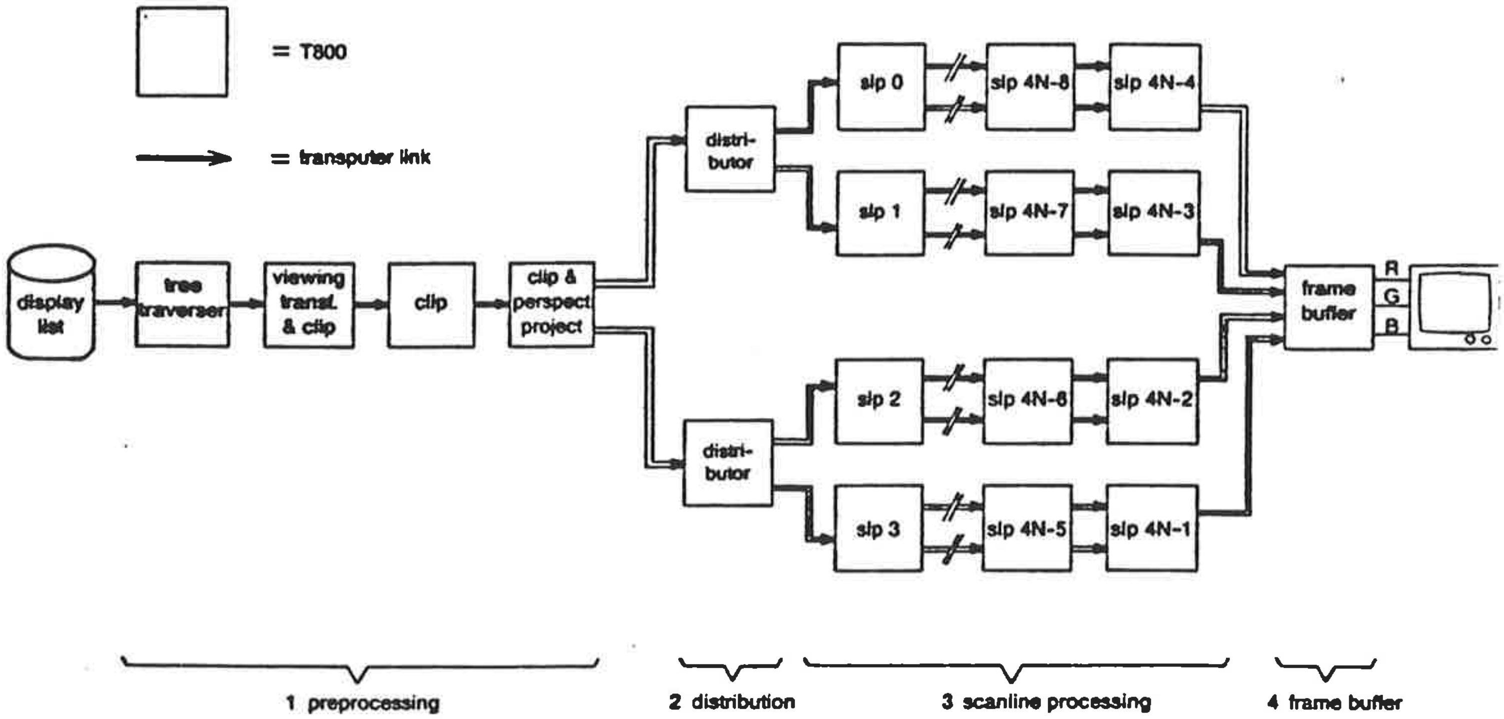

Fig. 2 First Generation Visual System

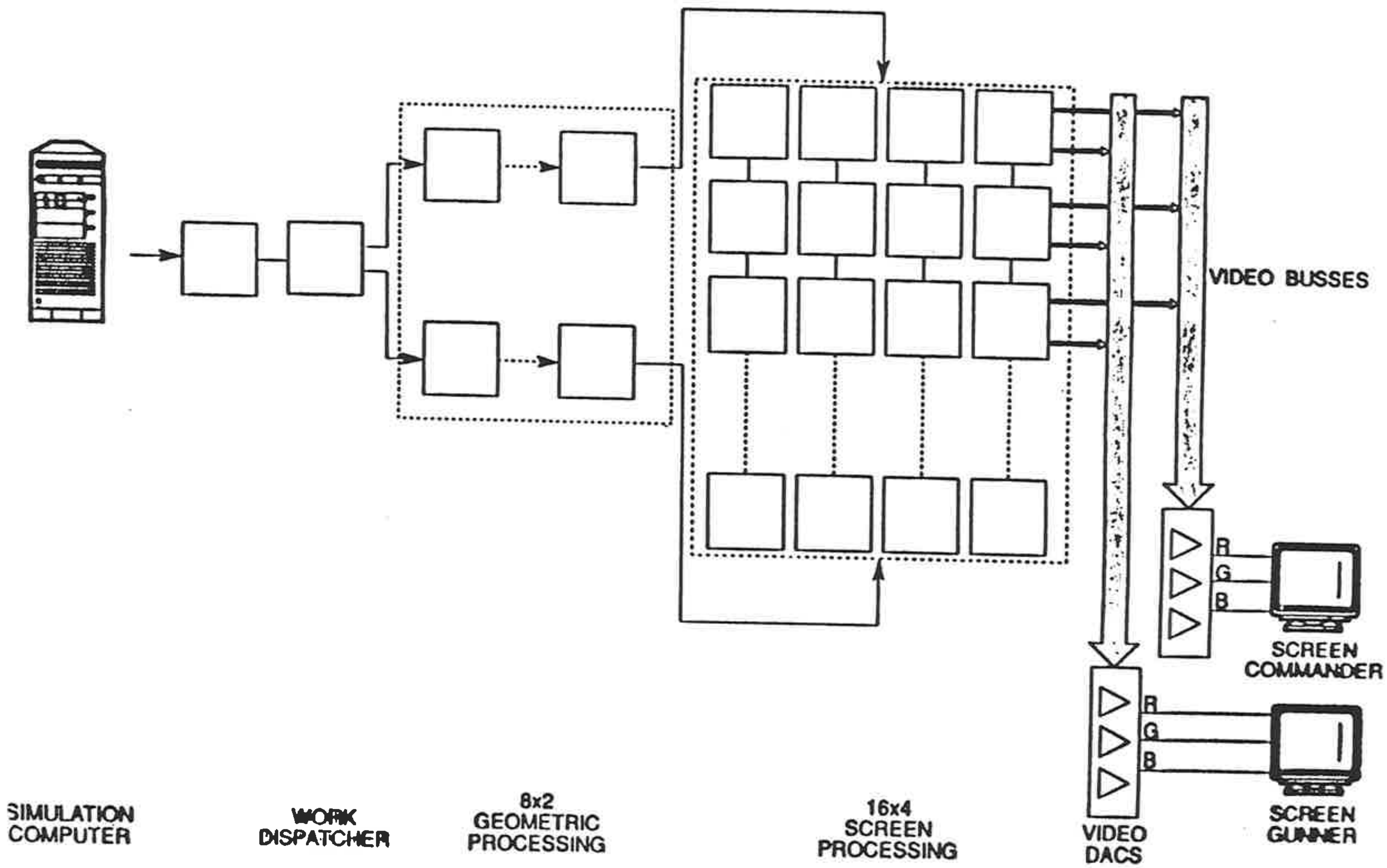

Fig. 3 Second Generation Visual System

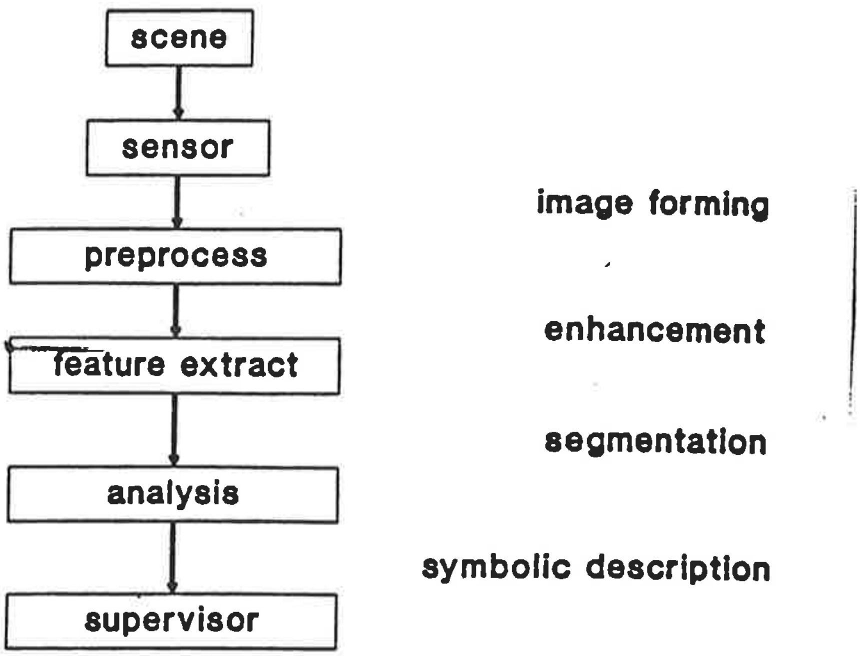

Fig. 4 The Vision Process

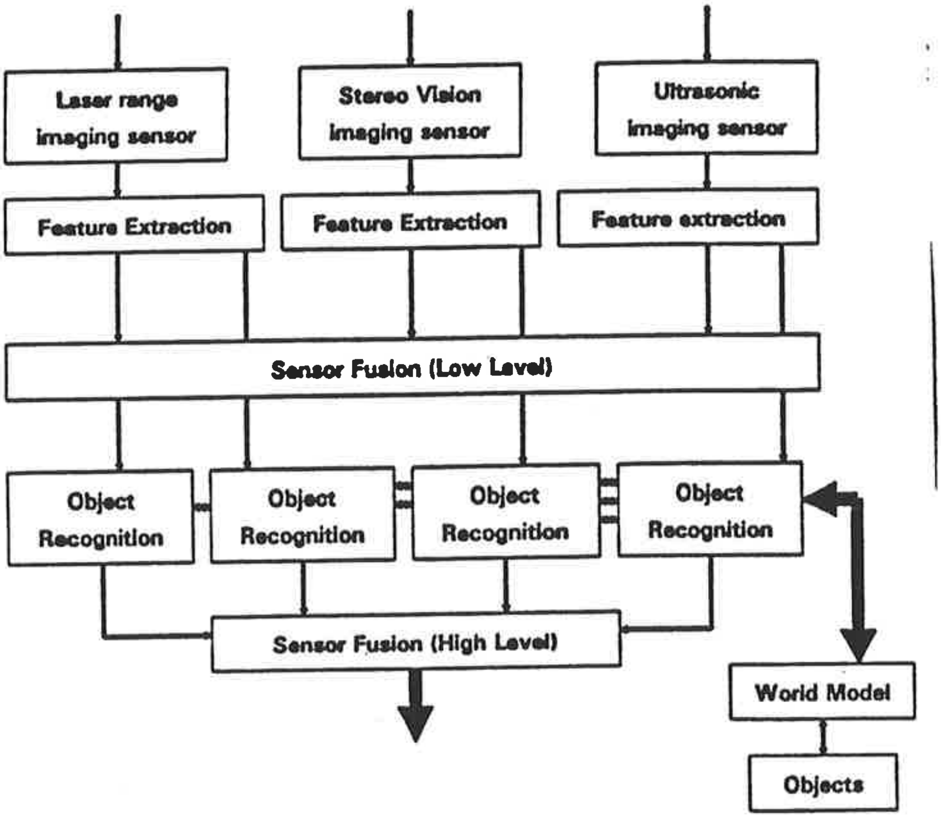

Fig. 5 UKIS Block Diagram

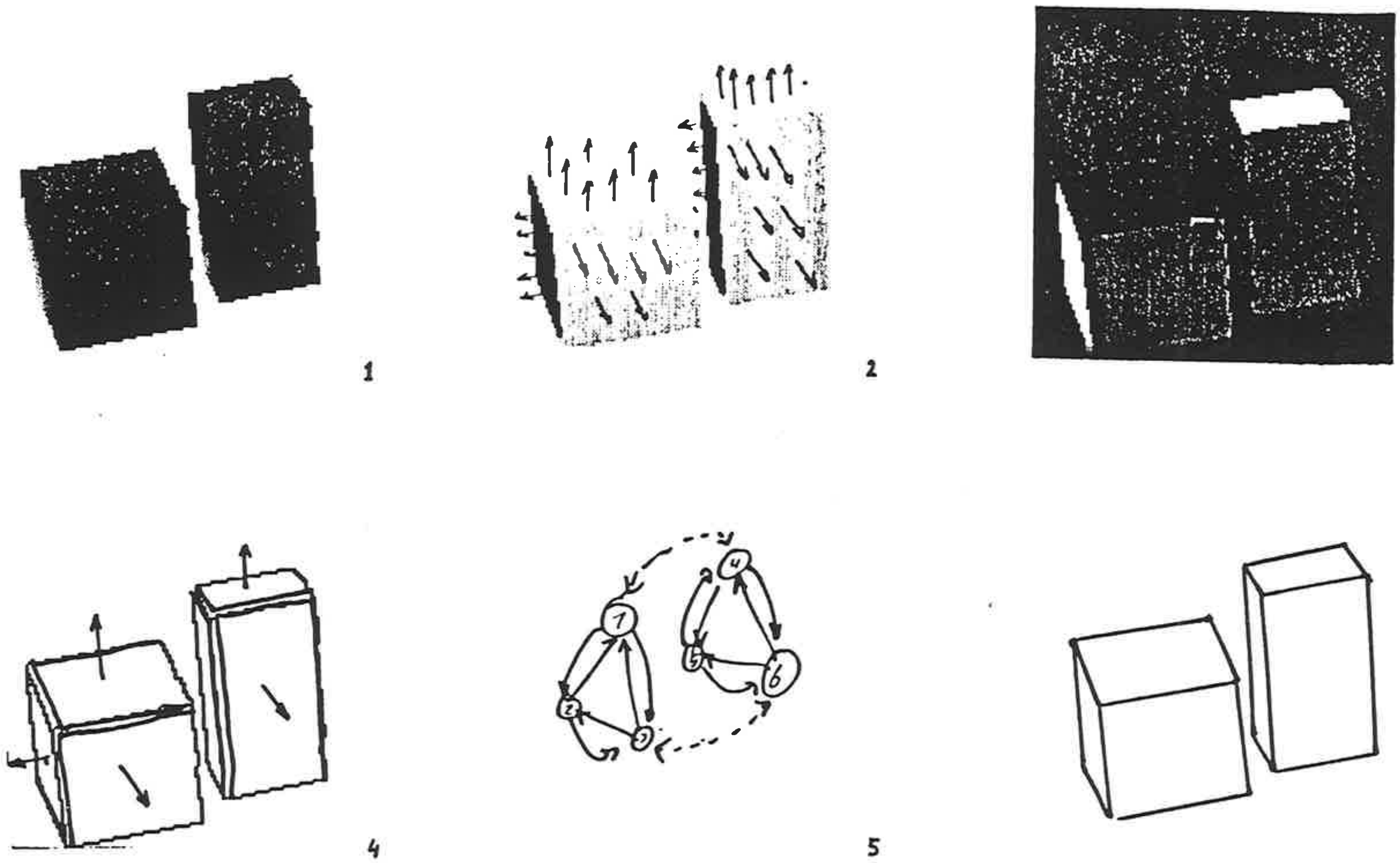

Fig. 6 Polygon Extraction From Depth Data

TRANSPUTERS *for* INDUSTRY

fourth
international
transputer
conference
and
exhibition

ANTWERP 21 NOVEMBER 1991

bira

BELGIAN INSTITUTE FOR AUTOMATIC CONTROL

in cooperation with

acm

Association for Computing Machinery