A TRANSPUTER BASED 3D-GRAPHICS SYSTEM

Klaus Alvermann
German Aerospace Research Establishment
Institute for Flight Mechanics
3300 Braunschweig
Federal Republic of Germany

ABSTRACT

The Institute for Flight Mechanics operates the flying simulators ATTAS (a wing aircraft) and ATTHeS (a helicopter), their respective ground based simulators and uses realtime and offline simulations for system identification and other purposes. Based on a parallel transputer architecture, a 3D-graphics tool for visualization and view simulation to be used with the simulations has been developed. The tool uses data received by telemetry, realtime data from a simulation, or recorded data to show the movement and orientation of an aircraft in realtime 3D-graphics. The aircraft or scene may be observed from any point of view. Placing the camera in the cockpit of the aircraft and showing the environment results in a view simulation.

The use of a parallel transputer architecture allows a modular and scalable structure, i.e. the system may be adapted to the needs of the application. By adding software modules and transputers we may include 24 bit colour, shadowing, a higher resolution, a better shading algorithm or other things which are required by an application. On the other hand we may remove transputers to get a small and cheap system if the requirements are low. A small system may consist of only 8 transputers, whereas a big system may include 50 or 60 transputers.

Key words: Transputer, Visualization, View Simulation.

INTRODUCTION

Aircraft simulations, whether realtime or offline, produce a data stream of flight mechanical state variables. It is difficult to imagine the flightpath from the (x, y, z) coordinates or the orientation of the aircraft from the three Euler angles (N, h, R). To get a general idea of the movement and orientation of the aircraft we have developed a 3D-graphics tool called 3DV (3D-visualization) to produce images from (x, y, z, N, h, R) in realtime, i.e. with an acceptable image rate (16 to 25 images per second) and a short timelag (50 to 200 milliseconds). The 3D-tool was designed such that it accepts object description data and movement data for several objects (not necessarily aircrafts) together with the data for the camera movement and orientation to produce an image of the scene.

3DV can use data coming from any source: it can be a realtime simulation, such as a hardware-in-the-loop simulation, it can be data coming from a disk originating from a mainframe offline simulation, or it can be telemetry data sent to a ground station via downlink. In the latter case the aircraft, which may be miles away, can be observed as if a chase plane with a video camera would be following it. 3DV is not restricted to show aircrafts, it can show any technical system, such as the flapping, lead-lag, and torsion motion of a (simulated) helicopter rotor [1].

3DV is implemented on a parallel architecture using transputers and is normally used with a host system (a PC or workstation) to supply the object description data and an user interface. A typical system can be seen in Figure 1. The host PC is at the right, the output monitor at the left side. In the middle the transputer box can be seen, containing the 3D-graphics network and the realtime hardware-in-the-loop simulation of a Space Shuttle model [2]. Figure 2 is a typical output showing the flying simulator ATTHeS.

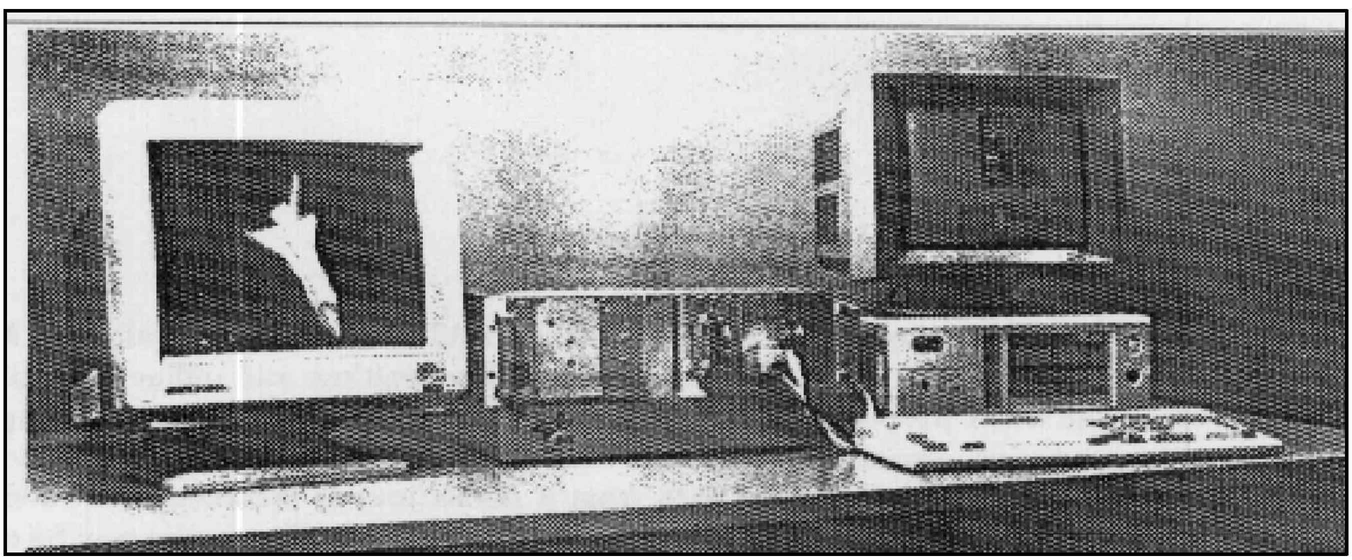

Figure 1: The complete SD-graphics system

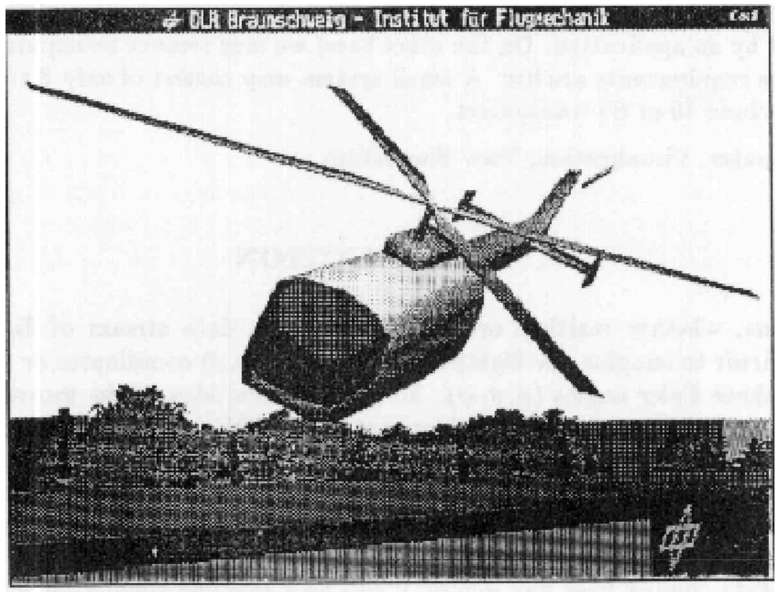

Figure 2: Visualization of the flying simulator ATTHeS

TRANSPUTERS AND OCCAM

The 3D-tool 3DV is implemented on a parallel processor architecture with distributed memory and asynchronous communication. The processors used are transputers. These are Von Neumann processors, each having its own local memory. A transputer has four bidirectional serial links. These links enable it to exchange data with other transputers. By ‘linking’ several transputers a transputer network is built-up. The single transputers in this network are working in parallel and asynchronous; there is no ‘master’ or ‘supervising operating system’. The transputers synchronize their work by exchanging data.

We use the transputer T800 which has the following technical data. The T800 has (as a single processor) a performance of 10-15 MIPS and 2.0 MFLOPS (it has a built-in floating point unit). There are 4 KByte of on-chip memory, the 3DV modules having 1 or 4 MByte of external memory. Each of the four links works with its own link engine at a data rate of 20 MBit/sec = 2.5 MByte/sec in every direction (both to and from the transputer). In this way communication along the four links and calculations in the CPU are done in parallel. If we specify parallel tasks on a single transputer, task switching is done in hardware and, therefore, very fast (a few microseconds).

Several transputers may be connected via their links in any way. This allows the mapping of the physical problem onto an appropriate network of transputers and processes. The network structure of 3DV is derived from the graphics algorithm used and the hardware available to connect transputers and a video controller.

Transputers were developed according to the concept of the programming language `occam` [3]. The basic element of `occam` is a process. A process is an assignment (i.e. `a := b+c`), a communication (i.e. `receive data from` or `send data to`) or a combination of other processes. Apart from the usual constructs (such as `if`, `while` and so on), processes may be combined in three principle ways: sequential, alternative or parallel. Sequential processes are executed one after the other, just as usual. Of alternative processes one is executed depending on communication. Parallel processes are executed in parallel. As mentioned above, even processes on one transputer may work in parallel (i.e. communication and calculation).

Processes are ideally working with their own local memory and communicate via channels. Processes are synchronized by data transfer along these channels. An output process is waiting until all specified data is sent and then is terminated. An input process is waiting until all specified data is received and then is terminated. If an input/output process has to wait, it is (of course) descheduled by the hardware of the transputer, such that other (parallel) processes on the same transputer may be executed.

Thus, the 3D-graphics system is data driven. The system takes the newest set of movement data and then produces the image. If the image is completed, again the newest data set is taken. Complex scenes may take longer, simple scene are displayed fast. In this way 3DV adapts the image rate to the complexity of the scene.

A complete program must be mapped onto the available hardware. With `occam` we place the processes on transputers and the channels on the links. This is done in the so called configuration part. If we change the hardware for 3DV (by adding more transputers to speed up or expand the network) we only have to change the configuration to place modules formerly running on one transputer onto several transputers.

THE 3D-GRAPHICS ALGORITHM

Objects for 3DV are described in their own Model Coordinate System using points, lines and faces (convex plane polygons). Several properties such as colour, shading, texture etc., are assigned to each face. A World Coordinate System is used to describe the environment. The objects are transformed into the World System by using the input movement data (x, y, z, N, h, R) (and optional scaling). Using the input data for the camera all objects are transformed into the View Reference Coordinate System, placing the camera onto the z-axis and the projection plane into the xy-plane. The objects are clipped at a front plane (just in front of the camera) and at a back plane (simulating the horizon). They are transformed into the Device Coordinate System using the input of screen resolution and window coordinates. The faces are then clipped at the boundary of the screen window and are now ready for display on the screen (see [41]). The process of transforming objects from one coordinate system to the next induces a pipeline of transputers up to this point.

To display not only a wire frame model but a shaded volume model the problem of hidden surfaces has to be solved. From the multitude of algorithms (see e.g. [5]) we have selected the z-buffer algorithms [4]. For every pixel of a face a depth value (representing the distance from the camera, which is the z-coordinate in the View Reference System) is calculated. For every pixel of the screen window a depth value is stored in a buffer (the 'z-buffer'). This buffer is initialized with 'infinity'. If a pixel is to be displayed its depth is compared to the value stored in the buffer. Only if the pixel depth is smaller, it is displayed and the depth buffer is updated.

The advantage of this approach is that objects may be defined without restrictions and may overlap during movement. The disadvantage is the need for computational power and memory. But both is available on a transputer network. Since visibility is calculated on pixel level the (screen) lines of a face can be processed in parallel. The z-buffer algorithm can be embedded into the rasterization process of the faces, which can also be parallelized on line level. This suggests the following structure for the transputer network. The data from the transformation pipeline is distributed to several transputers working in parallel. Each transputer processes every n-th line of the image (n being the number of paxallel. transputers). Upon completion of the image, the lines are sent to the video memory for display.

DESCRIPTION OF THE SYSTEM

A typical communication and process structure of 3DV is shown in Figure 3.

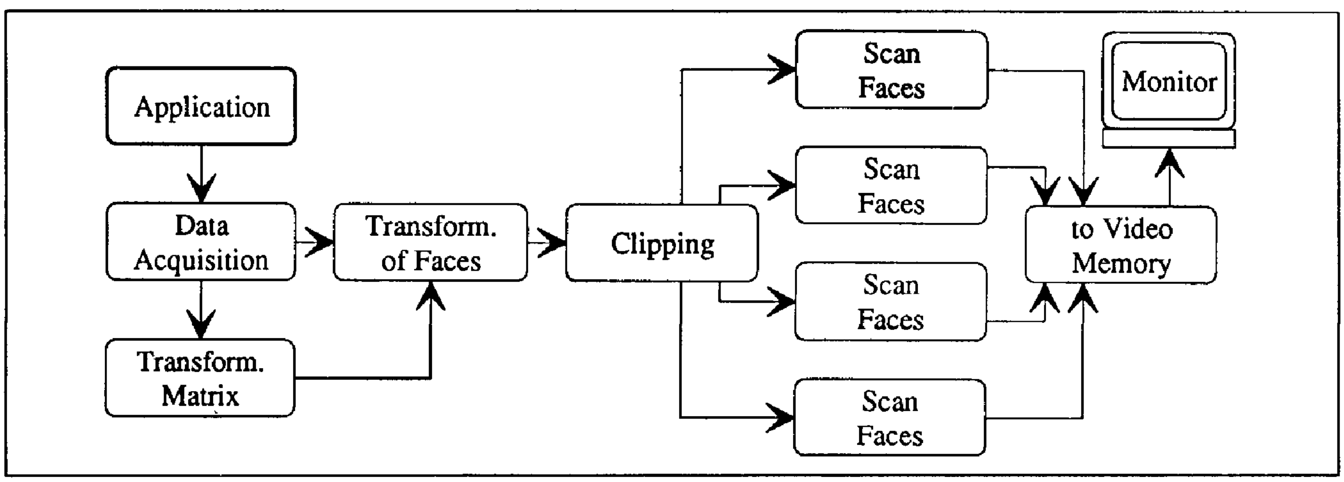

Figure 3: Structure of the 3D-graphics system

The Data Acquisition Process inputs data from the application. Object description data, movement data and camera data may come from different sources (e.g. object descriptions once from a host, movement data from the application, camera data calculated from the movement data such that the camera follows the aircraft or looks out of the cockpit). Upon request from the next process the relevant data is sent.

The next process calculates the transformation matrices for each object and sends them to the Transformator Process. There, the objects are transformed into the View Reference System, clipped at the back and front plane, transformed into Device

Coordinates and clipped at the window boundaries. These single tasks may be subprocesses which can be distributed to several transputers if necessary.

The clipped faces are distributed to several Scanner Processes which rasterize the face, calculate shading, texturing etc., and perform the z-buffer comparision. Each Scanner Process is responsible for every n-th line of the screen, where n is the number of Scanning Processes. If the image is completed, the image parts are sent into the video memory.

The image is displayed by a special transputer which uses the video memory as part of its own memory. It can receive data in parallel along its four links, therefore, the image parts are sent in parallel into the video memory. By using an add-on transputer, four more Scanner Processes can send data into the video memory in parallel.

The collection of the image parts is the main bottleneck of the system. Because of the available hardware the number of Scanner Processes is limited to eight at the moment. In spring 1992 a special transputer image data bus will be available. This communication backbone has a bandwidth of 400 MByte/sec and allows an 'unlimited' number of transputer modules access to the video memory.

The resolution and colour mode depends on the application. Normally 3DV is used with a resolution of 640 x 480 pixels and 8 bit colour. It produces 16 to 40 images/sec depending on the complexity of the scene and the number of transputers used. For the visualization of a few objects (without a landscape, but with a reference grid) 18 transputers produce more than 25 images/sec with a time delay smaller than 80 milliseconds.

APPLICATIONS

The ground based simulators for the aircraft ATTAS and the helicopter ATTHeS at the Institute are equipped with a small version of 3DV using a computational cheaper hidden surface algorithm [6]. The system consists of 8 transputers, has a screen resolution of 640 x 480 pixels and shows the simulated aircraft in a reference grid at 16 to 25 images/sec with a small time delay. Figure 4 shows an image series from this application. The system will be enhanced to a full view simulator during an European ESPRIT project using the new hardware available.

The 3DV system has also been connected to a telemetry stream to show the real aircrafts during test flights or to replay a recorded flight for analysis.

A similar small system has been used during the FALKE campaign in France [2). The 3DV system was connected to a transputer based hardware-in-the-loop simulation for the on-board computer of a 6 m Space Shuttle model. The changing of parameters and the influence of simulated maneuvers could be seen at once on the screen.

Using an interface transputer 3DV is connected to a realtime simulation of the Institutes's helicopter rotor model. The rotor may be viewed from the simulated rotating camera mounted on the rotor shaft or from the blade tip to examine changes in parameters more closely.

The main system of the Institute (19 transputers, resolution 640 x 480 pixels, 8 bit colour depth) is mainly used to replay mainframe simulations of diverse objects (airplanes, helicopters, parachutes etc.) calculated offline on a mainframe computer. If the data is available the velocity vector may be shown with the aircraft. If the velocity vector is

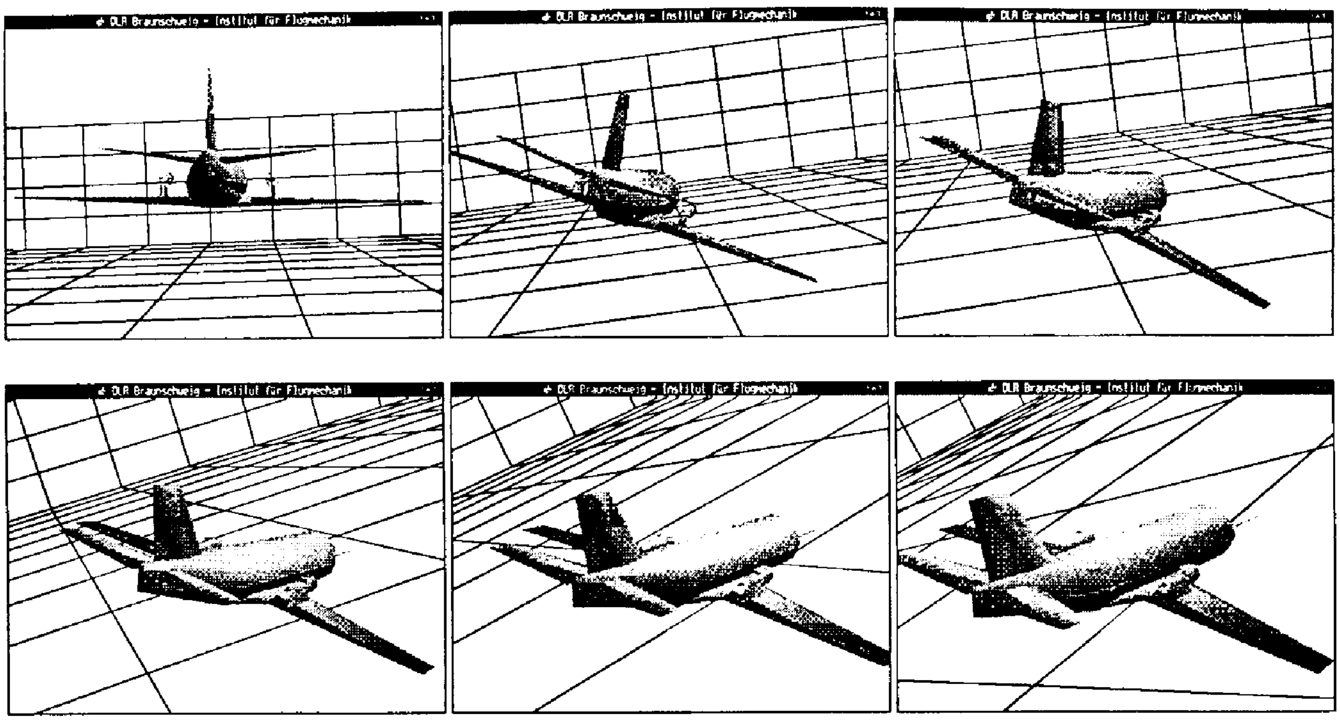

Figure 4: Image series with the flying simulator ATTAS

different from the longitudinal axis of the aircraft, two planes open up showing the angle of attack and the angle of sideslip.

3DV is also used with a landscape implementing texturing and fog simulation resulting in a simple view simulation. The image rate may come down to 4 to 10 images per second for complicated scenes.

This is mainly due to the above mentioned limit of eight Scanner Processes but will be overcome by the transputer image data bus system and/or the T9000 transputer.

REFERENCES

[1] Oertel, C.-H.: "A fast realtime simulation of a complex mechanical system on a parallel hardware architecture," 28th Telemetring Conference, October 1992.

[2] Oertel, C.-H.; Alvermann, K.; Gandert, R.; Gelhaar, B.: "A new approach to hardware-in-the-loop simulation (FALKE Shuttle)," AGARD Conference Proceedings No. 473, Computer Aided System Design and Simulation, Paper No. 44, May 1990.

[3] INMOS Limited: "occam Reference Manual," Prentice Hall, New York, 1988.

[4] Rogers, D.F.: "Procedural Elements for Computer Graphics," McGraw-Hill, New York, 1985.
[5] Sutherland, I.E.; Sproull, R.F.; Schumacker, R.A.: "A Characterization of Ten Hidden Surface Algorithms," Computing Surveys, Vol. 6, No. 1, 1974, pp-1-55.
[6) Fuchs, H,; Kedem, Z.M.; Naylor, B.F.: "On Visible Surface Generation by a priori Tree Structures," ACM Computer Graphics, Vol. 14, 1980, pp.124-133.