3D Graphics For Consumer Applications - How Realistic Does it Have to Be?

Paul Winser

Home Interactive Systems Group
Philips Research Laboratories
Redhill, Surrey
England

The design of graphics IC's for the consumer market has performance limitations imposed by the need to maintain low cost, and must be driven by consideration of the potential applications. The likely requirements for a consumer aimed real time 3D graphics system are stated in terms of performance and rendering techniques, and a research prototype of a 3D display processor is presented. The processor performs polygon drawing with smooth shading, Z buffer, and texture mapping into standard memory components. Limitations of the system and necessary image quality improvements are discussed.

1. Introduction

The state of IC technology is now sufficiently advanced where it is possible to consider producing a chip set for real display of 3D models at a cost enabling incorporation in low cost PC and consumer products. We believe this capability will be a useful enhancement to existing tasks and open up new application areas of its own. The graphics requirements for such consumer products are quite different to those of, for example, a professional CAD workstation, and the chipset architecture has to be designed accordingly.

2. System Objectives

The performance requirements for a general purpose 3D display system are difficult to precisely define when the applications vary greatly in the load placed on the different sections of the processing pipeline. Some likely application areas for the consumer market are interactive reference books and education, where image quality is the major consideration, interior and garden design, and simulation of flight or driving and other games where speed of response and smooth motion are essential. Some important characteristics of the graphics system needed to support these kinds of applications are:

Resolution: For consumer products TV or sub-TV resolution displays are used. Typically displays of between 280 x 380 and 570 x 720 are required. Video pixel rates are then 8MHz to 13.5MHz.

Refresh rate: A 25Hz screen refresh rate is desired for smooth motion especially when the user is interacting with the scene with devices such as trackerballs or joysticks, to maintain good hand to eye coordination.

Model complexity: Polygons are the simplest modelling primitives for fast rendering and the display system must be capable of processing 1000+ in order to create an interesting level of detail for typical scenes. Display performance should decline gracefully with increasingly large numbers of polygons.

Lighting: Gouraud shading is essential and specular reflection desirable for visual effect. In general the display hardware should not limit the lighting techniques available to application software, for example in number of light sources.

Texture mapping: Although expensive to implement in terms of the additional complexity of display hardware and image memory, texture mapping greatly enhances the capability of the system to create realistic images of natural scenes as well as providing distance and motion cues for fast moving applications like games. Texture mapping also provides a means of reducing the polygon count needed to model some objects, by replacing real surface detail with mapped images. A brick wall is very wasteful of display capability if individual bricks are all polygon modelled.

Antialiasing: At the low resolutions used aliasing is a severe problem since jaggies, noticable on still pictures, become very distracting on moving images. An antialiasing scheme is therefore necessary to produce smooth boundaries at polygon edges. Texture maps must also be antialiased to avoid flickering effects due to arbitrary sampling frequencies of the texture image.

3. Graphics Processor Solution

A single chip graphics processor (GP) can perform the polygon drawing function with the desired performance. This device may be loaded with polygon parameters by a processor performing the 3D geometric vertex transformations and screen clipping. This paper is not concerned with the nature of the vertex processor. The GP can directly drive a VRAM frame buffer and provide programmable video timing control. Z buffer hardware can be incorporated into the GP to achieve a high pixel writing speed, eliminating the need for parallel pixel accesses.

A number of incrementor units working in parallel calculate X, Y address, Z, and colour values for each pixel by linear interpolation, on a row by row and pixel by pixel basis. In this way Gouraud shading is performed and a piecewise linear approximation to specular highlights can be produced, which suffices provided that curved surfaces are tiled sufficiently finely. The same approach may be applied to texture mapping, although perspective distortion of the texture can still be severe and it is necessary to replace the linear interpolators with more complex elements for the texture coordinate calculation if acceptable results are to be achieved.

Conventionally the texture map is placed between the GP and the frame buffer, with texture look-up performed prior to pixel writing. The low video dot frequency, however, gives the option of placing the texture map after the frame buffer, which would then store the map coordinates U,V instead of colour values. This enables the use of slower memory devices in the texture map store since the pixel writing rate into the frame buffer will in general be higher than the video rate out. The disadvantage of this arrangement is the need for a wider frame buffer to store the texture map coordinates. The graphics processor should be able to operate with both texture map architectures, for flexible system design. Our current prototype system is organised with the texture map following the frame buffer.

4. A Prototype System

The current prototype has been implemented on a pair of VME boards in TTL. Applications run on a 68020 CPU board, and vertex processing is performed by one or more DSP chips. The GP draws trapezoidal polygon segments into a double buffered VRAM frame store of resolution 256 x 512. The frame buffer stores non-textured polygons in hue, saturation, value (HSV) format in 24 bits, and textured polygons by their map coordinates plus intensity (36 bits). Video data from the frame buffer is passed through HSV to RGB colour conversion or texture map look up accordingly.

The polygon processor has 2D linear incrementors for Z, S, V, and texture coordinates as well as X, Y address calculation. A 22 bit Z buffer is implemented in VRAM, with old Z values read from the serial port and new Z values written to the DRAM port. For each pixel these operations are pipelined so that on one clock cycle a new Z value is generated and compared with the corresponding old value read from the VRAM serial port, and on the next cycle the new value is written, qualified by the result of the comparison, to the DRAM port. Pipelining in this way allows one pixel to be written every clock cycle using page mode write access to the VRAMs, instead of the usual read-modify-write cycles of conventional Z buffers. When pixels are horizontally interleaved between two banks of VRAMs a pixel writing speed of 16MHz is easily achieved using this method.

The HSV to RGB converter is implemented by a combination of look-up table and 8 bit multipliers. A texture map of 1024 x 1024 texels may be subdivided to store a number of individual images, and each one is held as multiple resolution copies differing by linear factor 2 in order to avoid aliasing of the texture. The map resolution used is determined on a polygon by polygon basis. If the texture map is point sampled for each pixel a disturbing flicker occurs on moving objects as samples cross texel boundaries. For this reason the texture map memory is interleaved four ways to allow bilinear interpolation between four adjacent texels for each pixel. 8 bit DACs generate analogue video signals from the 24 bit RGB from the texture map and HSV to RGB converter.

5. Performance

The pixel writing rate of 16MHz allows animation at 25Hz with a screen coverage of 200% on each frame, with hidden surface removal. This is assuming that 50% of the time per frame is available for pixel writing, the remainder being consumed by set-up times for polygon parameters and VRAM transfer cycles preceding each span of horizontal pixels drawn. In practise this is sufficient for approximately 1500 polygons. The use of VRAMs in the frame buffer allows a fast screen clear time of 200μs.

6. Limitations And Future Development

The most severe defect of the system at present is the lack of antialiasing, which is difficult to perform with Z buffer hidden surface removal since the correct background colour for mixing at the edge of polygons is unknown due to the random order of arrival at the display processor. The mixing may be performed once all polygons have been drawn, and such a scheme looks promising for the future.

Texture mapping has several problems: switching between map resolution copies causes "popping" in the level of detail, which may be eliminated by trilinear interpolation between two sets of four pixels in adjacent map resolutions. Solving the perspective distortion problem will require considerably more complex processing to compute the non-linear interpolation function of texture coordinates across polygons. Correct computation requires divisions per pixel which is undesirable given the word length involved, but approximation methods may prove to be acceptable.