1 Issues and Directions for Graphics Hardware Accelerators

Kurt Akeley

1.1 Introduction

Hello, it's a pleasure to be here. I was introduced to graphics by professor James Clark at Stanford University during the summer of 1981. I didn't see much of Jim that summer, however, as he was very busy completing the development of the first Geometry Engine integrated circuit. During some of his little spare time he guided me in the design of an NMOS-based framebuffer controller which could serve as a back-end to a pipe of Geometry Engines in a complete graphics system. While my conceptual framebuffer design was never implemented, Jim asked my to join him and several others in a venture based on the Geometry Engine technology. With Jim in the lead this group founded Silicon Graphics in the summer of 1982, having begun development of a Geometry Engine-based graphics system in the fall of 1981. Thus this talk roughly commemorates my tenth anniversary in the field of graphics.

During my ten years I've watched first hand the tremendous growth in both computer and graphics capability. Processors shipped by Silicon Graphics during that period have improved from roughly 1/4 MIP performance (early 68000) to over 250 MIP performance (8 parallel R3000), a ratio of 1000 to 1. Raw graphics performance has increased at an even greater pace, from a few hundred Z-buffered polygons per second in our first machine to over a million in the current offering. And there's no end in sight, of course!

The remainder of this talk is a series of brief technical observations, followed by a personal conclusion.

1.2 Hardware Generations

Though I've been a witness to the 1000x performance increases of the past decade, as a practical engineer/tactician I don't consider myself to be well qualified to make predictions far into the future. So I won't try to see beyond the next significant step in the future of hardware-based graphics accelerators. Here I predict that a third generation of high-end graphics accelerator will become prevalent in the early '90s. Accelerators of this new generation will be tuned to draw texture mapped, antialiased polygons, rather than the Gouraud shaded, Z-buffered polygons of second generation machines, or the flat-shaded lines of first generation machines. Third generation accelerators will not, however, be fundamentally designed to accelerate either ray tracing or radiosity calculations. Thus the quality and capability of their rendering will increase, but the fundamental paradigm will not.

Advances in texture mapping capability will be more significant than those in antialiasing, because texture mapping not only increases image quality, as does antialiasing, but also opens entirely new application possibilities. It is through hardware accelerated

texture mapping capability that the heretofore separate fields of image processing and geometric graphics will merge. For example, classical image processing operations such as warping and distortion correction can be accomplished quickly and efficiently as a simple texture mapping operation. Texture mapping also allows polygon accelerators to handle new image composition algorithms such as volume rendering without need for "special purpose" hardware. (In other words, with hardware that is tuned for traditional polygon rendering.)

1.3 Parallelism

The very notion of a hardware accelerator implies parallelism, as the accelerator itself operates in parallel with its host CPU system. Historically high-end accelerators have been implemented with substantial internal parallelism as well. The most significant trend in the organization of the processors in graphics accelerators is from long pipelines of similar processors toward much shorter, truly parallel systems. This trend is driven both by the ability to implement true parallel systems, and the need to implement such systems. The ability derives from the availability of inexpensive CPU components that included on-chip cache memory, thus yielding raw processing power at reasonable code-storage cost and board real estate. The need derives from the more complex algorithms required to render advanced primitives, such as texture mapped, antialiased, smooth-shaded, Z-buffered polygons. These more complex algorithms are inherently difficult to code-balance on long pipeline machines, both as a direct result of algorithm complexity, and of the user-specified options that that complexity allows.

A secondary trend is that of increasing emphasis on per-pixel computations, as compared to geometric, or per-vertex calculations. This trend again follows directly from the requirements of second and third-generation graphics systems. While smooth shading, Z-buffering, texture mapping, and antialiasing all increase the per-vertex calculation requirements, they have a huge effect on per-pixel computations, moving them from almost nothing (flat-shaded, no Z-buffer) to very complex (trilinear texture sampling, for example). The Pixel Planes project at the University of North Carolina, Chapel Hill, was a harbinger of this trend.

1.4 What Drives Development?

Silicon Graphics began life in 1982 as a graphics-technology company, using full-custom VLSI design capabilities to build superior graphics systems. By 1986, though its products were doing well in the market place, it was in trouble internally. Integrated circuits were being designed for graphics systems that had not yet been designed themselves, because the development cycle for the integrated circuits, at up to 2 years, was much longer than that of the corresponding systems. As a result, when systems using these integrated circuits were designed, it was found that the circuits did not implement the correct functionality at the system level. To correct for this mismatch, Silicon Graphics began to design semi-custom integrated circuits rather than full custom circuits. Because the semi-custom

circuit development time nicely matched the system development cycle, integrated circuits were again specified by the systems architects, rather than by circuit implementors. The company had evolved from a graphics-technology company into a graphics-systems company, refocusing itself on systems application of appropriate technology, rather than compulsory use of the most advanced technology.

Today Silicon Graphics is both a graphics-systems company and a computer-systems company. The CPU side of the design process emphasizes technology over architecture, because there is relatively little flexibility in CPU architecture. Graphics design, on the other hand, continues to be driven by architecture rather than by technology. This is possible because graphics hardware, unlike CPU hardware, is "protected" from application programmers by a layer of software (the Iris GL). Thus graphics architecture can be changed substantially with no apparent change to programs or programmers.

1.5 What Belongs in a Graphics Accelerator?

As general-purpose processors continue to both run faster and truly be more general (RISC rather than CISC) it is increasingly difficult to justify special-purpose graphics accelerators. I believe that this will drive graphics accelerators to be very special-purpose, because a "general-purpose" accelerator will compete badly with general-purpose processors. Thus it will continue to make a great deal of sense to accelerate simple, well-defined, frequently used operations such as pixel blending and pixel depth comparison, but it will not make sense to accelerate NURBS surface tesselation, for example.

I offer disk controllers as an analogy. Though there was speculation for many years that disk controllers for UNIX machines would eventually understand and operate on UNIX i-nodes (internal operating system structures), this hasn't happened. Rather, disk controllers continue to handle the simple, well-defined tasks of signal conversion, head control, error correction, and (sometimes) data caching. Simply put, I believe that good graphics accelerators are to graphics what disc controllers are to data storage.

1.6 Personal

Silicon Graphics is the only company I have worked for in my professional career, and my ten years there are by far the longest period of my life spent at any one institution. So I asked myself, as I was preparing for this talk, why I had stayed so long, and have never even thought of leaving. My answer is two things: the people and the creative opportunities. Though I have not had the experience of working at any other company, I can't imagine that a group of better, more driven, more enthusiastic, or more fun engineers exists anywhere. The resulting job environment would be reason enough to stay. The fact that graphics architecture design allows such flexibility and creativity is icing on the cake, so to speak. At Silicon Graphics we have made much of the resulting creative opportunities, and have had the satisfaction of seeing customers put the resulting machines to an amazing variety of uses. Thanks you for having me. I look forward to your presentations and to interacting with you.