ABSTRACT

Disruptive Technologies in Computer Graphics: Past, Present, and Future

Donald P. Greenberg

The history and famous landmarks of computer graphics hardware are well known. Starting with Ivan Sutherland's Sketchpad system in the early 1960's, the first generation of computer graphics hardware consisted of calligraphic (vector) displays capable of drawing complex three-dimensional wireframe models at interactive rates. In the early 1970's expensive color frame buffers with the capability for displaying static color images were introduced. Although more and more intelligence was added to these frame buffers, Jim Clark's geometry engine and the first graphics workstations were not introduced until the 1980's. During the 1970's, only the very costly and specialized hardware used for military and aerospace simulations was capable of real-time surface color display.

By the 1990's, very high-end graphics servers evolved, but the computer graphics hardware industry previously occupied by numerous workstation vendors rapidly became dominated by the use of personal computers with high-performance graphics accelerator boards. Today we can purchase even lower cost systems from game manufacturers with startling real-time graphics capabilities. The differentiation between the high-end and low-end systems is now getting more difficult to perceive.

From an algorithmic point of view, at least with respect to rendering, much research of the first decade was devoted towards solving the visible surface or hidden surface problem. The Gouraud and Phong lighting algorithms were introduced in the early 1970's. The first scan-line algorithms and the z-buffer approach as well as texturing algorithms quickly followed. The computational requirements for global illumination were too excessive to attempt. It was not until 1979 that Whitted's ray tracing algorithm was published, and Cornell's first radiosity images appeared in 1984. Much of the research during the following decade concentrated on light reflection models, physically-based rendering, and stochastic methods to accurately simulate light interaction. Today we can produce accurate photorealistic images of startling quality, complex scenes with textures, shadows, shading and all of the subtle effects of global inter-reflections. Because the modeling tasks are so difficult and because image acquisition techniques have become so commonplace, these object-based algorithms are now being supplemented or replaced by image-based rendering algorithms, even further improving the image quality.

During this same forty-year time period, computing environments have also changed dramatically. From the early 1960's to the mid-1970's the information technology industry was dominated by batch computing and time-sharing systems. Starting at research laboratories and universities, the next decade was marked by distributed

computing based on the introduction and rapid acceptance of the mini-computer. With the advent of microprocessors and Ethernet during the past fifteen years, the environments have gradually transformed to environments of networked personal computers. Today, we are dependent on open systems consisting of clients and servers and the use of parallel computing, and as devices become smaller, we are rapidly moving into the decade of information appliances.

At each stage of this evolutionary process, more powerful, more efficient, more compact, and cheaper devices have replaced the functionality and performance of the previous era. Perhaps just as significantly, the leading manufacturers in one era did not recognize the potential impact of the new technologies of the following era. Note the change in the dominant suppliers as the mainframes were replaced by mini-computers, or the minis were replaced by microprocessor-based personal computers. The same phenomena can be observed in the mass storage sectors. As noted by Clayton Christiansen in his book, “The Innovators Dilemma”, these are examples of a typical “disruptive technology”.

It is interesting to examine the progress, the change, and the disruptions in computer graphics, not just by identifying the historical landmarks by themselves, but within the context of the available computer technologies, as well as the cost structures associated with those technologies. This evaluation can not only explain the changes of the past, but perhaps help in recognizing the changes which will occur in the future.

Although actual comparisons are much more complicated, I have chosen four major criteria for context and comparisons: memory capacity, processing power, bandwidth, and display resolution.

For each of these independent variables, it is important to consider not only the absolute performance or capabilities, but also the economic efficiencies in terms of unit costs (e.g. Megabytes per dollar for memory, MIPS or FLOPS per dollar for processing). When examined in this context, the evolutionary changes in computer graphics are easily explained.

Since the improvements in each of these four parameters occur at different rates, at any point in time, the lack of performance in one category creates the current impediment. The bottlenecks then become cyclical, explaining, in the words of Ivan Sutherland, “the wheel of reincarnation”.

To clarify my concepts, it is helpful to understand the original optimization procedures applied to manufacturing processes, building construction, etc. Critical path methods (CPM) consist of first finding the longest path in terms of time through a system. Without violating any of the precedent relationships, optimal solutions next find the most cost-effective means to shorten this path. For example, in a construction process, where does one assign additional labor to most cost-effectively truncate the total time to completion?

The analogy is similar to the primary goal in the design of computer graphics systems. How do we compute the image in the shortest time, or in the graphics pipeline, create the most complex images in real time? Although graphics hardware designers have been balancing their pipelines for each changing technology, it should be noted that the models are also changing. The first pipelines were balanced for 1000 pixel polygons, but we are now approaching 10 pixels per polygon. Should the system stay the same?

With an increase in memory density of five orders of magnitude, an increase in processing capacity of four orders of magnitude and a bandwidth increase of three orders of magnitude, all occurring at different rates and times, the systems and architectures change. Perhaps, more importantly, the costs efficiencies of the memory, processing, and bandwidth components further exaggerate the rates of change. Only display resolution has remained relatively constant with less than an order of magnitude increase.

But the graphics industry is technology driven, and we can with some degree of confidence predict the change in technology. How will this effect the future graphics hardware and algorithms?

With free memory, and ample processing power, the implications include different architectures and buffering schemes, an increase in image based rendering, and greater emphasis on perceptual approaches to reduce computational complexity. Display resolution will vastly increase, displays will be omnipresent and ultimately real-time, pixel-based, and global illumination algorithms will replace the standard graphics pipeline.