VISUALIZE Workstation Graphics for Windows NT

By Ken Severson
HP Workstation System Lab

BACKGROUND

HP set the standard for performance and functionality for the Microsoft Windows NT-based technical workstations when it introduced systems with VISUALIZE-fx4 graphics for Windows NT in 1997. In 1998, HP raised the bar with the VISUALIZE-fx6+, designed specifically for optimal performance in Intel/Microsoft based systems. Now, HP has broadened the offerings with a family of products for NT with the VISUALIZE-fx4+ and soon the VISUALIZE-fx2+.

The HP VISUALIZE fx+ family of products take full advantage of the latest technology to accelerate applications that use the OpenGL graphics API, achieving outstanding performance with an industry-leading feature set. It is a second-generation product family, designed specifically to take advantage of the latest in processor and system design technology.

The following sections focus is on the capabilities of the graphics, describing the unique features, strengths and technical information on the performance differences between the ultimate performance of the VISUALIZE-fx6+ and the lower cost VISUALIZE-fx4+ and VISUALIZE-fx2+, and how the differences can be expected to affect application performance.

VISUALIZE-fx+ Graphics - Features

Let's refer to the family ranging from the VISUALIZE-fx6+ to the VISUALIZE-fx2+ as the fx+ series family, and abbreviate its members as fx6+, fx4+, and fx2+. For compatibility and ease of application certification, the same features are supported across the product line.

Key 3D Features:

- AGP2X (133MHz) support, where AGP is the Advanced Graphics Port
- AGP DMA (Direct Memory Access) support
- OpenGL 1.1 compliance
  - Hardware acceleration of Key OpenGL extensions
  - Most OpenGL 1.2 features are supported as extensions
- 18MB SGRAM frame buffer memory:
  - 24-bit true color, double buffered
  - 24-bit Z-Buffer
  - 8-bit overlay planes
  - 4-bit stencil and 5-bit Windows ID
- Optional Texture acceleration and dedicated texture memory
- Full set of OpenGL features accelerated in hardware:
  - Hardware Accumulation Buffer (Requires driver version 1.12 or later) for advanced features:
    - full-scene anti-aliasing
    - motion blur
    - multi-pass rendering algorithms
  - Anti-aliasing for vectors and points
  - Gouraud shading (smooth shading)
  - Alpha blending for transparency
  - Fog/depth cueing
- Optional hardware texture acceleration:
  - Dedicated texture acceleration processors
  - Dedicated texture SDRAM memory
  - Point-sampled, bilinear, and trilinear MIP mapping
  - 3D texture mapping
  - Shadow texture mapping
- HP Color Recovery technology for double buffering at 1600x1200 resolution
- Hardware acceleration of occlusion culling
- Stereo vision support, with refresh rates up to 120 Hz (effectively, 60Hz each for the left and right eye)

Although the fx+ series was primarily designed for OpenGL features and performance, 2d performance and features are important in the design. Gone are the days when to get 3D graphics performance, you needed to compromise the graphics performance for word processors, spreadsheets, etc. The fx+ series also has hardware support for Microsoft's DirectDraw interface, which is useful for accelerating viewing video clips for computer based training or as part of web browsing.

VISUALIZE Graphics Architecture

HP's VISUALIZE graphics accelerators are designed with a scalable architecture that allows us to build a compatible family of products from the same building blocks. The result is a range of price and performance points that can address a range of customer needs. Note that there are 2 geometry pipelines per geometry chip, e.g., the VISUALIZE-fx6+ has 3 geometry chips, or 6 geometry pipelines.

Table 1. The VISUALIZE-fx+ Graphics Family

	fx6+	fx4+	fx2+
Geometry Pipelines	6	4	2
Host Interface Chip	1	1	1
Rasterization Chips	2	2	2
RamDac Chip	1	1	1
Frame buffer memory	18MB	18MB	18MB
Texture Accelerators Chips (Optional)	2	1	1
Texture memory (Optional)	32MB	16MB	16MB

The heart of the geometry engine is based on customized silicon, using floating point technology leveraged from HP's PA-RISC, long a leader in floating point performance.

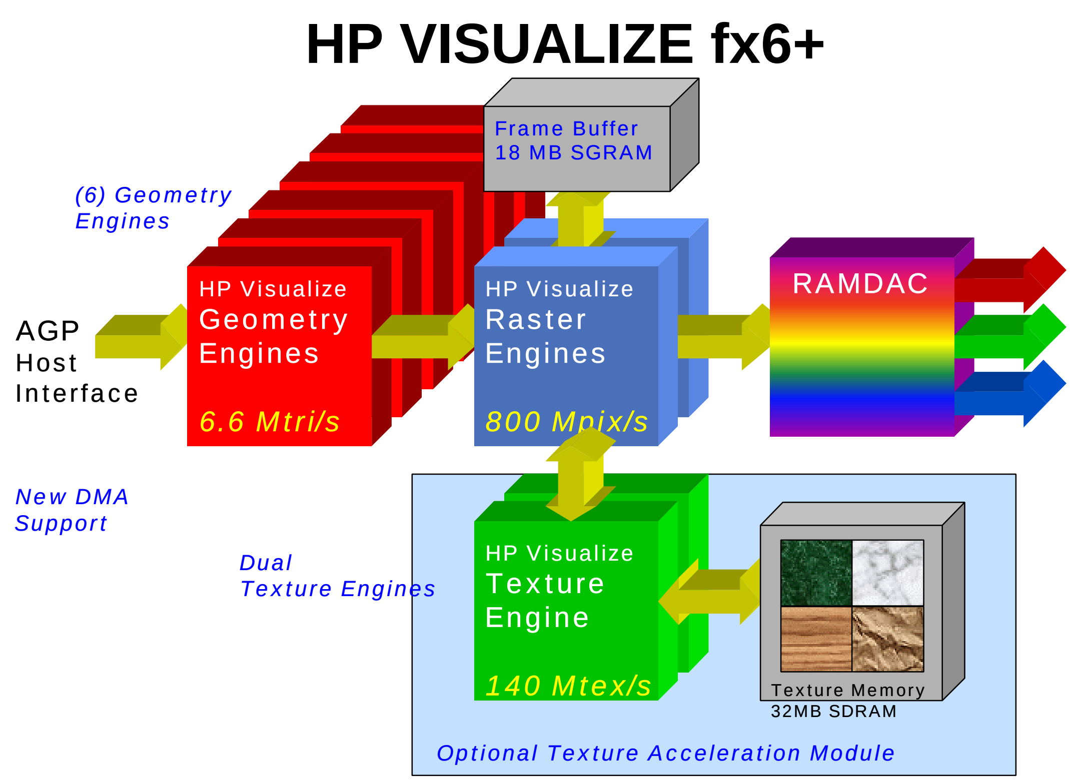


Figure 1. Block diagram of the VISUALIZE-fx6+.

If an application certifies the fx6+ graphics, it is guaranteed to work with the fx4+ and fx2+. The family is supported by the same set of drivers. (For NT4.0, use the version 1.10 driver or later). The driver detects the configuration at initialization, and the difference between the fx4+ and fx2+ is handled by literally 1 line of code in the driver. The task of dividing up the work between the "downstream" chips is a hardware function of the Host Interface chip.

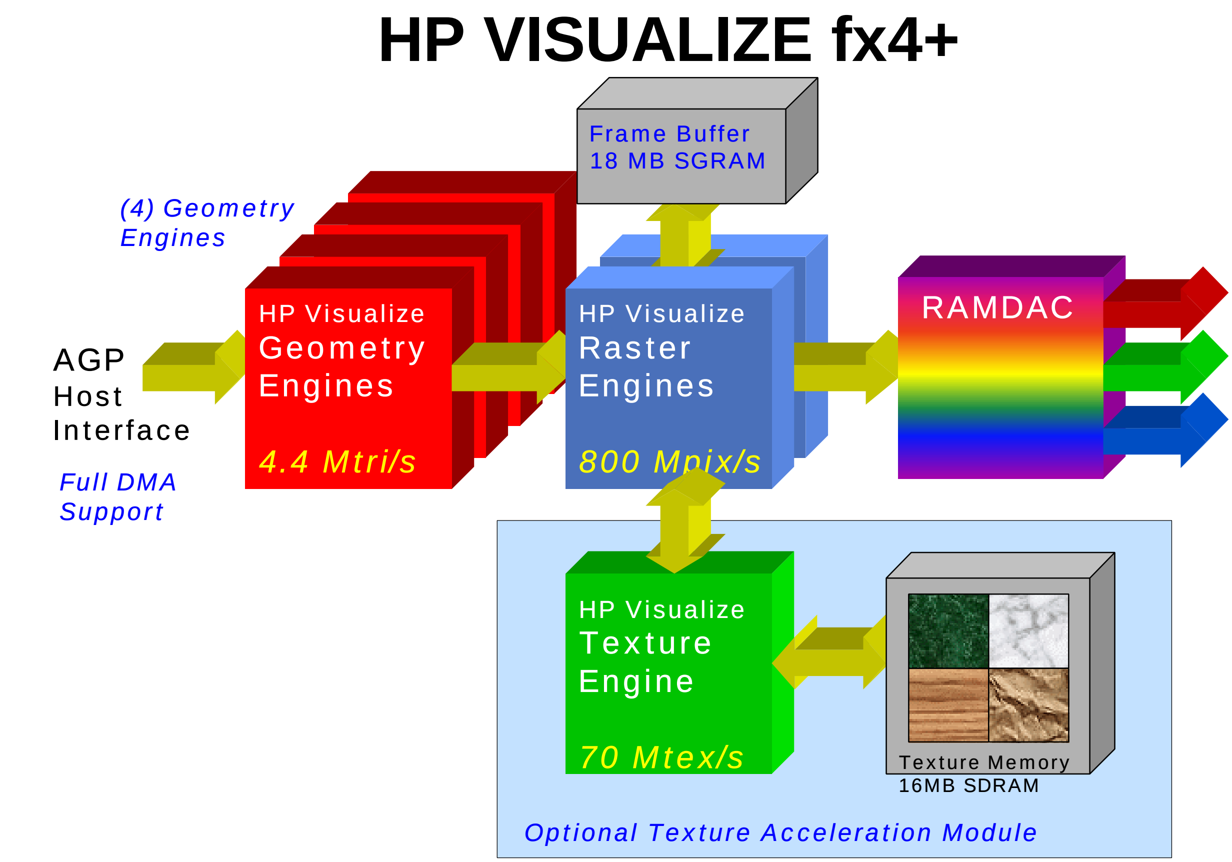


Figure 2. Block diagram of the VISUALIZE-fx4+. The VISUALIZE-fx2+ only differs by reducing the number of geometry pipelines to 2, with a theoretical maximum of 2.2 Million Triangles/second

Although a similar naming convention is used, the VISUALIZE graphics products for the Windows NT environment are different implementations than for the Unix Workstation systems. The underlying architecture is the same. In particular, the Host Interface chip and the Rasterization chips are different in the Personal Workstation systems than in the Unix Workstation systems. In addition, the Windows products include VGA graphics for Windows compatibility when booting and for full-screen DOS mode.

In the block diagrams in Figure 1 and Figure 2, the theoretical maximum throughputs of the chips are shown. Actual throughput measurements and the predicted effects on application performance are addressed below.

Display List or Immediate Mode with OpenGL

There are 2 fundamental types of rendering with OpenGL: Display List and Immediate Mode. Many applications use a mixture, but most are predominately one or the other.

In immediate mode, the application calls OpenGL to draw a primitive (e.g., line or polygon) or modify the hardware state (e.g., line color or a transformation matrix). The operation is performed, for all intents, “immediately,” and the driver does not keep a copy of the primitives.

In display lists, the application tells OpenGL to create a copy of the data that it is sending. The creation of the list is called “compiling” the display list, and the drawing from the display list is known as “executing” the display list. Because the OpenGL driver can store the information in a format that is the most convenient for the hardware, the graphics throughput for display lists is almost always higher. Display lists are very effective when the same data is to be displayed multiple times. Good examples are an architectural walk-through of a building or a fly-through of a complex mechanical assembly.

Nevertheless, there are compelling reasons for some applications to use immediate mode. In general, immediate mode is an easier programming paradigm for an application programmer.

For a large model, the display list itself could use many megabytes and is a duplicate of the model already in the application controlled memory. So an immediate mode application may require less memory than display list, and this could be the difference between being able to work on the model at all in a system with a maximum memory configuration of 2 GB.

Immediate mode is useful if the data changes significantly from frame to frame – for example, when displaying time-varying scientific data.

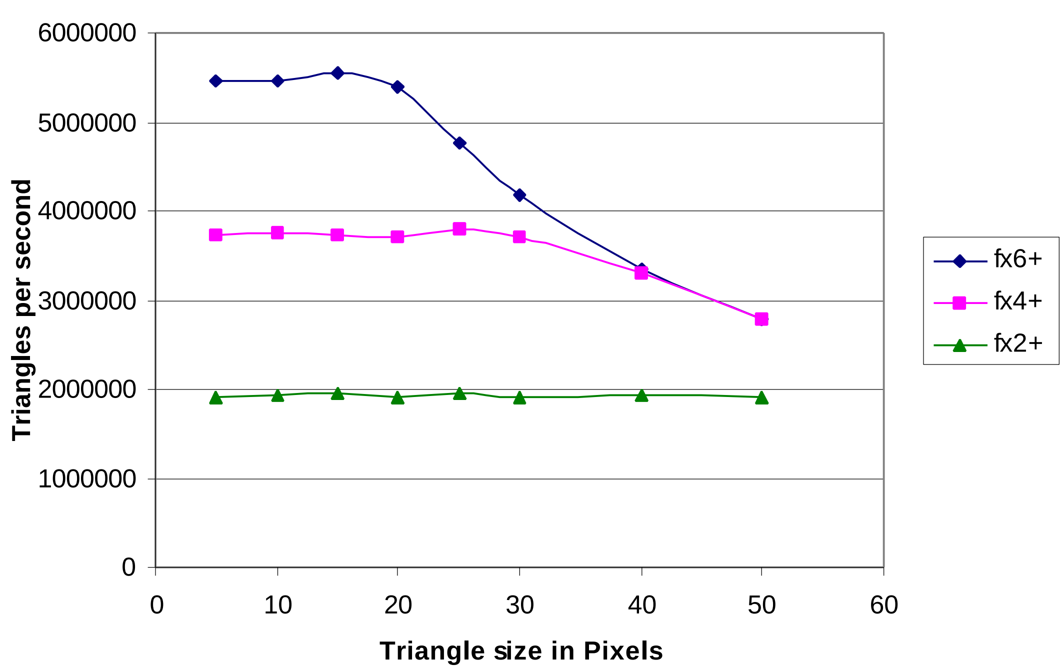
Examples of applications that run on Windows NT that are primarily Display List include Unigraphics and SDRC's IDEAS suite of applications. Examples of immediate mode applications are PTC Pro/E, Kinetix 3D Studio Max, Avid SoftImage, and Alias/Wavefront Maya. The same company may produce applications with different rendering characteristics: while PTC Pro/E is primarily immediate mode, PTC Pro/Designer is a display list application.

Performance Characteristics of the VISUALIZE fx+ Family

Since the feature set is the same, the distinguishing characteristic between members of the family is performance in two areas: how much geometry acceleration and how much texture acceleration.

To illustrate the differences, the following measurements were taken on a system with a 450MHz Xeon processor and the Intel 440GX chipset, using a tool called "glperf" created by the OpenGL Performance Characterization (OPC) project.

Figure 3. Triangle Strip Display List Performance



In application workloads that we have analyzed, most average triangle sizes are less than 25-pixels. While most data sets have large polygons as well, the small average size is due to approximating curved surfaces with a set of triangles.

In Figure 3, the power of the fx6+ is evident, showing the ability to achieve over 5.5 Million triangles per second by using OpenGL's triangle strip primitive. The fx4+ achieves 3.8 Mtri/s and the fx2+ achieves 1.9 Mtri/s. The triangles are smooth shaded (Gouraud), with 1 light source. As the size of the triangle increases, the performance of the Rasterization chips begins to dominate, resulting in a lower effective triangle throughput. The immediate mode performance (Figure 4) shows a similar relationship between the members of the family, but with lower peak triangle performance due to the per-call overhead of immediate mode. The triangle size at which rasterization speed begins to be a factor is for a larger size.

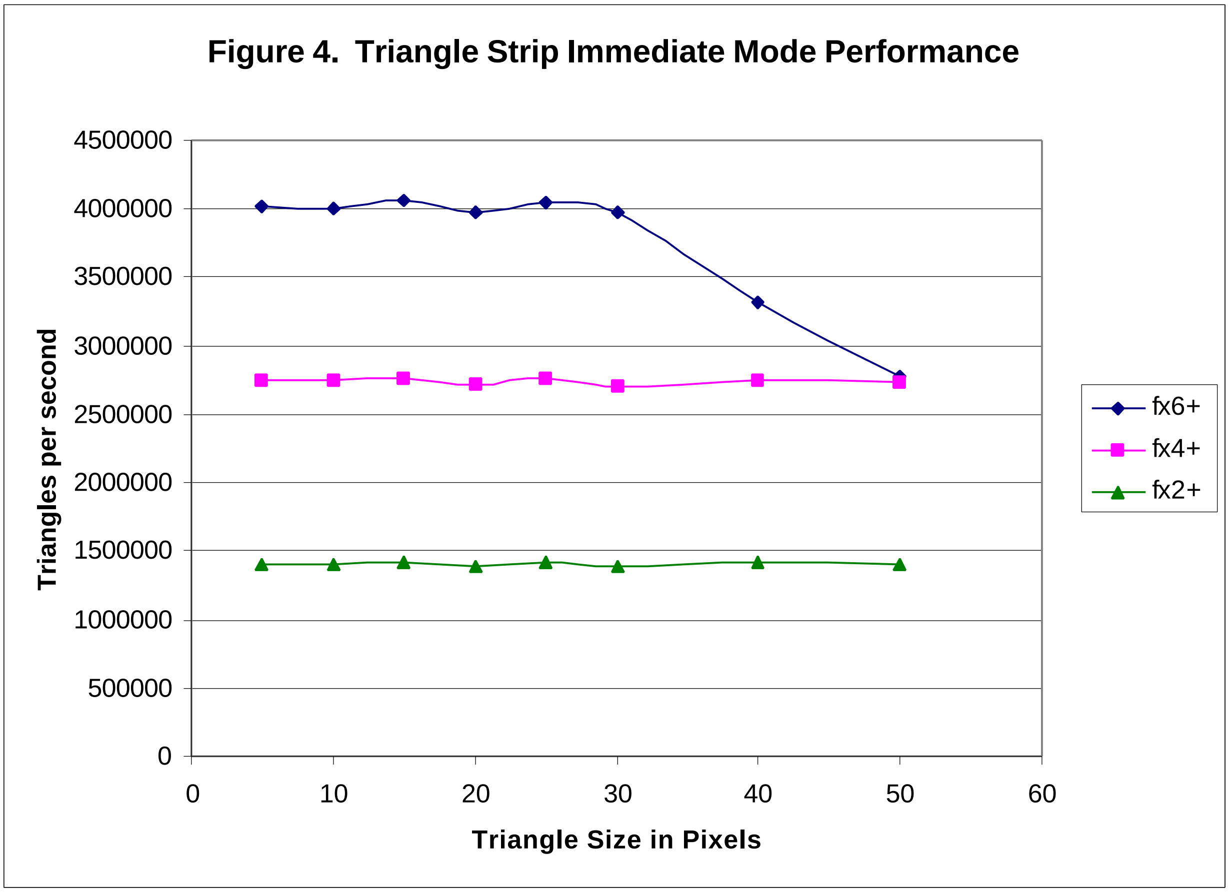


Figure 5 illustrates that the peak fx6+ texture performance is double the performance of the fx4+ and fx2+. Texturing can be done in 3 modes: Point-sampling (the least computationally demanding), bilinear sampling, or trilinear sampling (the most computationally demanding). With point-sampling, the fx6+ is fast enough in performing set-up computations that with triangles of 100 pixels or more, the texture engine achieves the peak near 140 Million Texture elements per second (MTex/s). With half the peak capability of slightly over 70 Mtex/s, the fx4+ has enough geometry power to keep the texture engine busy even with triangles as small as 50 pixels.

With trilinear sampling, the texture engines are capable of performing almost identical to point sampled, as shown in Figure 6. The main difference is that the additional computation for setting up trilinear sampling does not feed the texture engine fast enough for triangles as small as 50 pixels.

With some knowledge of how applications use OpenGL, this data is enough to begin understanding which member of the VISUALIZE-fx+ family would give the best price/performance solution for a given application. If you are working with large, complex models with either a Mechanical Design Automation (MDA), Digital Content Creation (DCC), or Scientific Visualization (SciVis) application, then the fx6+ is the best answer. If you are working with smaller assemblies or parts and much of the work doesn't involve viewing the part, then the fx4+ might be a better compromise for the budget. If you are working with a visual simulation application, which in the past have tended to use lots of texture but relatively few, large polygons, the fx6+ is the best answer. But if you can't afford it, you may want to skip the fx4+ and go all the way down to an fx2+, since the texture fill rate is likely the dominant characteristic.

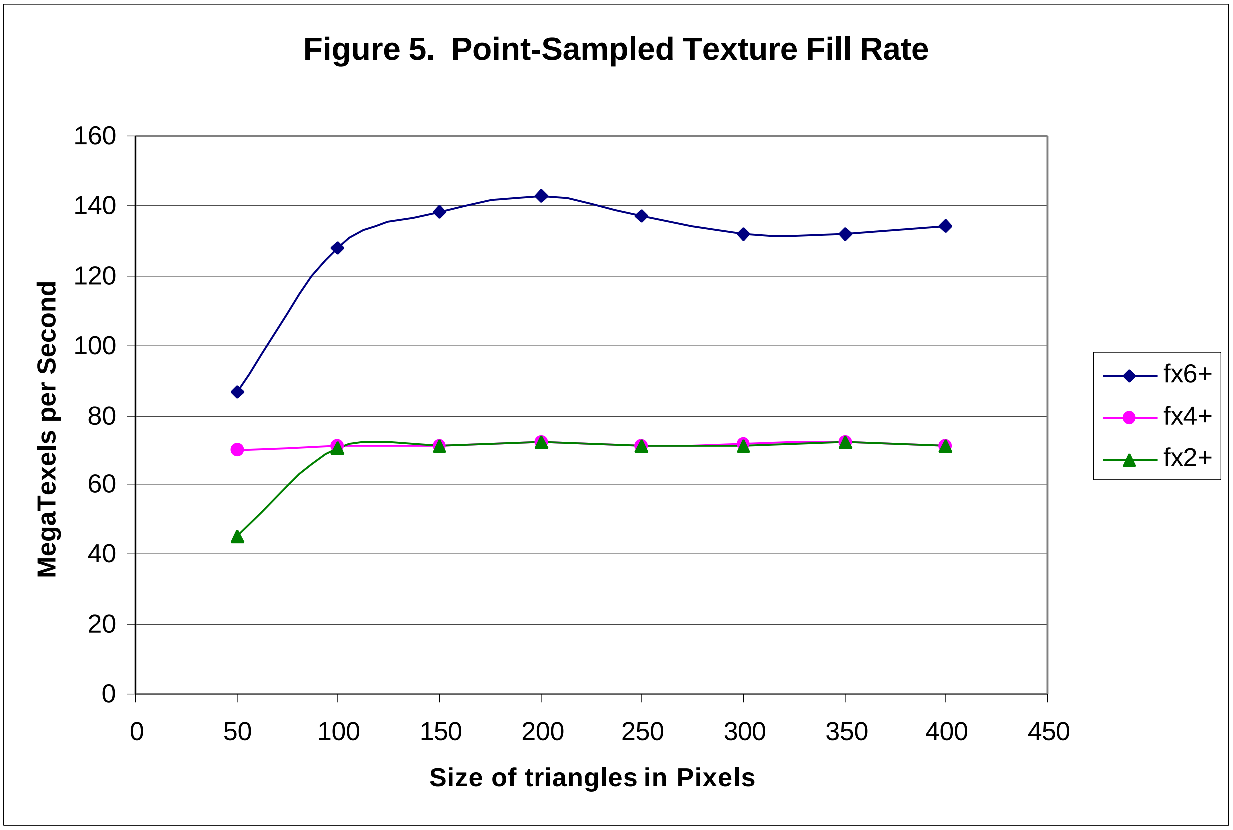


Figure 5. Point-Sampled Texture Fill Rate
MegaTexels per Second
Size of triangles in Pixels
fx6+
fx4+
fx2+
0
20
40
60
80
100
120
140
160
0
50
100
150
200
250
300
350
400
450

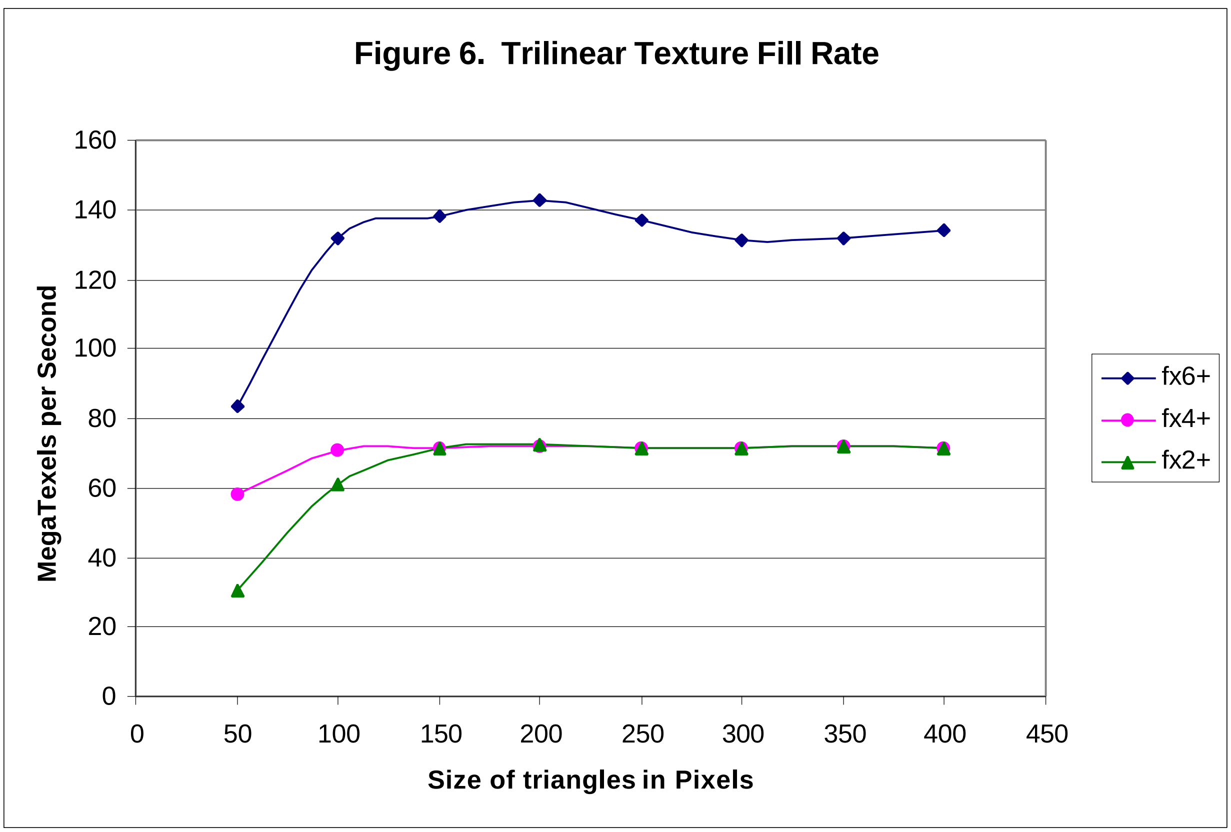


Figure 6. Trilinear Texture Fill Rate
MegaTexels per Second
Size of triangles in Pixels
fx6+
fx4+
fx2+
0
20
40
60
80
100
120
140
160
0
50
100
150
200
250
300
350
400
450

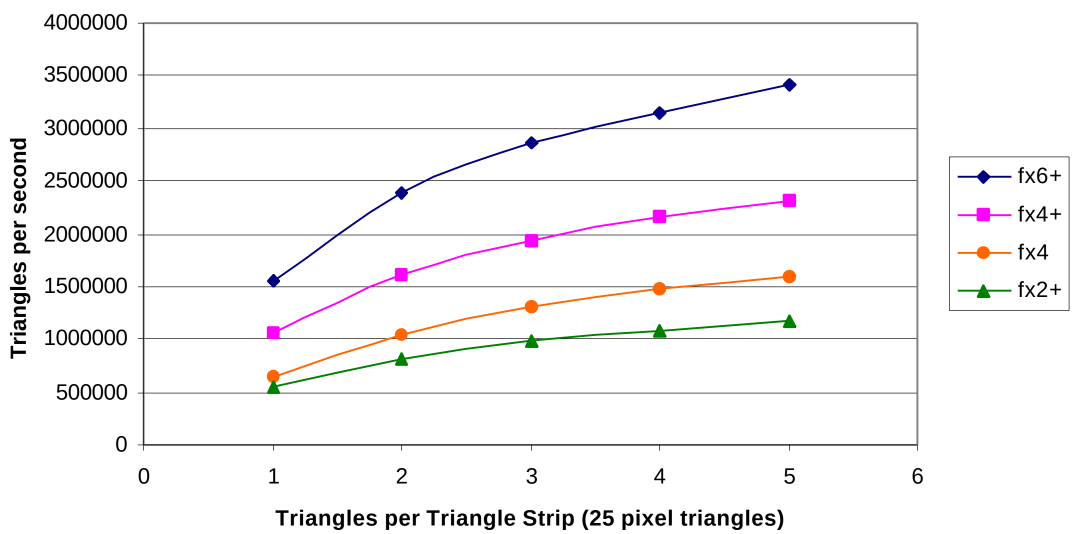
Comparing the VISUALIZE fx4+ to the VISUALIZE fx4

The following features or improvements have been added to the Visualize FX+ products to significantly enhance performance, and quality, as compared with the Visualize FX4.

- AGP DMA Engine. This increases a driver's ability to drive the rest of the hardware to its full capacity.
  - Immediate mode graphics (higher performance and better parallelism with applications)
  - Display list performance for primitives with less than 6 elements (e.g., less than 6 triangles per triangle strip)
  - Faster pixel operations (e.g., glDrawPixels)
  - Faster texture download to hardware (for the initial download, or applications that need more textures than fit in on-board texture memory)
- Support for DirectDraw hardware acceleration
- Asynchronous cursor for better interactivity (e.g., with Unigraphics)
- Faster 2d operations through driver improvements.

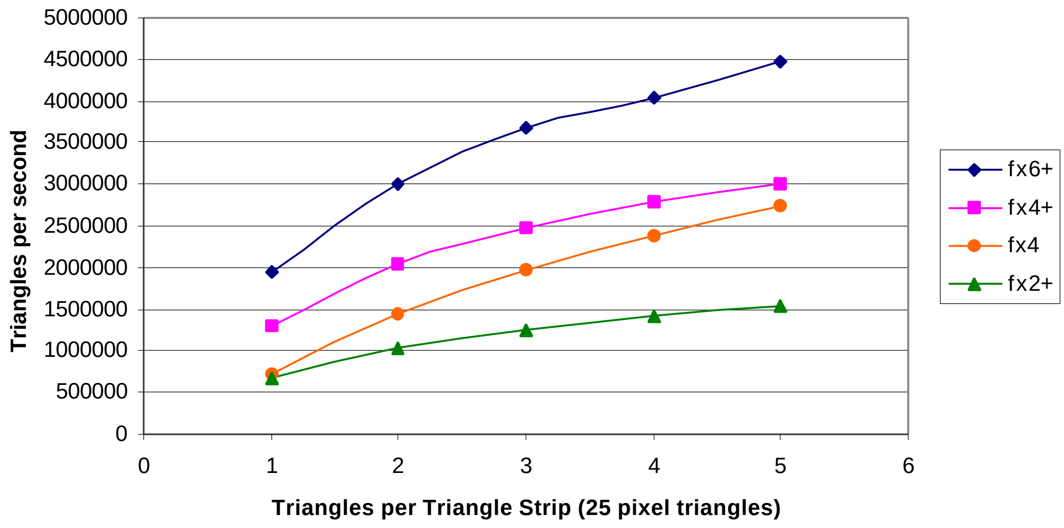
To see the improvement in immediate mode performance, Figure 7 compares the older fx4 technology to the fx+ series. The performance of the fx4 is always bracketed between fx4+ and the fx2+, but generally closer to the fx2+ for single processor systems.

Figure 7. Immediate Mode Performance



The display list performance difference is not as dramatic. The fx4 is comparable to the fx4+ for triangle strips with 7 or more triangles per triangles strip, but is significantly less than the fx4+ for 6 or less triangles in a triangle strip (see Figure 8). This is significant because many applications generate triangle strips with 6 or less triangles per strip.

Figure 8. Display List Performance



.

Occlusion Culling - Why draw what you can't see?

HP is the first to introduce hardware occlusion testing in a personal workstation, and of all the unique features in VISUALIZE graphics, this is a feature worthy of a close look. Occlusion culling requires specific application modifications to take advantage of it. The idea is simple: before rendering a portion of a complex image that could contain thousands of polygons, first see if the object is already hidden by what was rendered before. The cost of the test involves checking if a bounding volume (think of it as a cube with 6 faces) is already occluded. Because of the round trip necessary to get the answer back from the hardware, it is more expensive than just rendering the 6 faces. The actual break-even point for doing the test will vary with the type of data and type of rendering being performed, but is roughly equivalent to drawing 190 25-pixel triangles on an fx6+, and only 65 25-pixel triangles on an fx2+. As the processor performance improves, the cost of the test will continue to go down.

Applications written by EAI and Division are already taking advantage of this feature, with dramatic results. More applications will be released from other companies in the near future. This type of feature will be folded into a standard interface through the Fahrenheit large model efforts that involve cooperation between HP, Microsoft, and SGI.

Graphics Benchmarks and Application Performance

Especially with Windows NT, vendors are susceptible to quoting a single benchmark number. When comparing graphics systems, there is no substitute for benchmarking the actual application you plan to use.

The most popular benchmark used in the last couple of years is the now-obsolete CDRS-03. CDRS is part of the Viewperf suite of benchmarks sponsored by OPC. The CDRS-03 benchmark had two distinguishing characteristics: 1) it is based on a Display-List application (the application has evolved into PTC Pro/Designer), and 2) it heavily weights anti-aliased lines. If the applications you plan to use are not display-list based, and you do not frequently view models using anti-aliased lines, then the CDRS-03 score is irrelevant. All the Viewperf tests (including CDRS) have a description of the model and the test, and how the test weights various graphics features. To use the Viewperf data, it is important to know which graphics features are used by your primary OpenGL applications, and then see which Viewperf test is closest to your application and workloads.

Graphics benchmarks, even those based on the graphics output of real applications like Viewperf, don't simulate the actual characteristics of running the application on the system. For example, geometry accelerated devices (such as HP's fx+ series graphics) outperform non-geometry accelerated devices by more than Viewperf results would imply. The improvement is due to the combination of geometry acceleration and use of DMA to off-load the host CPU, allowing the application to process data in parallel with the graphics rendering.

When benchmarking graphics applications to evaluate systems, the following are recommended:

1) Use workloads that are representative of the actual data used.
2) Segment the benchmark to separate non-interactive operations (such as loading a model) from interactive operations (editing or zooming in on the model.
3) Weight the interactive portions of the benchmark the way an end-user of the application would.
4) Don't bias the evaluation by previous work patterns - some ways of interacting with the model might have been avoided in the past because the graphics subsystem wasn't fast enough, but might mean substantial productivity benefits with VISUALIZE-class graphics.

Conclusion

HP's VISUALIZE family of graphics accelerators for Windows NT deliver state-of-the-art features at a range of performance points. Graphics performance is affected by many variables, which makes predicting the affect on application performance inherently difficult. The actual performance will not just vary from application to application, but it can also be highly dependent on the workload used. By looking at a few salient performance characteristics, one can begin to predict the affect of the graphics performance aspects to the type of application.