Data Complexity for Virtual Reality: Where do all the Triangles Go?

Michael F. Deering
Sun Microsystems Computer Corporation
2550 Garcia Avenue, Mountain View, CA 94043
e-mail address: michael.deering@Eng.Sun.COM

ABSTRACT

Screen space rendering statistics were gathered from 150 3D objects, each modeled by between 2K and 40K triangles. While there is wide variance by individual object, the overall trend is that the distribution of triangles by screen size is roughly exponential in the direction of small triangles. From a subjective esthetics point of view, tessellations required 10K visible triangles per quarter million pixels covered for acceptable results.

1.0 INTRODUCTION

The ability of modern computers to simulate the interaction of human senses with the physical world is the technology that makes Virtual Reality possible. It falls to the technology of 3D computer graphics to attend to the needs of the most important of these senses: vision. Everything we see in virtual space must be created by computer graphics rendering techniques. Limitations of computer graphics technology thus limit the visuals that can be used in Virtual Reality applications.

Most current 3D graphics rendering hardware requires the virtual world to be represented at some level as collections of 3D triangles. Both the basic shape and fine detail of virtual objects are captured by these triangles. Beyond mere esthetics, some minimum quantity of triangles must be used to make objects both recognizable and functional. The brutally minimalistic triangulations of most present Virtual Reality systems are at best “3D icons”; industrial Virtual Reality applications will require more detail. To gain a better understanding of this representation complexity, a study was made of the rendering characteristics of 150 objects, ranging from 2K through 40K triangles in size. Combined with theoretical analysis, estimates of realistic display requirements for Virtual Reality applications can be made. Models of current and future hardware triangle rendering throughput, along with estimates of useful rendering frame rates, yields an envelope of feasible scene triangle budgets, which can be contrasted with the application needs.

2.0 WHY TRIANGLES?

It is beyond the scope of this paper to exhaustively document every alternative rendering primitive to the triangle, and describe why in many cases triangles require less computation. The point is *not* that triangles are the best rendering primitives; for example both NURBS and texture mapping are qualitatively superior; but that most such techniques achieve higher quality at very high computational costs. This section will briefly discuss some of the alternatives to triangles, and the trade-offs involved.

What about the lowly line? Most industrial 3D graphics used in the world today is still done with 3D lines, and modern antialiasing line rendering techniques can achieve very high quality results. The problem with lines is that for Virtual Reality they don’t do a very good job of mimicking real world physical objects. Our own experience is that when used with headtracked stereo displays, lines leave a lesser sense of "presence" than do shaded primitives. Still, they will continue to be a viable alternative for some applications.

Most of today's high quality MCAD editing systems employ some equivalent to trimmed NURBS (Non-Uniform Rational B-Spline) patches to represent very accurately the shape of industrial parts to be constructed. These NURBS models are accurate to within thousandths of an inch, and eventually serve as the master model from which the real parts are physically manufactured. For rendering, NURBS allow a compact representation that can be diced up to different degrees of fineness depending on the needs of a particular frame of rendering [Abi-Ezzi91]. Most NURBS implementations at some level eventually decompose the NURBS into triangles. However, the potential savings in the number of triangles dynamically generated historically have been eaten up by the overhead of evaluating the NURBS and their trimming curves. Also NURBS are primarily useful for representing very fine tessellations of doubly curved surfaces. For simple representations of objects for Virtual Reality, many times a useful minimum triangulation would never use more than two triangles per NURBS surface patch, eliminating any advantage of employing NURBS.

The human visual system is very sensitive to edges, which texture maps supply in abundance. The problem with texture maps for headtracked stereo display is that they are flat. Texture maps have done a good job in flight simulation, where most of the world is beyond the operative distance of human stereopsis. But at close distances, texture maps work best on flat or only slightly curving surfaces (such as wood or stone tables or floors). Bump mapping and displacement mapping are extensions that can improve things slightly, but the uses of texture mapping in representing complex three dimensional machinery are very limited.

3.0 MODEL OF 3D RENDERING HARDWARE

Triangle rendering performance for computer graphics hardware is typically expressed in terms of "n m-pixel triangles per second", where m lately has been in the range of 50 to 100 pixels. (Other variables include the degree of chaining, number and type of light sources, etc.) What must be understood is that for many machines, this performance represents the balance point between the floating point computation needed to process a triangle independent of pixel area, and the frame buffer fill capacity. Thus triangles larger than m pixels typically will render at a rate less than n, because the triangles are fill dominated. Similarly, triangles smaller than m pixels will render at a rate no faster than n, as such triangles are floating point dominated.

Triangulations of "solid" objects have the property that only front facing triangles need to be rendered for proper display. While some objects are more detailed on one side than the other, in general, for most objects, close to 50% of the triangles will turn out to be backfacing for any given view (as will be born out in the statistics below). These 50% of the triangles will "render" at a rate somewhat faster than visible triangles, unless data transmission is also saturated.

But real triangles are rarely all of one uniform size; tessellated objects give rise to a mix of different triangle sizes in model space. The projection to screen coordinates further smears this mix. Statistically each size of triangle in model space results in an even distribution of sizes of screen space triangles, from 0 pixels to some maximum, weighted by the relative population of that triangle size in model space. To accurately judge application performance, one needs to know the overall distribution of triangle areas. The main empirical result of the next section gives us this information.

4.0 EMPERICAL TRIANGLES

Models of 3D objects in computer readable form have historically been hard to obtain. Liability concerns have deterred the manufacturers of the real objects from letting out their design databases. Point by point digitizing of physical objects has been expensive, and techniques using video cameras and/or range sensors are still mostly experimental. Objects can be constructed from scratch using 3D CAD systems, but the resulting triangulations are usually un-optimal without additional post processing. One result of this state of affairs is that most 3D graphics practitioners and users do not have very good intuitive understanding of the forces driving the number and range of sizes for various qualities of triangulations.

The author was fortunate enough to have access to more than 150 optimized triangulations of 3D objects from Viewpoint Animation Engineering for statistical study. This section presents the main results of this study, and a sampling of individual statistics. All of the objects represent hand digitized solid shells of real objects. The hand sampling was biased to result in a minimum number of triangles for a given level of detail. Human judgment decided what details could be eliminated at low resolutions. For example: automobiles lost their door handles and outside mirrors; tigers lost their teeth; warships lost their guns.

All objects were scaled to the same size in model space in their maximum dimension, and projected into a 700×700 region of screen space (most filled about 250,000 pixels). The area of each triangle was recorded, both in model and screen space. Information was also gathered about pixel depth complexity and the percentage of backfacing triangles. Some representative individual object statistics are presented in Table 1. (Wire frame hidden line removed renderings of several of these objects are presented in figure 2.) The first column gives the name of the object. Most of the names are self descriptive, the first several are automobiles. The second column gives the total number of triangles in the object, the third gives the percentage of these that were eliminated by backface culling. The *Kpixel rendered* column gives the total number of pixels that were scan converted. *Avg sz* is the average screen space size of the rendered triangles, *Med sz* is the screen space pixel size of the median triangle. The *0 nul* column indicates the percentage of visible triangles that contained no valid pixel samples. The next nine columns shows what percentage of the visible triangles had screen space pixel area in the columns' indicated range. Note that each range is (mostly) twice as large as the last. The last six columns are similar bins, but for model space triangle area (and include all triangles in the model, not just the visible ones).

The 50% backfacing assumption held to within a few percent for most of the objects examined. There was some angle of view dependence: when cars were viewed from below, the backface rejection went up, as most have lightly detailed undersides. (The rejectance remained steady for one car with a very detailed underside.)

The depth complexity (ratio of total rendered pixels to final visible pixels) stayed generally in the range of 1.25 to 1.5 for most objects that had little or no internal detail, but increased considerably as such detail was added (well above 2).

The most interesting result of the study was that the distribution of triangle area in both model and screen space is roughly exponential in the direction of small triangles. This can be seen in the table data in how much smaller the median triangle area was from the average triangle area. About the only difference between objects of similar area was in the overall breadth of the distribution; at the narrow end 80% of the triangles varied in area by less than a factor of 32, at the wider end by a factor of 128. As object complexity (number of triangles) rises, the distribution naturally shifts down toward smaller triangles, and generally the shift is linear with the complexity increase. While the individual data presented here was computed for a particular screen image size, the results should all be linear in any scale change.

The data does show variance with individual object nature. The object in the table labeled "duneM" is a dunebuggy, with no body panels, just lots of narrow curved tubes, and as a result has a larger depth complexity. The "joshtreH" was the only artificial object, and lots of small spines are occluding others, resulting in a depth complexity over 5. "Mortimer" (a mouse) and "low-top" (a shoe) are made up of much more uniform triangles than the other objects, as evidenced by the smaller ratio in average to median area size compared to the other objects. There also is some variance by view angle. "R85skylark" is the same "85skylark" car object, but at a direct side view, rather than the angled view used for all the other objects.

While much more subjective, the analysis also gave intuitions into trade-offs between realism and object complexity. The simplest cars were about 2K triangles, and looked very cartoon-ish. Even the 5K triangle "78malibu" still felt overly simplified. Most objects started looking real good when the triangle count got into the 10K to 20K range. It must be remembered that the statistics presented here are for *one* object; many Virtual Reality applications will want to fill the entire field of view with the same complexity that these statistics fill only a quarter million pixels with.

Despite the number and range of types of objects tested, the results here must be treated with a grain of salt; additional types of objects may have quite different statistics. The main trend likely to hold up is the heavy skew of triangle population toward those of smaller area, especially as more detail is added. What this means for application rendering performance is that it will typically be dominated by the floating point stage. The good news is that rendering performance in triangles per second will approach the vendor quoted performance; the bad news is that the effective pixel area fill rate will fall well below the benchmark levels. When backfacing triangles and depth complexity are factored in, final filled pixels in objects similar to those in our sample database will fill at least 5 to 10 time slower than non-overlapping larger benchmark triangles.

5.0 INTERESTING FRAME RATES

It has been known since the early nineteenth century that the human visual system requires frame updates rates of a minimum of 14 to 18 Hz for "motion fusion" to occur, e.g. for a succession of still frames to merge together into the illusion of continuous motion. Most early Virtual Reality systems were hampered both by long latency 3D tracking devices and limited hardware rendering rates, and very rarely achieved anything like true 14 Hz updates. But for Virtual Reality to be used for commercial applications, reasonable frame rates must be achieved. It is not a question of "the user getting used to" slower rates. Frame rates in the range of 2 to 10 Hz drive the human motor system into oscillation, e.g. low level motor control of the head will keep overcorrecting for perceived visual input changes. (Studies are currently underway to quantify user application task performance drop-offs at sub-critical frame rates.)

Thus interesting frame rates for head tracked stereo Virtual Reality systems are two frames (left/right) times 18 to 60 Hz.

6.0 COMBINING COMPLEXITY AND FRAME RATES

Historically the "scenes" of MCAD application screens have been idealized (at least in demos and benchmarks) as an object (like a gear or auto body) tumbling on the screen. There is no environmental background (as there is in flight simulation and many VR applications), the majority of the pixels on the screen are cleared to a constant background color. Typically hardware can perform this clear function even faster than the peak triangle fill rate. Thus only a fraction of the pixels on the screen must be filled by rendered triangles. But the scenes of Virtual Reality applications tend to be more complete. Even with excellent bounding box culling, an inclusive environment filled with (virtual) physical artifacts and background at the same triangle complexity as the acceptable looking models of our statistics would require a minimum scene complexity of 30K triangles; with complexities of 300K easily imaginable.

Figure1 plots this scene complexity on the horizontal (logarithmic) axis with the frame rate requirements of the last section on the vertical (logarithmic) axis. The shaded box represents the intersection of these two desires; the diagonal lines are the implied triangle per second rendering rate. The result is a graphics rendering rate beyond what is supported in today's desktop computers, but well within the range of where the technology should be headed within the latter half of this decade.

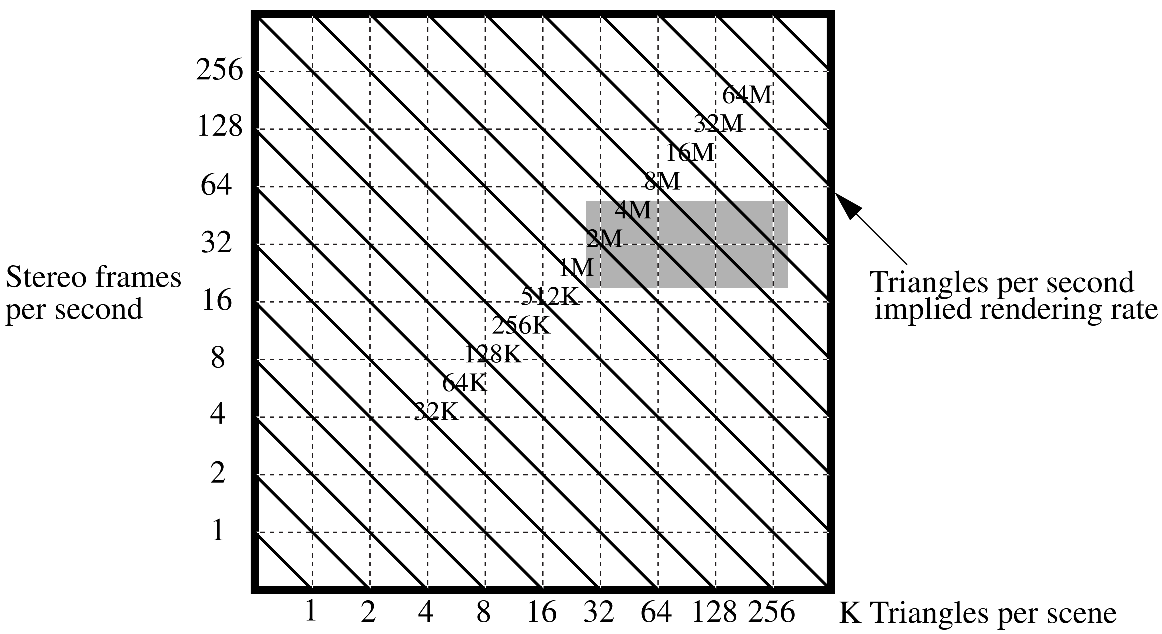

Figure 1. Intersection of desired scene complexities and frame rates.

7.0 CONCLUSIONS

For Virtual Reality applications to be successful we must be realistic about the graphics performance needed to support these applications. For applications employing triangulations of objects for rendering purposes, rendering statistics were gathered from a large number of nearly optimally tessellated 3D objects. The result bears out many people's intuition: even moderate amounts of realism requires lots and lots of very small triangles. Combined with physiologically acceptable stereo frame rates, it can be seen that further increases in hardware rendering rates will be needed to achieve the minimal requirements of many industrial Virtual Reality applications.

8.0 ACKNOWLEDGMENTS

The author would like to thank Scott Nelson for his help with setting up the database, and Viewpoint Animation Engineering for support of their database.

9.0 REFERENCES

[Abi-Ezzi91] Abi-Ezzi, Salim and L. Shirman. Tessellation of Curved Surfaces under Highly Varying Transformations. Proc. Eurographics '91 (Vienna, Austria, September 1991), 385-397.

duneM

tigerH

85skylark

tigerM

86taurus

tigerL

80deloreanH

Figure 2. Wire frame images of sample objects.

This

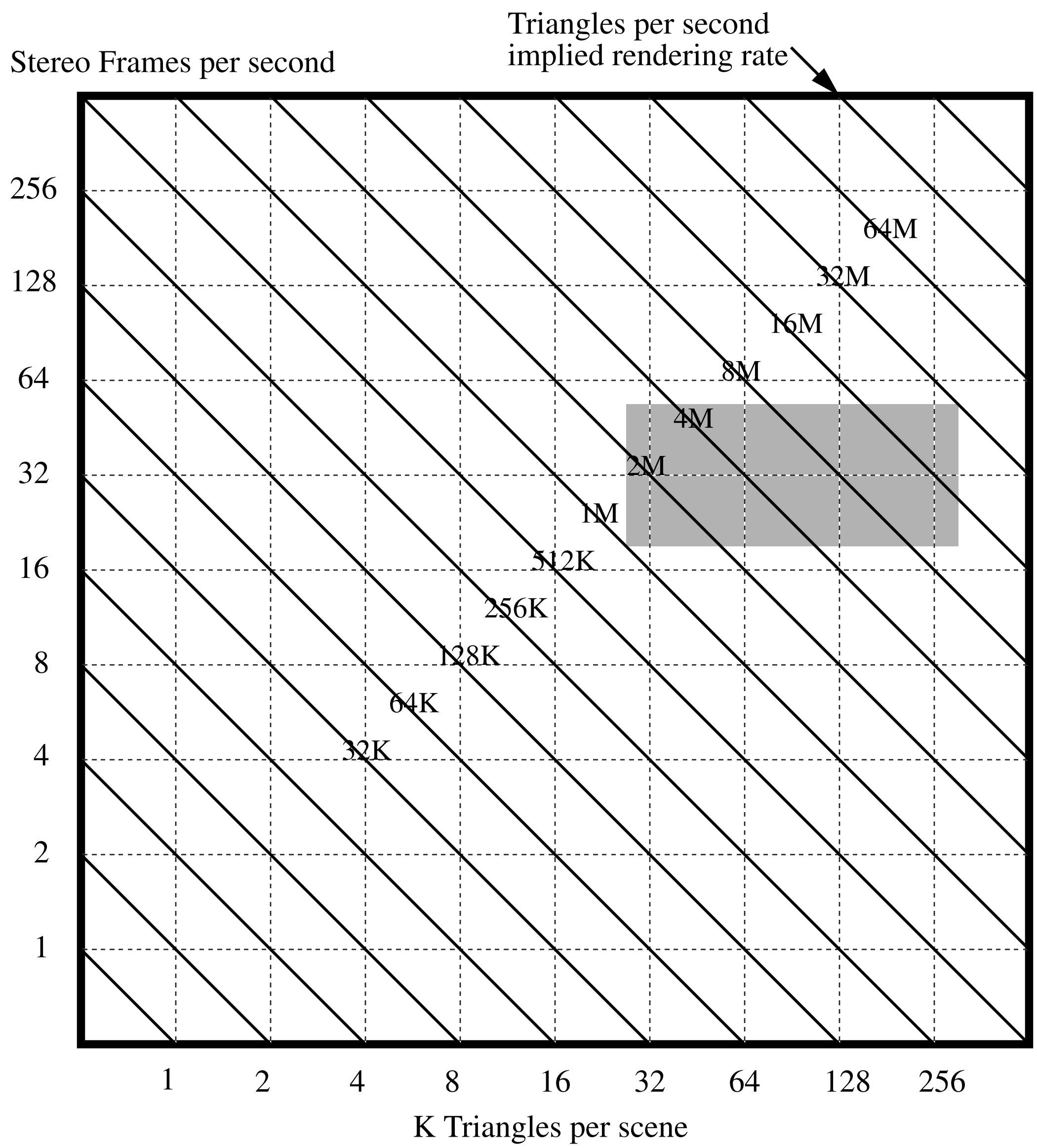

Figure 2.Intersection of desired scene complexities and frame rates.

That

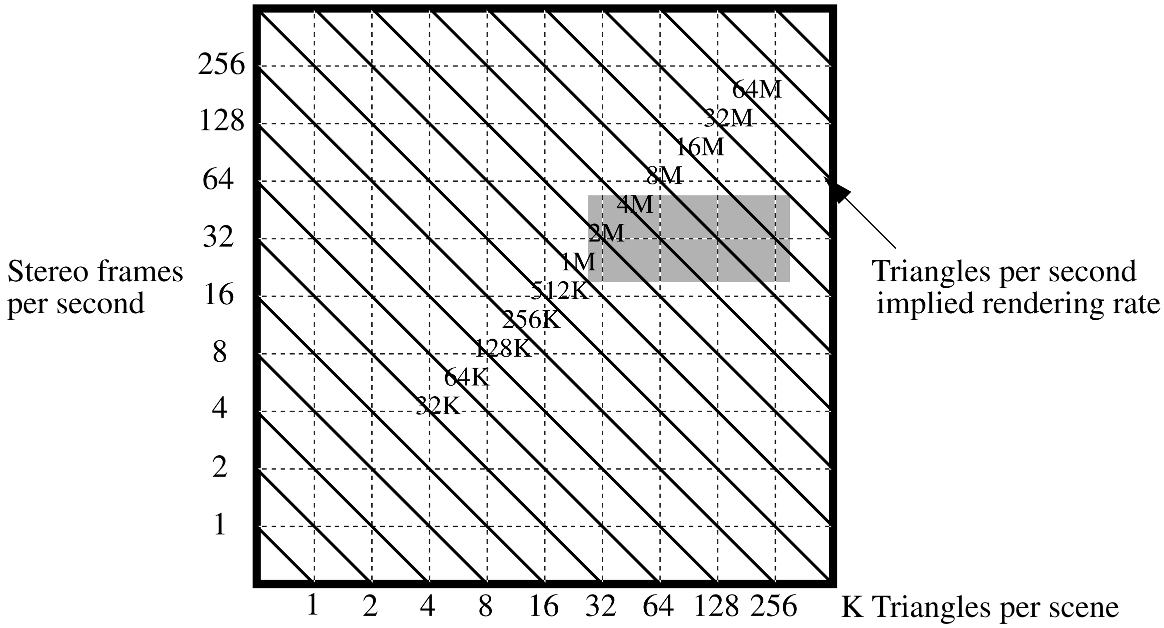

Figure 2. Intersection of desired scene complexities and frame rates.