Polygon Rendering with Adaptive Bit Resolution

5H−3

Jacob LEVISON, Ichiro KURODA
Information Technology Research Laboratories, NEC Corporation
e-mail: jacob, kuroda@dsp.cl.nec.co.jp

1 Introduction

Polygon rendering is among the most computational and memory intensive part of computer graphics, and dedicated graphics accelerators are often applied to obtain the required performance.

As programmable processors such as DSPs are getting more powerful, including high memory bandwidth, new potential application areas such as memory intensive polygon rendering are emerging. Obviously programmable architectures offer more flexibility than dedicated hardware, thus the rendering algorithms should be enhanced to exploit these capabilities.

An adaptive rendering algorithm, presented in this paper applies the fact that a large percentage of triangles in computer graphics scenes are small. For small coordinate and color spans, low bit resolution provides sufficient precision, allowing implementation on fixed-point architectures with small word size.

2 Triangle Rendering

In shaded rendering, pixel coordinates and colors are interpolated across the surface of each triangle. This linear color interpolation is also named Gouraud shading.

2.1 Line Rasterization

A line interpolation algorithm, Digital Differential Analyzer (DDA), offers an incremental evaluation of the equation of the line $x = my + b$. Each time y is incremented by 1, x is incremented by m. If the coordinate values are unsigned 7 bit numbers, the maximum and the minimum slope m is 127 and 1/127 respectively. Thus, 7 bit is required for the integer part as well as the fractional part, resulting in 14 + 1 sign bit. On the other hand, large triangles need 15 bit unsigned numbers, thus, 30 + 1 sign bit accuracy is required. Handling 32 bit data types in a 16 bit architecture such as a fixed-point 16 bit DSP [1] results in an increase of instruction cycles.

2.2 Interpolation of Color and Z

A triangle is defined by the implicit equation

$$F(z, y, x) = ax + by + cz + d = 0, \qquad (1)$$

where (a, b, c) is the normal vector to the plane. The z-value can be expressed by the equation

$$z(x, y) = -(ax + by + d)/c. \qquad (2)$$

d is found from Equation 2 by inserting a vertex coordinate. By replacing z, corresponding equations for R, G, and B are found. Equation 2 can be regarded as the scan function of the triangle plane and a corresponding setup function is required to calculate the constants a/c, b/c, and d/c.

Equation 2 can be calculated by an incremental algorithm in a similar manner as the DDA. In this case, two scan functions are needed, one when incrementing x (scan-x) and another when incrementing y (scan-y).

Table 1 shows the number of cycles required for the setup, scan-y, and scan-x functions, calculating the z and RGB values on a fixed-point 16 bit DSP. Changing the accuracy from 16 to 32 bit increases the cycles for the setup, scan-y, and scan-x functions by 133 %, 75 %, and 60 % respectively.

ヤコブ レビソン、黒田 一朗
NEC 情報メディア研究所

表 1: Plane DDA instruction count

Instruction	Setup		Scan-y		Scan-x	
	16	32	16	32	16	32
Load/Store	153	306	12	24	12	24
Add/Sub	31	62	4	4	4	4
Div	60	180	0	0	0	0
Shift	7	21	0	0	4	4
Mpy	24	72	0	0	0	0
Total	275	670	16	28	20	32

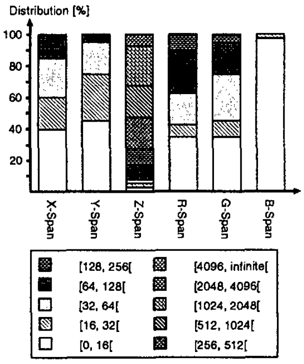

図 1: Span distribution for *Teapot*, 240 polygons.

3 Adaptive Rendering

By analyzing a number of computer graphics scenes it is found that most of the spans between the vertex points are relatively small. Figure 1 shows the distribution of the maximum span size for each triangle in the test scene *Teapot*, which consist of 240 polygons. The image resolution is 515x512. Excluding the z-span, 90% of the triangles' spans can be represented by less than 8 bit.

It is desirable to use only the necessary bit width. As 75 % of the z-spans require more than 7 bit representation, the focus is directed towards the x-, y-, and RGB-span. As the constants a/c, b/c, and d/c depend on the x- and y-span as well as the RGB-span. Thus, by evaluating the x-, y-, and RGB-span for each of the three lines, it can be decided whether 16 bit accuracy is sufficient. The additional decision code can be implemented in less than 75 cycles, increasing the setup cycle count by 21 % for 16 bit, which is negligible.

4 Performance Improvement

The adaptive algorithm applies 16 bit width for the x-, y-, and RGB-span whenever the spans are less than 128, otherwise 32 bit. Figure 2 compares 32 bit rendering to the new adaptive approach, assuming that the x-, y-, and RGB-span are less than 128. If rendering the scene with the span distribution shown in Figure 1, 90 % of the triangles will be rendered with only 16 bit precision for the x-, y-, and RGB-span, resulting in 30 % reduction of the processing cycles for those triangles.

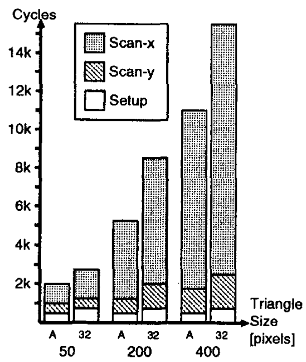

図 2: Reduction of processing cycles.

The rendering speed, in terms of triangles per second, can be found by including the cycles required for z-buffer operaton. Assuming the DSP has access to a fast frame buffer, the operation can be completed in 5 cycles per pixel. With a 33 MHz clock rate, this gives a performance of 14.000 triangles per second for 50 pixel triangles.

5 Conclusion

An adaptive polygon rendering algorithm has been proposed. For a 16 bit DSP architecture, the total number of processing cycles are reduced by 30 %.

参考文献

[1] NEC Corporation, "μPD77016 Digital Signal Processor", *User's Manual*, August 1993