4F－5

FlashPix based Texture Mapping*

Kei Yuasa

email:ky@jp.hpl.hp.com

Hewlett-Packard Laboratories Japan

1. Introduction

Due to advancements of PC and networks, 3 dimensional computer graphics (3DCG) have become popular throuout the Internet world. 3DCG basically consists of polygons and textures which have contrary features. The number of polygons has direct effects to a calculation time while the texture size does not affect it directly. So, a few polygons with rich textures can provide realistic and natural images at low computation power. However, an abuse of texture data makes transmission time so long particularly in a network environment.

This problem originates in the nature of conventional rendering algorithm. I changed the algorithm to minimize the loading time. A new buffer is introduced for deferring the texture loading and FlashPix was utilized for texture transfer.

This report describes the new method. Section 2 review problems of the conventional algorithm. Section 3 explains my new algorithm and data structure. Section 4 compares the new method with the conventional one, and Section 5 is for conclusion.

2. Conventional Algorithm

Among many 3DCG techniques, Z-buffering is the most popular due to its simple mechanism[2,3]. The outline of the algorithm is as follows:

1. *Load Phase* loads all polygons and textures
2. *Geometry Phase* calculates perspectives
3. *Register Phase* registers the polygon to a depth and pixel buffers to remove hidden surfaces.
4. *Display Phase* outputs the result

Texture data are accessed in the Register phase for pixel values. Actually, not all textures appear in the final image because some polygons are occluded, back-faced, or clipped. Necessary resolution of texture image depends on the result of rendering. However, it is impossible to know which part of textures are necessary before the calculation, so that in the first phase, it is necessary to load all texture images in highest resolution.

3. New Method

3.1 Algorithm

The new algorithm is as follows:

1. *Polygon Load* phase loads only polygons
2. *Geometry phase* calculates perspectives
3. *Register phase* rasterizes the polygons to eliminate hidden surfaces
4. Finally, *Display and Texture Load phase* loads necessary parts of textures and outputs the result

The difference from the conventional algorithm is the timing of texture loading. It is deferred after the Register phase so that only necessary parts for the final image are loaded.

3.2 Reference buffer

A reference buffer was introduced between the depth and pixel buffers in order to defer the texture loading. Its each element holds the intermediate status of texture mapping calculations. It is a union of a direct color (for polygons without textures) and a texture reference whose structures are as follows:

- **<#D, pixel_value>**
- **<#T, pointer_to_texture, (s, t), intensity>**

For a direct color, RGB triple is stored intact. For a texture mapped pixel, the pointer to the texture, texture coordinates (s, t), and an intensity coefficient are stored. They are used for calculating pixel

---

* FlashPix によるテクスチャマッピング，湯浅 敬，ヒューレット・パッカード日本研究所

values. In the new algorithm, they are stored in the reference buffer for later calculation. When the reference buffer is copied into the pixel buffer, necessary parts of the texture are loaded from files or servers.

3.3 FlashPix Texture

FlashPix[1] is data format for Internet Imaging which was originally made for 2D imaging. It consists of a resolution pyramid and tile. The original image is converted into a resolution pyramid and each layer is divided into 64 by 64 pixel tiles. IIP (Internet Imaging Protocol) can fetch texture images in tile by tile basis.

In our 3DCG browser, a proxy object is assigned to each texture to manages FlashPix tiles. In the Display and Tile Load phase, the object is requested pixel colors. It returns pixel values from currently loaded tiles and loads more tiles on demand. By iterating the last phase, it is possible to improve the texture resolution progressively. Figure 1 to 4 shows how the globe object improves its texture resolution.

4. Benchmarks

A benchmark was measured for comparison with the conventional algorithms. Both programs are executed on HPUX workstation. All data are stored in its local files and read at 1 Mbps. "The Globe" in Figure 4 has a sphere with 288 polygons and a texture of 1022 by 835 pixels. In FlashPix format, it has 683 tiles in 6 layers of pyramid. Table 1 shows the timings of the two algorithm

	Conventional	New Algorithm
Polygon Load	0.0027	0.0027
Texture Load	22.48	0.0983 / tile
Calculation	2.55	2.33
Display	0.082	0.315 / tile
1^{st} Image	25.147	2.746

Table 1 Benchmark Timings (in seconds)

The conventional algorithm load texture whose size is the same as 224 tiles while the new algorithm load only visible tiles. As the result, the new algorithm can output the first image in 2.75 seconds and the final image appears after 33 seconds. Although the first image has very low resolution, a user can grab the shape of the objects. If the data are taken from network server, this advantage will be much bigger.

5. Conclusion

I designed a new algorithm for texture mapping for a network environment. It introduces FlashPix objects and a reference buffer to defer the texture loading. As the result, the network resources can be used efficiently and the rendered image can be output in shorter time.

References

[1] H. J. Lee, "Imaging for Internet", Web Techniques, Vol. 1, Issue 9, Dec. 1996.

[2] Foley, "Compute Graphics Principles and Practice", Addison-Wesley, 1996

[3] Heckbert, "Survey of Texture Mapping", IEEE CG and A, 6(1986), 56-67

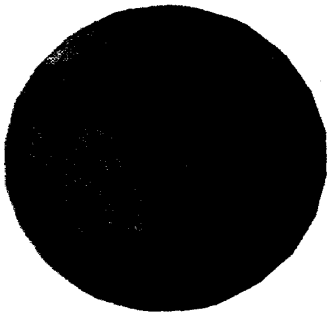

Figure 1. Sample 1

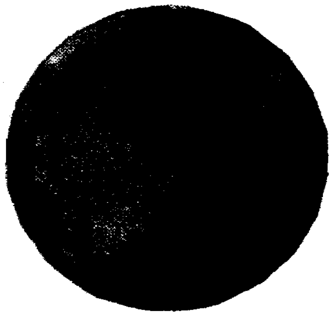

Figure 2. Sample 2

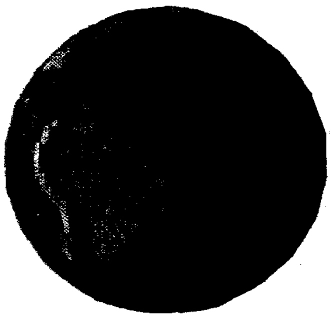

Figure 3. Sample 3

Figure 4. Sample 4