Minimizing Costs In Lighting Hardware

B. Bonello[&], C. Wolters[&], H.. Rüsseler[§], O. Wittig[&], T. Jung[§],

[§]German National Research Center for Computer Science,
Institute for Computer Architecture and Software Technology (GMD FIRST)
[&]NOVEDIA GmbH Technology Partners

Abstract

This paper presents a new adaptive method to support per pixel lighting for a fraction of the hardware costs of standard per pixel shading.
The Adaptive Sample Shading hardware approach simulates true per pixel lighting by analyzing the local lighting situation and evaluating the chosen lighting model for relevant pixels only. The lighting calculation at calculated samples is further optimized by computing only the relevant light components that influence the particular polygon. An efficient sample positioning algorithm employs spatial coherence and calculates the minimal number of samples to achieve better image quality adjusted by a threshold value. This approach allows a minimized lighting unit to cope with complex lighting.
In this paper a Phong hardware pipeline is used as a reference per pixel lighting unit. Advantages and disadvantages are presented based on statistics of sample scenes.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Hardware Architecture - Graphics processors; I.3.3 [Computer Graphics] Picture/Image Generation -Viewing and Display algorithms; I.3.7 [Computer Graphics]]Three-Dimensional Graphics and Realism - Color, shading, shadowing, texture.

Additional Keywords: adaptive shading, adaptive local lighting, scalability, illumination models, normal vector interpolation, per pixel lighting, spot-lights, Phong shading hardware.

1. INTRODUCTION

Today's graphics hardware focuses on high fill-rates, and a high polygon and texturing performance. NVidia for example first introduced the TNT graphics accelerator with a dual rendering pipeline which speeds up fill-rates significantly.
Specifically in VR and games, multiple texture passes are used to add highlight-, bump, reflection- and other special effects. Software optimizations for on the fly texture computation simulate reflected objects in mirrors and rich illumination [Mill]. Apart form the quality limitations and constraints that light maps have with varying material coefficients per polygon, they substantially increase bandwidth requirements.
Managing multiple textures is elegantly addressed e.g. in the virtual texture cache of the 3Dlabs Permedia 3 graphics accelerator. Specialized DRAMs [TexRAM] include logic for filtering, blending and z-compare to compensate texture bandwidth.
A recent development of the G400 graphics accelerator by Matrox supports per pixel environment bump mapping with similar features to those of the research prototype VISA+ [EnvBump].
Per pixel effects free texture bandwidth and gives way to use lesser and bigger triangles which in return relieve the burden on the geometry engine.
A next step for even better pictures as stated in [Kirk] will be the introduction of richer illumination algorithms. A new illumination algorithm has to be evaluated on a per pixel basis and is likely to be computational complex.
Straight forward implementations of per pixel lighting present a heavy performance penalty in the absence of parallel illumination units. The addition of which is costly and runs the risk not being extensively used in real applications.

For non per pixel lighting Hardware Cho, Neumann et al. [Neum] ,[Tek] presented a paper to adaptively re-tessellate polygons hit by highlights to reduce the Gouraud artifacts within the polygon. One limitation of this method is that it does not reduce artifacts for spotlights with a given cutoff-angle.
In this paper - based on per pixel lighting - a different approach is presented introducing light-cones. These light-cones detect relevant light sources illuminating the currently processed polygon.
Adaptive Sample Shading is not limited to Phong Illumination [Phong75] and the analysis of the local lighting situation would help to adaptively reduce complexity of other illumination models.

1.1 Redundancy in per pixel shading

Illumination intensities within a 3D-scene contain substantial redundancy which can be pinpointed using the following observations:

- Shiny surfaces have very small areas where the intensity changes substantially from pixel to pixel (around the highlights), the remaining areas are mainly affected by low frequency diffuse components or the specular component of the light source if it is very close to the polygon.
- For dull surfaces, larger areas are influenced by each light source but their intensity gradients are smaller.
- A particular polygon rarely contains all the highlights of all the light sources.
- Beyond its cut-off angle, a spot light source does not contribute to the intensity of a specific area.

Adaptive Sample Shading makes use of these observations to, thirsty introduce interpolation between computed light intensities

and secondly to control which parts of the lighting equation has to be computed for a given pixel of the scanline.

2. ADAPTIVE SAMPLE SHADING

Adaptive Sample Shading is a method by which the redundancy of the lighting calculation is reduced by dynamically setting samples per light source depending on the local lighting situation and viewing position. The sampling density is adjusted adaptively to achieve high quality with no perceivable difference between per pixel Phong shading and Adaptive Sample Shading.
Adaptive Sample Shading shows no artifacts at the polygon edges, fully supports highlights within polygons and sharp spot-light cutoff-angles.

With Adaptive Sample Shading the calculation expenses can be bounded to maintain a specific frame rate by adjusting error tolerances to enable high frame rates with complex lighting.
Adaptive Sample Shading can be applied either to the final illumination intensities, or alternatively to the separate light sources before their superposition.

At critical positions within the scanline, additional accurate illumination samples are calculated. The remaining pixels are interpolated.

2.1 Sample Positioning

For the positioning of samples it is important to reliably detect regions of large intensity variation (in proximity to highlights and spot cutoff-angle).
Within a polygon we define fixed sample positions with a pixel-distance of e.g. 8 or 16. For these sample positions the lighting equation is calculated. The difference of resulting intensities of two neighboring sample positions is compared against a given threshold.
Whenever this threshold is exceeded a new sample is generated as described later.

2.1.1 Conditions for adaptive samples: Threshold Method

To reduce the probability of unnoticed small highlights, it is advisable to ensure that the samples maintain a maximum pixel distance between each other called maximum sample distance. These samples and the edge samples are called regular samples.
As described later the maximum sample distance can be determined depending on whether a small highlight is assumed to affect a particular curved triangle.

2.1.2 Generating adaptive samples by binary Subdivision

A binary subdivision method can be used to generate adaptive samples. These samples are placed halfway between the existing sample pairs. This can be repeated recursively until the threshold condition is fulfilled. All remaining pixel are interpolated. For sample distances of sixteen pixels, four subdivisions suffice.

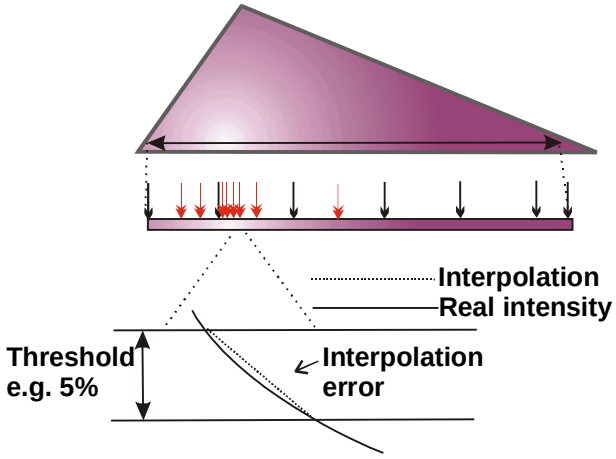

Fig. 1 Scanline with regular (black) and adaptive (red) samples

For a hardware implementation it is important to limit the number of recursions. By limiting the number of recursions it is necessary to either increase the threshold or to limit the distance between two regular samples. Increasing the threshold results in illumination artifacts. For example disappearing small highlights for large distances of regular samples and machbanding between two samples are likely to occur. Decreasing the distance between regular samples in return leads to better image quality with higher computational costs.
In our investigation a distance of 8 to 16 pixels between regular samples leads to a good image quality.

2.2 Adaptive Local Lighting

Adaptive Sample Shading can be further optimized if by analyzing the local lighting situation per polygon:

The following information is used to determine those light sources illuminating the current polygon:

- light source position
- light-direction
- cut-off angle
- material shininess
- attenuation

Fig. 2 illustrates an example illumination situation for a given surface.

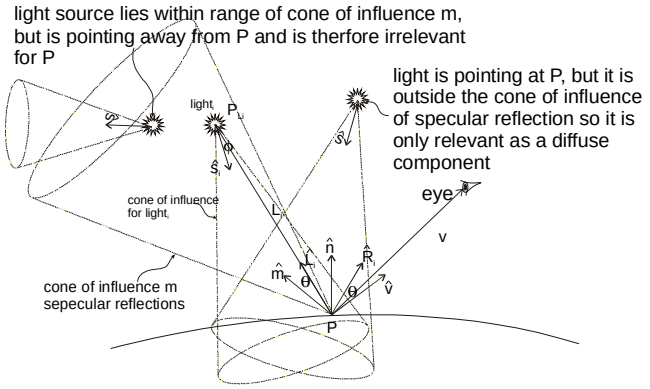

Fig. 2 A typical light situation

2.2.1 Detecting Potential Highlights

The detection of relevant lights for a surface are based on the following observations. Looking from the polygon into its surrounded scene, only those lights which contributes to its illumination will be seen. This principle will be described as the Bounding Cone Method in the next paragraph. Secondly, by looking from the light source onto the scene, one can detect all surfaces illuminated by this light source. This is called Volume of influence of light sources, described in section 2.3.3.

Cho, Neumann et al's [Neum] method to test, whether the center of a highlight potentially lies within a triangle for a particular light source was optimized to a few cross and dot products per vertex, per light source.

In our approach we compute a bounding volume including all reflected vertex eye vectors m_j. We enlarge this bounding volume by an angle to take highlights into account whose centers lie outside of the triangle. The angular enlargement of the bounding volume depends on the material properties. If a light source lies within this volume of influence, it potentially affects the polygon.

In the following we describe bounding cone method in detail.

2.2.2 Bounding Cone Method

In this paragraph we describe the calculation of the bounding cone.

The eye-vectors *V* are reflected by the normals n_j using

$$M_j = 2n_j\left(n_j \cdot V_j\right) - V_j .$$

The average of the normalized reflected eye-vectors m_i are used as the cone axis.

$$m_m = \frac{m_1 + m_2 + m_3}{\left|m_1 + m_2 + m_3\right|}$$

In order to find the opening θ_c angle of the cone which includes all m_j, the maximum angle with respect to m_m is calculated.

$$\cos(\theta_c) = \min\left[m_1 \cdot m_m \quad m_2 \cdot m_m \quad m_3 \cdot m_m\right]$$

The cone has to be opened up by $\Delta\phi$ to include the cases where the center of the highlight is just outside the polygon, but still affects it. This angle depends on the material coefficient S_{rm} and a tolerance threshold.

$$\theta_{cone} = \theta_c + \arccos\left(\sqrt[S_{rm}]{threshold}\right)$$

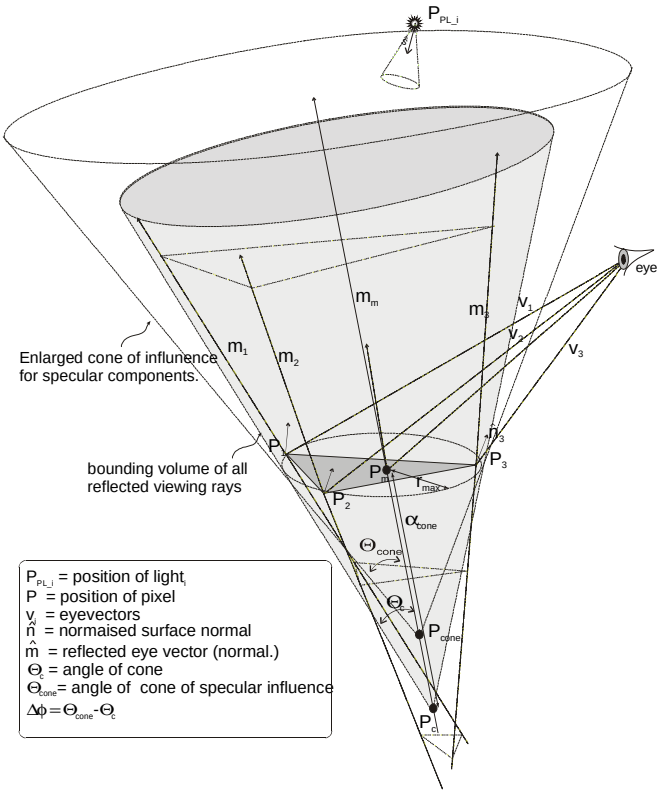

Fig. 3 Cone of specular influence

The cone vertex P_c has to lie on the m_m line passing through the center of the polygon $P_m \approx \frac{1}{3}\left(P_1 + P_2 + P_3\right)$

$$P_{cone} = m_m \cdot \left(\frac{r_{\max}}{\sin\left(\theta_{cone}\right)}\right) + P_m$$

The cone has to be positioned in a way that it includes the whole polygon. r_{max} is the Radius of the sphere with center P_m containing the polygon.

The overhead for the software driver is small because terms like m_i are also required for lighting calculation as in the $(m \cdot l)^{Srm}$ Phong lighting equation. Terms like $\Delta\phi$ are simple table lookups and r_{max} can be approximated. Further simplifications can be made by taking m_i instead of calculating m_m.

The condition for the specular relevance of a light source i is:

$$\cos(\theta_{cone}) \cdot \left|\left(P_{PL\_i} - P_c\right) \cdot m_m\right| \le \left(P_{PL\_i} - P_c\right) \cdot m_m$$

Note: $\theta_{cone} = \pi$ is known as backface culling.

2.2.3 Volume of influence of light sources

In the last paragraphs we followed the reflected viewing vectors to test the polygon for highlights.

Another way to determine the volume of influence is to look from the light source to the polygon.

The volumes of perceivable influence are defined by distance, and angular attenuation. For spot- lights the cut-off angle defines the conical volume of influence. For

specular highlights an implicit cut-off angle can be found as a function of the shininess:

$$\cos(\alpha_{\text{cutoff}}) = \left(\sqrt[s_{rli}]{threshold}\right)$$

The sphere of influence in the case of the Phong illumination model can be found by solving the quadratic equation.

$$\frac{1}{threshold} = k_{oi} + k_{1i}L_{\text{inf}} + k_{2i}L_{\text{inf}}^{\ 2}$$

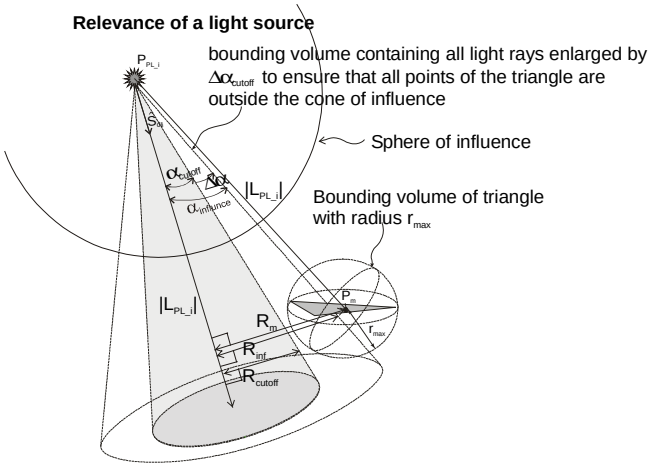

Fig. 4 Relevance of a light source

To ensure that no part of the polygon lies within the conical (directional light sources) and spherical volumes (point light sources) they can be enlarged by considering the bounding sphere of the polygon

The condition for which the polygon lies outside the cone of influence can be described as follows:

$$R_m > R_{cutoff} + r_{\max} > R_{\text{inf}}$$

Whereby r_{max} is the radius of the bounding sphere of the polygon, R_m is the length of the perpendicular line from P_m to the axis of the cone and R_{inf} is $|L|\sin(\alpha_{inf})$ (see Fig. 4).

$$R_{cutoff} = \left|L_{PL\_i}\right|\sin\left(\alpha_{cutoff}\right)$$

$$R_m^2 > (R_{cutoff} + r_{\max})^2 > R_{\text{inf}}^2$$

The condition simplifies to:

$$\left|L_{PL\_i}\right|^2 - \left(\hat{s}\cdot L_{PL\_i}\right)^2 > \left(\left|L_{PL\_i}\right|\sin\left(\alpha_{cutoff}\right) + r_{\max}\right)^2$$

$\sin(\alpha_{cutoff})$ is constant for a scene and r_{max} and P_m is calculated in the bounding cone method.

3. IMPLEMENTATION IN HARDWARE

Fig. 5 shows a possible hardware implementation for Adaptive Sample Shading.

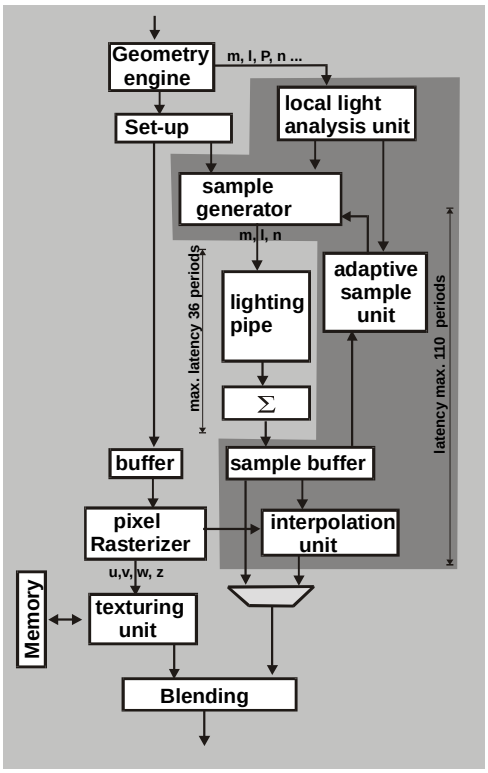

Fig. 5 A graphic pipeline with Adaptive Sample Shading (dark grey)

3.1 Local Light Analysis unit

This unit performs the tests per polygon to deliver useful information about the local lighting situation. It uses the information explained in section 2.3.2, 2.3.3 to select the relevant light sources.

3.2 Sample Generation Unit and Adaptive Sample Unit

The sample generator calculates the input parameters (m,l,n) for the lighting pipes at the regular or adaptive sample positions using the scanline output parameters.
The second task is to find an optimal schedule for the data flow within the lighting pipe.

Under the assumptions that:
- The regular sample distance is 8 pixels,
- the sample generator delivers one sample every fourth clock and
- the latency of the lighting pipe is assumed to be 20,

figure 6 shows an optimal scheduling for the lighting pipe utilization which is delivered by the adaptive sample unit. Furthermore the adaptive sample unit computes the newly inserted position of an adaptive sample and performs the threshold test.

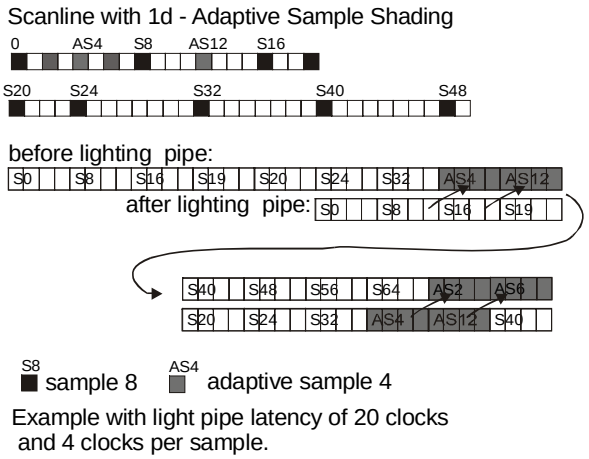

Fig. 6 Load balancing the light unit

Note that adaptive samples AS4, AS12 can only be calculated after regular sample S0 and S8 have passed the lighting pipe. An optimal schedule is found when AS4 is inserted after S32 according to Fig. 6.

3.3 Lighting Unit

As a reference implementation the DirectPhong lighting pipe of the VISA+ system [Graficon] is used. It implements the OpenGL 1.2 lighting equation [OpenGL] with the following modifications:
- the (**m**·**l**) –model is used instead of the (**n**·**h**) model,
- parameters are pseudo-float or integers,
- color accuracy is 12 Bit

Using these simplifications the lighting unit can be implemented with approx. 180K gates in 0.35 micron (80 MHz) and a latency of 36 clock-cycles. It delivers one illumination sample per clock.

3.4 Sample Buffer and Interpolation Unit

The samples are written into the sample buffer in the order as they are rendered and read out in the order required by the linear interpolators.
The interpolation unit interpolates the pixels between the samples.

4. ANALYSYS OF ADAPTIVE SHADING

4.1 Adaptive Sample Shading Simulation and triangle statistics

In this case study, a teapot, a sphere and a simple scene is used to demonstrate the capabilities of Adaptive Sample Shading without Adaptive Local Lighting analysis.

The teapot in Fig. 7 is shown to visualize the regular samples generated by the adaptive sample unit for a sample distance of 16 pixels. Fig. 8 shows the resulting image which is free of artifacts in critical regions.

Another image shows the originally Phong shaded sphere (see Fig. 10). Four lights are used to illuminate the sphere. These light sources are positioned in such a way that overlapping light sources form maximum regions for high gradient changes. Fig. 11 is generated using the Adaptive Sample Shading algorithm with a threshold set to 8.5% and the regular sample size was set to 16 pixel. The two pictures demonstrate that there is no visible difference between the original image and the sampled image. 19.25% of all pixel were per pixel Phong shaded.

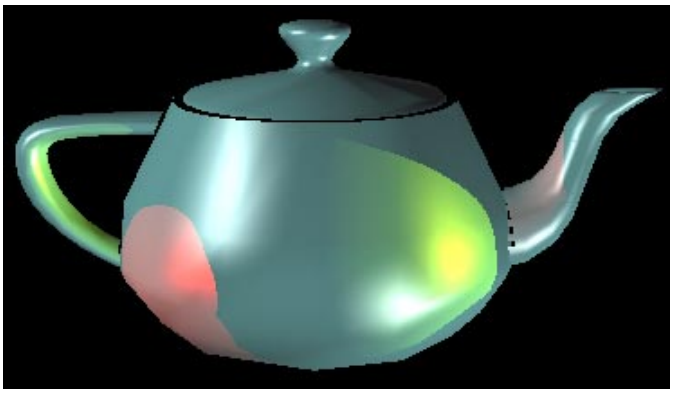

Fig. 7 One dimensional Adaptive Sample Shading with 31.4% samples

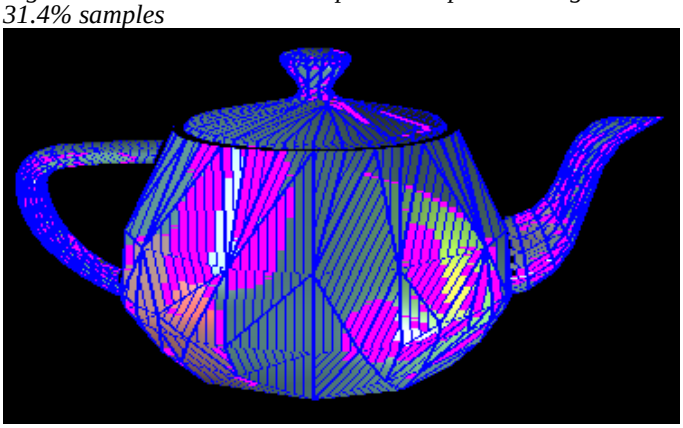

Fig. 8 *One dimensional Adaptive Sample Shading; regular samples blue, adaptive samples without binary subdivision in magenta*

*Fig. 13*2 is the same as Fig. 11, with all sampled pixels colored in red. The critical regions like large color gradient around highlights and cutoff-edges are reliably detected .

In

Fig. 13 four lights illuminate a scene with relatively large triangles. The shown Adaptive Sample Shaded picture only illuminates approx. 5% of all pixels. It can be seen that along the cut-off regions and regions where light sources overlap no artifacts are visible.

Fig. 14 and Fig. 15 demonstrate that there is no significant difference in the number of sampled pixels when the number of light sources increase.

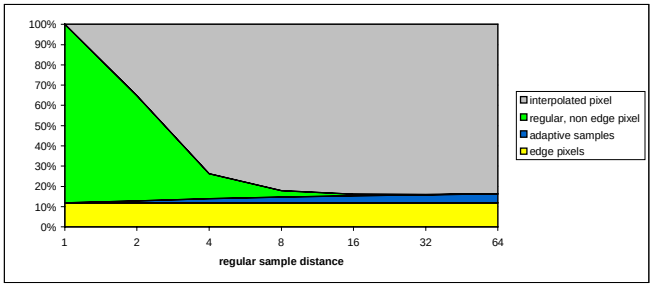

Fig. 9 Distribution of different sorts of pixel dependant on regular sample distance

Fig. 9 shows the illumination of different pixel types versus the regular sample size. The data was taken from the scene presented in Fig. 15. The pixels are classified into edge samples, adaptively placed samples by Adaptive Sample Shading, regular illuminated samples and interpolated pixels. The fixed number of edge pixel (for this scene is approximately 13%) represents the upper limit of the maximal achievable gain for this specific scene and implementation. The number of adaptively placed samples increase slowly when the regular sample distance is increased. The interpolated pixels demonstrate the overall gain. Best results can be archived with a regular size of 8-16 pixels.

Fig. 10 Original Phong shaded sphere

Fig. 11 Two dimensional Adaptive Sample shaded sphere with a quality threshold of 8.5%; 19.25% sampled

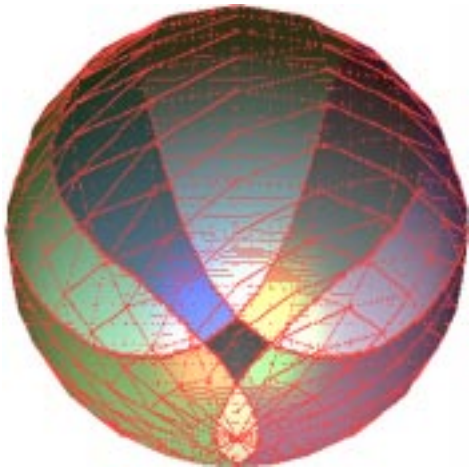

Fig. 12 all samples are coloured red; high colour gradients are found and better illuminated, quality threshold 8.5%, 19.25% sampled

Fig. 13 Simple scene with cube; threshold value is set to 8.5%; 5.12% sampled

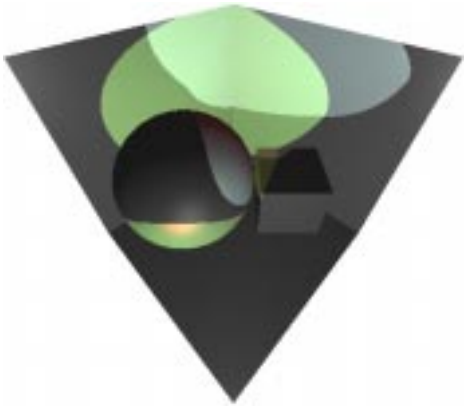

Fig. 14 Scene with two light sources; threshold value is set to 8.5%; 17.32% sampled

Fig. 15 Scene with 4 light sources; threshold value is set to 8.5%; 18.01% sampled

4.2 Adaptive Local Light simulation statistics

Fig. 16 and Fig. 17 demonstrated the affectivity of the bounding cone method. The left of the two scenes was rendered with the specular components adaptively switched off. Only 6% of the pixels had to be calculated with all components turned on.

Fig. 16 Left: scene computed with cone method; Right: visualisation of cone method: the darken the more specular parts are switched off

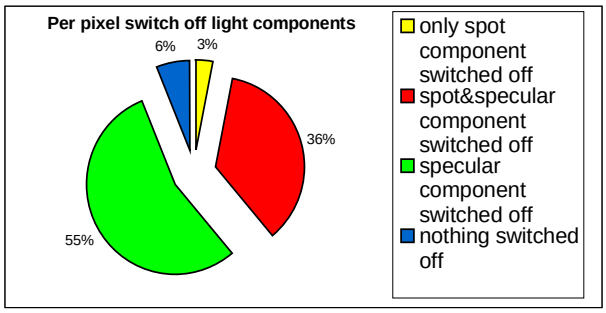

Fig. 17 Statistics for the local lighting analysis of scene.

5. CONCLUSIONS AND FUTURE WORK

This paper described an approach to support per pixel lighting without the performance penalty caused by the number of light sources used. Adaptive Sample Shading with local light analysis reduces the cost of shading by more than an order of magnitude compared to per pixel shading. Furthermore available animations have shown that the algorithms described are robust to animated light sources, small moving highlights and small spot angles. There are no suddenly disappearing light sources, machbanding or jitter effects.

Our future work will include the implementation of edge interpolation approaches to improve the performance for small triangles.

Significant improvements can be made introducing chunk-based rasterizers in combination with two dimensional Adaptive Sample Shading.

6. REFERENCES

[OpenGL] M. Segal, K. Akeley; "The OpenGL Graphics System: A Specification, Version 1.2", March 23, 1998

[TexRAM] A.Schilling, G. Knittel, and W. Strasser, "Texram: A smart memory for texturing"; IEEE Computer Graphics & Applications, 16(3), pp 32-41, May 1996

[Kirk] David Kirk, "Unsolved Problems and Opportunities for High-quality, High-performance 3-D Graphics on a PC Platform"; Keynote speech at EUROGRAPHICS Hardwareworkshop 98, Lisboa, August 98

[VISA] S. Budianto; O. Wittig; "VISA+: The Overall Sytem", Proceedings of GraphiCon97, pp 136-141; Moscow, May 21-24

[EnvBump] K. Bennebroek, I. Ernst, H. Russeler, O. Wittig; „Design Principles of Hardware-based Phong Shading and Bump-mapping"; Computer & Graphics, Vol.21, No.2, pp. 143-149, 1997

[Gour71] Gouraud, H. " Continuous Shading of Curved Surfaces", IEEE Trans. On Computers, C-20(6), June 1971, 623-629.

[Phong75] Phong, Bui-Tuong, "Illumination for Computer Generated Pictures" CACM 18 June 1975, 311-317.

[Benne] K.R.C. Bennebroek, "Design of a fast hardware implementation of the OpenGL Illumination Model for Surface Rendering Applications", Masters Thesis, 1996.

[Mill] Miller, Halstead, Clifton, "On-the-Fly Texture Computation for Real-Time Surface Shading", IEEE Computer Gaphics and Applications, March/April 1998.

[Neum] Youngkwan Cho, Ulrich Neumann, Jongwook Woo; "Improved Specular Highlights with Adaptive Shading"; Proceedings of Computer Graphics International, pp. 38-46, June 1996

[Tek] "Method of shading a graphics image"; European Patent Application EP 0 366 463 A2, Applicant: TEKTRONIX, INC; Inventors: J.C. Dalrymple, V.B. Sureshkumar Filing Date: 26.10.89

Acknowledgments

We are grateful to Alexander Weidt, S. Budianto, Alexander Huck and Peter Gober for their suggestions and contributions to this work.