Object-Space Variance Estimators

John Amanatides

Dept. of Computer Science
York University
North York ON M3J 1P3
Canada
amana@cs.yorku.ca

ABSTRACT

This paper describes an exploration of anti-aliasing techniques for ray tracing that glean variance information from object space rather than relying on samples to detect regions of aliasing.

Introduction

One of the biggest problems with ray tracing is that it is inherently a point-sampling process. Because of this anti-aliasing is problematic. The usual solution is adaptive supersampling [1], and it is used to reduce both aliasing and total computational cost. Unfortunately, adaptive point sampling is a post-sampling process and can be fooled. Because the samples themselves are affected by aliasing they may not trigger an increased sampling rate in problem regions.

What are needed are good estimators for variance, hopefully ones that are not as sensitive to aliasing. For example, when one is near a silhouette or on a textured object aliasing is more likely. If one knows this one can increase the sampling rates in these regions. One can imagine an advanced adaptive sampler that returns a variance estimator (along with the intensity value) when it samples the intensity function and uses this variance estimator to more effectively choose where to sample at higher resolution. In this paper we will explore this idea.

Ray Tracing and Anti-aliasing

Right from the beginning aliasing was a problem with ray tracing [1]. Whitted used several approaches to reduce it. First, he used adaptive supersampling. If the difference of neighbouring samples was above a threshold the region was recursively subdivided and more samples were taken. Second, he placed special bounding volumes around particularly small objects; if a ray intersected these bounding volumes the sampling rate was automatically increased to make sure that the object wasn't missed, at least for primary rays. Because of the advanced shading effects possible with ray tracing much effort was extended to reduce aliasing artifacts.

One early general approach of aliasing reduction was to extend the ray to allow for the gathering of neighbour information and thus to perform filtering; this included both cone tracing [2], beam tracing [3, 4] and pencil tracing [5]. Its dual, extending the objects by providing "covers" that indicate boundary regions was also explored [6]. All of these approaches required complex intersection computations which limited their usefulness.

The second general approach was to try to do the best one could with point sampling. It was observed that non-uniform point sampling replaced regular aliasing patterns with noise, which was visually less disconcerting [7, 8, 9]. As this required minimal changes to the ray tracer and produced significantly less noticeable aliasing this approach became very popular [10, 11, 12]. The general approach is to sample at a low rate and if the contrast or variance of neigbouring samples is high the sampling rate is increased in the local neighbourhood.

Towards Better Variance Estimators

In trying to reduce aliasing via adaptive supersampling one can use more than the variance of the samples to detect problem regions. For example, Cook would automatically increase the sampling rate in regions where objects were moving quickly [9]. If neighbouring samples land on different objects the probability of aliasing is increased. If one is near a silhouette then the probability of aliasing is increased. One simple local method of estimating this is by looking at the dot product of the surface normal and the eye vector. By looking at the curvature and the specularity of a surface one can detect regions of potential aliasing [13]. Almost invariably, if a region is textured there is a high probability of aliasing [14, 15]. If an object appears very small on the screen then a model that encodes levels of detail can both reduce computation and aliasing [16, 17].

In this paper we would like to propose a scheme where the ray tracer returns a variance estimator along with the intensity value. This variance estimator influences the local sampling rate. Different schemes for generating the variance estimates can be explored; it can be a simple as taking the dot-product of the surface normal and the eye direction or whether the surface is textured. However, what is really needed to do much of the above anti-aliasing is to estimate over how much of the viewing screen an object projects onto.

The relative size of the pixel area versus the object is very useful. This was why many of the early researchers tried to extend rays into cones or beams; the information gleaned could be used for anti-aliasing. This is straightforward for primary rays but is problematic for the rest as the rays quickly become distorted. Alas, the intersection algorithms were too expensive/complex. However, one does not have to go all the way with the solid angle approach. For example Genetti and Gordon [18] describe a variation of cone tracing where the cone is used for culling and for the detection of when an object completely surrounds a cone cross-section. If it does, one sample is taken; otherwise the partial intersection is adaptively supersampled.

In conjunction with this variance estimator we would also propose that we extend the ray into a solid-angle ray but keep the intersection algorithms unmodified; they would be the standard ray-object intersection algorithms. The only purpose of the solid angle ray is to determine the relative size of the object w.r.t. the pixel area. This will allow us to use many of the other anti-alising techniques described earlier in this section to help generate better variance estimators.

It should be noted that the "added value" of this variance estimator is to identify regions where adaptive supersampling would result in inadequate sampling rates and thus signal the need for increased sampling in these regions.

Conclusion

We have introduced a possible avenue of research that will allow one to compute better variance estimators when adaptively supersampling ray tracing. This should result in more appropriate sampling rates in problem regions and thus reduce aliasing.

References

1. T. Whitted, ‘‘An Improved Illumination Model for Shaded Display,’’ *Comm. of the ACM*, vol. 23(6), pp. 343-349, June 1980.
2. J. Amanatides, ‘‘Ray Tracing with Cones,’’ *Computer Graphics*, vol. 18(3), pp. 129-135, July 1984.
3. P. Heckbert and P. Hanrahan, ‘‘Beam Tracing Polygonal Objects,’’ *Computer Graphics*, vol. 18(3), pp. 119-127, July 1984.
4. A. Garcia, *Efficient Rendering of Synthetic Images*, PhD Thesis, Massachusetts Institute of Technology, February 1986.
5. M. Shinya, T. Takahashi, and S. Naito, ‘‘Principles and Applications of Pencil Tracing,’’ *Computer Graphics*, vol. 21(4), pp. 45-54, July 1987.
6. D. Thomas, A.N. Netravali, and D.S. Fox, ‘‘"Anti-aliased Ray Tracing with Covers,’’ *Computer Graphics Forum*, vol. 8(4), pp. 325-336.

7. M.A.Z. Dippe and E.H. Wold, ‘‘Antialiasing Through Stochastic Sampling,’’ *Computer Graphics*, vol. 19(3), pp. 69-78, July 1985.

8. M.E. Lee, R.A. Redner, and S.P. Uselton, ‘‘Statistically Optimized Sampling for Distributed Ray Tracing,’’ *Computer Graphics*, vol. 19(3), pp. 61-67, July 1985.

9. R.L. Cook, ‘‘Stochastic Sampling in Computer Graphics,’’ *Trans. on Graphics*, vol. 5(1), pp. 51-72, January 1986.

10. D.P. Mitchell, ‘‘Generating Antialiased Images at Low Sampling Densities,’’ *Computer Graphics*, vol. 21(4), pp. 65-72, July 1987.

11. J. Painter and K. Sloan, ‘‘Antialiased Ray Tracing by Adaptive Progressive Refinement,’’ *Computer Graphics*, vol. 23(3), pp. 281-288, July 1989.

12. M. McCool and E. Fiume, ‘‘Hierarchical Poisson disk sampling distributions,’’ *Proc. Graphics Interface 1992*, pp. 94-105, May 1992.

13. J. Amanatides, ‘‘Algorithms for the Detection and Elimination of Specular Aliasing,’’ *Graphics Interface ’92*, pp. 86-93, May 1992.

14. P.S. Heckbert, ‘‘Survey of Texture Mapping,’’ *IEEE Computer Graphics and Applications*, vol. 6(11), pp. 56-67, November 1986.

15. T. van Walsum, P.R. van Nieuwenhuizen, and F.W. Jansen, ‘‘Refinement critera for adaptive stochastic ray tracing of textures,’’ *Eurographics ’91*, pp. 155-166, September 1991.

16. J.H. Clark, ‘‘Hierarchical Geometric Models for Visible Surface Algorithms,’’ *Comm. of the ACM*, vol. 19(10), pp. 547-554, October 1976.

17. R. Hall, ‘‘Hybrid Techniques for Rapid Image Synthesis,’’ *Image Rendering Tricks, Siggraph ’86*.

18. J. Genetti and D. Gordon Ray Tracing With Adaptive Supersampling in Object Space, *Graphics Interface ’93*, pp. 70-77, May 1993.