Unbiased Sampling Techniques for Image Synthesis

David Kirk
California Institute of Technology
Computer Graphics 350-74
Pasadena, CA 91125

James Arvo
Program of Computer Graphics
Cornell University
Ithaca, NY 14853

Abstract

We examine a class of adaptive sampling techniques employed in image synthesis and show that those commonly used for efficient anti-aliasing are statistically biased. This bias is dependent upon the image function being sampled as well as the strategy for determining the number of samples to use. It is most prominent in areas of high contrast and is attributable to early stages of sampling systematically favoring one extreme or the other. If the expected outcome of the entire adaptive sampling algorithm is considered, we find that the bias of the early decisions is still present in the final estimator. We propose an alternative strategy for performing adaptive sampling that is unbiased but potentially more costly. We conclude that it may not always be practical to mitigate this source of bias, but as a source of error it should be considered when high accuracy and image fidelity are a central concern.

CR Categories and Subject Descriptors: I.3.7—[**Computer Graphics**]: Three-Dimensional Graphics and Realism; I.3.3—[**Computer Graphics**]: Picture/Image Generation;
General Terms: Algorithms, Graphics
Additional Key Words and Phrases: Adaptive Sampling, Anti-aliasing, Monte Carlo, Statistical Bias.

1 Introduction

Many of the sampling techniques employed in computer graphics are adaptive in the sense that they attempt to concentrate effort in areas where complexity is high. In particular, adaptive anti-aliasing schemes choose to sample at a higher rate where the scene is interesting, such as near edges. Many such schemes have been devised, both deterministic [11, 4] and stochastic [6, 2, 9, 7, 8]. The latter category has received the most attention and essentially consists of multi-stage Monte Carlo integration techniques. Common to all of these is the notion of using a small number of samples to detect regions where additional sampling is required to achieve a reliable answer, that is, one with an acceptably low level of noise.

While all of these methods have been reasonably successful in achieving this goal, it is important to understand the statistical effects of such a strategy. To do this we must examine multi-stage sampling plans *in toto* and characterize their statistical behavior. In particular, we wish to determine whether they in fact attain the correct answer on average.

Every stochastic anti-aliasing algorithm can be viewed as defining a random variable at each pixel to estimate the quantity of interest. This quantity is typically the unknown image function integrated with a filter kernel such as a gaussian or a box-filter. The purpose of adaptive sampling is to reduce the variance of these random variables, or *estimators*, with minimal increase in computation. If the expected value of an estimator is the solution we are seeking, it is said to be *unbiased*. If the estimator has a bias that can be made arbitrarily small, perhaps by increasing the number of initial samples sufficiently, then it is said to be *consistent* [3].

By analyzing the behavior of a prototypical multi-stage sampling algorithm operating on a simple class of test cases, we will show that most adaptive sampling plans fall into the category of consistent but biased estimators. Although the bias is typically small, this is a source of error that should be taken into consideration when high accuracy is required.

2 Common Sources of Bias

Sources of statistical bias can be found in many seemingly innocuous operations in image synthesis. For example, pixel values are frequently truncated or otherwise transformed so as to fall within the gamut of color monitors. Removing out-of-gamut colors can shift the distribution mean. At a very low level, the pseudo-random number generators at the heart of Monte Carlo approaches often have a built-in bias. At higher levels, the practice of *importance sampling* [5, 10] reduces variance by sampling more frequently where the result is large, which requires precise renormalization if the original expected value is to be maintained. Another example is the practice of truncating excessively deep ray trees in ray tracing. This can cause a systematic bias by eliminating a large number of small contributions [1].

In general, whenever we depart from naive Monte Carlo in an attempt to improve statistical efficiency, care must be taken to avoid introducing unnecessary bias. This is also true in screen space, for example, when anti-

```
EstimateMean(X, n, ε)
    begin
        Draw a set of n identically distributed random
        samples from X.
        S_n ← {X_1, X_2, ..., X_n};
        if Variation(S_n) ≤ ε then begin
            This is the "easy" case: use the sample
            mean as an estimate of the true mean.
            ξ ← mean(S_n);
            end
        else begin
            This is the "hard" case: invoke a costly
            oracle to compute the true mean.
            ξ ← TrueMean(X);
            end
        return ξ;
    end
```

Figure 1: A hypothetical adaptive sampling algorithm similar in spirit to most existing algorithms. This is biased for most inputs.

aliasing at the pixel level. As we show in the following section, adaptive anti-aliasing algorithms can introduce a systematic bias dependent upon the image function. This bias is greatest in areas of high contrast and is caused by early stages of sampling systematically favoring one extreme or the other. In Section 4 we propose a modified approach that is unbiased.

3 Bias From Adaptive Sampling

In this section, we examine the statistical behavior of common adaptive anti-aliasing algorithms. We begin by formulating a hypothetical sampling algorithm that retains the salient features of most multi-level sampling plans yet is simple enough to allow convenient analysis. The basic strategy is to use samples sparingly except where more work is deemed necessary. The decision to invoke a more costly method as a second stage is based upon a statistic we will call "variation," a function of the first-stage, or *pilot*, sample. This could be the sample variance, the "contrast", or a function of the sample size and variance as in [6] and [9]. An idealized algorithm using this strategy is shown in Figure 1, where X is the "population" whose mean we wish to estimate. For anti-aliasing, X will be the set of image values with probabilities influenced by the filter kernel.

Although the meaning of variation differs among the various approaches, a universal feature is that it goes to zero as the maximum deviation within the sample goes to zero. Thus, any such algorithm would be satisfied with only the first-stage sample when all values are identical.

If the variation is greater than some ϵ, then we will classify the population as "hard" to sample, and invoke a more expensive second-stage sampling technique. For simplicity we will assume here that the second-stage "oracle" computes the exact mean; in reality, this action would be simulated through a large number of samples. Because this ideal can be approximated to any given precision, our conclusions carry over to real algorithms, although the actual amount of bias will differ.

To demonstrate that the strategy in Figure 1 can be problematic we need only examine its behavior on a simple class of inputs. In particular, we will assume that X consists of a finite number of distinct values, $I_1, I_2, \ldots, I_k$, with corresponding probabilities $\omega_1, \omega_2, \ldots, \omega_k$. This situation occurs, for example, when applying a box filter to a pixel area consisting of k constant-intensity regions; the I's would represent the intensities within the pixel and the ω's would represent their fractional coverages. The actual mean is then

$$I = \sum_{i=1}^{k} I_i \omega_i. \tag{1}$$

With this characterization of X we can easily compute the expected value of the random variable ξ returned by the algorithm in Figure 1. Using conditional expectations based on a classification of "easy" or "hard", indicating that the variation of S_n is below or above the threshold ϵ, respectively, we have

$$\begin{aligned} \mathrm{E}[\xi] &= \mathrm{E}[\xi \mid \mathrm{easy}] \times \mathrm{Prob}[\mathrm{easy}] \\ &+ \mathrm{E}[\xi \mid \mathrm{hard}] \times \mathrm{Prob}[\mathrm{hard}] \end{aligned} \tag{2}$$

The oracle guarantees that $\mathrm{E}[\xi \mid \mathrm{hard}] = I$, the true mean. To analyze the conditional expectations we observe that any sample, S_n, can be characterized as a k-tuple, $(n_1, n_2, \ldots, n_k)$, where n_j is the number of samples assuming the value I_j. Then $n_1 + \cdots + n_k = n$ and the probability of a k-tuple is given by the multinomial distribution [3]:

$$\mathrm{Prob}[n_1, n_2, \ldots, n_k] = \frac{\omega_1^{n_1} \omega_2^{n_2} \cdots \omega_k^{n_k}}{n_1! n_2! \cdots n_k!} n!. \tag{3}$$

Using this fact, we can compute $\mathrm{E}[\xi]$ for any input of the form described above. We simply step the algorithm through all distinct k-tuples and sum the resulting values of ξ weighted by the corresponding probabilities. However, if we assume ϵ to be sufficiently small that S_n will be classified as "easy" only when all n samples are of the same value, then Equation 2 reduces to a very simple expression. In this case we have

$$\mathrm{Prob}[\mathrm{easy}] = \omega_1^n + \omega_2^n + \cdots + \omega_k^n \tag{4}$$

and the expected value of ξ, given that the initial sample was found to be "easy", is

$$\mathrm{E}[\xi \mid \mathrm{easy}] = \frac{\omega_1^n I_1 + \cdots + \omega_k^n I_k}{\omega_1^n + \cdots + \omega_k^n}. \tag{5}$$

Substituting these into Equation 2 and observing that $\mathrm{Prob}[\mathrm{hard}] = 1 - \mathrm{Prob}[\mathrm{easy}]$ we arrive at the expression

$$\mathrm{E}[\xi] = I + \sum_{i=1}^{k} \omega_i^n (I_i - I). \tag{6}$$

Because I is the true mean, the summation on the right of Equation 6 is the amount of bias. This will be nonzero for all but a small class of inputs. The bias diminishes as the number of initial samples increases, indicating that the estimator is consistent. In Section 5 we present experimental data obtained from Equation 6.

4 An Unbiased Adaptive Sampling Plan

The hidden flaw in the algorithm above is that the first-stage samples deemed "easy" are not completely random, and therefore may not fairly represent the entire population. That is, the test for accepting a first-stage sample is usually correlated in some way with the mean of the sample.

There is a straightforward modification of the above sampling plan to avoid this bias. First select a small subset of the area X, call it R, and draw a sample of size n from this subset. We may examine this sample to determine the number of samples to draw from the rest of the region, $X-R$, but in any case we use the initial sample mean to estimate the mean of R. Because we do not alter the estimate of R, no bias is introduced there. Also, because the second stage is simply a choice among two or more unbiased estimators for a disjoint region, it also remains unbiased. It follows that a weighted sum of the these sample means, weighted proportionately by area, results in an unbiased estimate over the entire region. This approach is outlined in Figure 2.

As with any multi-stage scheme, the goal is to estimate the population variance by means of a first-stage sample. To the extent that the region R is representative of the entire region, drawing the pilot sample from it will serve this purpose. This suggests that R should be "scattered" throughout X.

As a special case of this strategy, note what happens if we allow the area of region R to shrink to zero. The result is a strategy whereby the pilot sample is used solely to select the sample size for the second stage - not for estimating the mean. This clearly avoids any possibility of a correlation between the estimator of the mean and the variation of the pilot sample.

These examples suggest a simple rule that will avoid introducing bias in multi-stage sampling schemes: decide how a sample is going to be used *before* it is drawn - not based on the actual values drawn. Observing this rule prevents us from modifying estimates in any way that may be correlated with the result.

This technique can be applied in a hierarchical fashion and stratified, similar to [6]. After the first decision has been made based on the pilot sample, we can make additional decisions later, provided that we either discard the samples used to influence the strategy, or decide ahead of time that they will be used to estimate the mean over the subregion from which they were drawn.

The main disadvantage of using a technique such as this is that it is difficult to avoid either wasting samples or producing a high-variance result that cannot be remedied. The former occurs if R is chosen to be so small that the pilot sample contributes very little to the final estimate. The latter occurs if R is large and the pilot sample fails to provide a sufficiently reliable estimate of its mean. We are then left with a poor estimate. Improving it with further sampling will, in most cases, alter the distribution of the "easy" cases and introduce bias.

UnbiasedEstimateMean$(X, R, p, n_1, n_2, \epsilon)$
begin

Draw a set of p identically distributed random samples from $R \subset X$.

$S_p \leftarrow \{X_1, X_2, \ldots, X_p\} \subset R$;

if Variation$(S_p) \leq \epsilon$ **then** $n \leftarrow n_1$
else $n \leftarrow n_2$;

Draw a set of n identically distributed random samples from the rest of X.

$S_n \leftarrow \{X_1, X_2, \ldots, X_n\} \subset X - R$

Compute ξ based on the unbiased estimates of the two disjoint components.

$\xi \leftarrow \overline{S_p} \times |R| + \overline{S_n} \times |X - R|$;

return ξ;
end

Figure 2: An unbiased adaptive sampling algorithm. It is assumed that $R \subset X$ and $n_1 < n_2$.

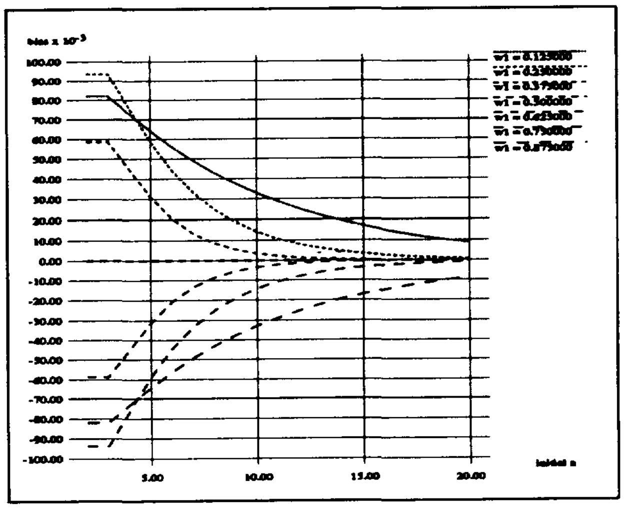

Figure 3: Absolute Bias as a function of initial sample size n for a collection of fractional areas.

5 Results

To study the extent of the biasing problem we have computed the exact bias introduced by the algorighm in Figure 1 for a range of initial sample sizes and a variety of two-intensity pixels. In this case Equation 6 provides the actual bias. Both Figures Fig. 3 and Fig. 4 show curves for ω_1 ranging from 0.125 to 0.875. For each of these curves, $\omega_2 = 1 - \omega_1$, $I_1 = 0$, and $I_2 = 1$. Note that while the absolute bias is symmetric about zero, the percent bias increases as the actual mean decreases.

While these figures are informative, it is difficult to see how this really affects an image. Fig. 5a (upper left)

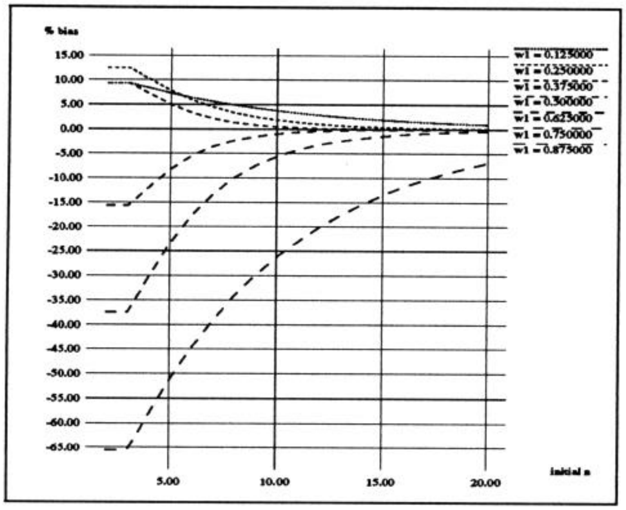

Figure 4: Percent Bias as a function of initial sample size n for a collection of fractional areas.

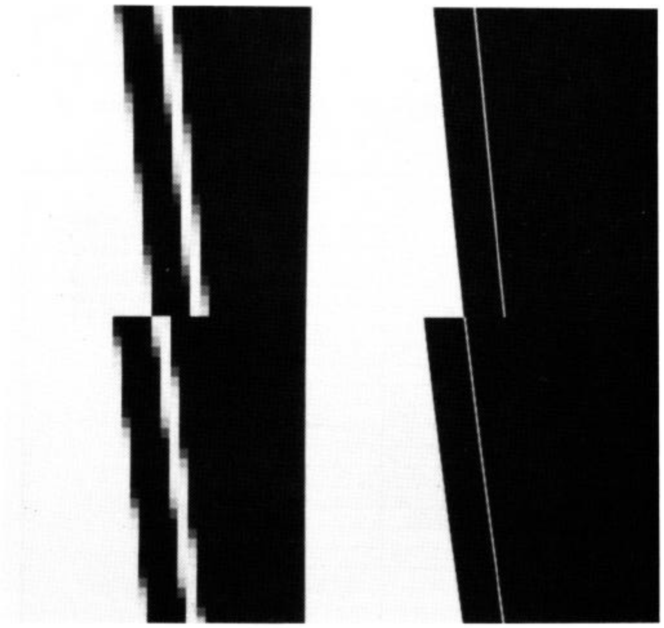

Figure 5: a) Unbiased image, b) Unbiased image (hi-res), c) Biased image, d) Biased image (hi-res)

shows a black / white edge, and a thin white polygon on a black background, at 32x32 resolution. Fig. 5b (upper right) is a hi-res version of Fig. 5a. Fig. 5a & b were computed using the expected value of the unbiased algorithm from Figure 2. Fig. 5c & d were computed using the expected value of the biased algorithm described in Figure 1. Therefore, these images illustrate the *tendencies* of these algorithms, not actual results. Pixels in all figures were integrated using a box filter.

Note that in Fig. 5c & d, the familiar "roping" in the antialiased edges is worse than in Fig. 5a & b. This is because the small partial coverages in each pixel are underestimated in the biased approach. Likewise, the large partial coverages are overestimated. In this case, the bias accentuates problems with antialiasing of edges.

6 Conclusions

We have shown that common adaptive anti-aliasing algorithms can be statistically biased, and have proposed an alternative algorithm that is unbiased.

It may not always be worthwhile to remove this source of bias. The error is typically small, especially when the initial sample is large. Our alternative sampling plan, while unbiased, possesses other drawbacks in terms of additional cost and parameter selection. For each application the cost must be weighed against the benefit of improved accuracy.

The analysis presented here has identified a subtle deficiency hidden within most anti-aliasing approaches which should be addressed in future schemes.

Acknowledgements

Much of this research was performed while the authors were employed at Apollo Computer and Hewlett-Packard. The authors also wish to thank the anonymous reviewers for their thoughtful and detailed comments.

References

[1] Arvo, James, and David Kirk, "Particle Transport and Image Synthesis," *Computer Graphics*, 24(4), August 1990, pp. 63-66.

[2] Dippe, Mark A. Z., and Erling Henry Wold, "Antialiasing through Stochastic Sampling," *Computer Graphics*, 19(3), July 1985, pp. 69-78.

[3] Freund, John E., and Ronald E. Walpole, *Mathematical Statistics*, 4th edition, Prentice Hall, New Jersey, 1987.

[4] Glassner, Andrew S., "An Overview of Ray Tracing," in *An Introduction to Ray Tracing*, A. S. Glassner, ed., Academic Press, New York, 1989.

[5] Kajiya, J. T., "The Rendering Equation," *Computer Graphics*, 20(4), August 1986, pp. 143-150.

[6] Lee, Mark E., Richard A. Redner, and Samuel P. Uselton, "Statistically Optimized Sampling for Distributed Ray Tracing," *Computer Graphics*, 19(3), July 1985, pp. 61-68.

[7] Mitchell, Don P., "Generating Antialiased Images at Low Sampling Densities," *Computer Graphics*, 21(4), July 1987, pp. 65-69.

[8] Painter, James, and Kenneth Sloan, "Antialiased Ray Tracing by Adaptive Progressive Refinement," *Computer Graphics*, 23(3), July 1989, pp. 281-288.

[9] Purgathofer, W., "A Statistical Method for Adaptive Stochastic Sampling," in *Proceedings of Eurographics 86*, ed. A.A.G. Requicha, Elsevier, North-Holland, 1986, pp. 145-152.

[10] Rubinstein, R. Y., *Simulation and the Monte Carlo Method*, J. Wiley, New York, 1981.

[11] Whitted, Turner, "An Improved Illumination Model for Shaded Display," *Communications of the ACM*, 32(6), June 1980, pp. 343-349.