Statistically Optimized Sampling for Distributed Ray Tracing

Mark E. Lee
Amoco Production Company
Tulsa Research Center

Richard A. Redner
Department of Mathematics

Samuel P. Uselton
Department of Computer Science

University of Tulsa
Tulsa, Oklahoma

Abstract

Cook, Porter, and Carpenter coined the phrase "distributed ray tracing" to describe a technique for using each ray of a super-sampled ray tracing procedure as a sample in several dimensions to achieve effects such as penumbras and motion blur in addition to spatial anti-aliasing. The shade to be displayed at a pixel is a weighted integral of the image function. The purpose of using many rays per pixel is to estimate the value of this integral. In this work, a relationship between the number of sample rays and the quality of the estimate of this integral is derived. Furthermore, the number of rays required does not depend on the dimensionality of the space being sampled, but only on the variance of the multi-dimensional image function. The algorithm has been optimized through the use of statistical testing and stratified sampling.

CR Categories and Subject Descriptions: I.3.3 [Computer Graphics]: Picture/Image Generation - display algorithms; I.3.7 [Computer Graphics]: Three-dimensional Graphics and Realism - Shading, Shadowing, Texture, Visible Line/Surface Algorithms;

Additional Keywords and Phrases: Ray Tracing, Anti-aliasing, Penumbras, Shadows, Translucency, Transparency

Introduction

The problem of visual artifacts appearing in synthesized images due to the finite resolution of displays has been a concern in computer graphics for many years. Several techniques, collectively referred to as anti-aliasing, have been developed for combatting the problem [3,5,6,7,9,10,14,16]. Only recently, however, has this problem been a paramount concern in the implementation of ray tracing algorithms [1,4,11,15]. All solutions to the problem begin with the same theoretical basis [5]. The correct value to display at a pixel is a weighted integral of shades in the neighborhood of the pixel. Direct computation of this integral is expensive, especially in color displays, since it must be done for at least three primary values. Several authors have suggested sampling points or areas around the pixel as a method for approximating this integral. An important question is "How many samples is enough?". The two standard answers are "more is always better" and "we find that n is usually enough," where n is some integer between 4 and 256. Given that color values are stored with limited precision, it seems likely that the number of useful samples per pixel is also limited. If the sampling is done in some fixed pattern, then geometries always exist for which that particular sampling pattern generates a poor estimate of the integral and unwanted artifacts are created. However, if the sampling is done randomly, this problem can be eliminated. Furthermore, a statistical test can be developed to determine when enough samples have been used. The following section describes the derivation of this result.

In the implementation of a ray tracing algorithm, this result can be used to estimate the number of rays needed to accomplish spatial anti-aliasing. It is also demonstrated that the same result holds when the ray is considered to be a multidimensional sample. It applies, then, to implementations which use each ray as a sample, not only of the pixel area, but also of light source area, surface reflection direction, surface refraction direction, and similar variables. The number of samples required do not depend directly on the number of dimensions being sampled, therefore, sampling an additional dimension may not increase the number of samples needed. A statistical technique known as stratified sampling is used to select samples such that, if the variation among the first samples is small, no further samples are required. This technique is used to reduce sampling, as in adaptive subdivision algorithms [2,15], without disturbing the statistical properties required for the main result to remain valid.

In the results section, several images are presented to demonstrate the quality that can be achieved. To show the efficiency of the implementation, a two-dimensional histogram of the number of sample rays for each pixel are given for each image. In addition, histograms showing the number of pixels requiring various amounts of sampling are provided.

Theory

In the following, X represents the point in multidimensional space to be sampled. F(X) is the "true" continuous image to be approximated. P(X) is the filter used in smoothing the image to accomplish the anti-aliasing.

For $X \varepsilon R^n$, we must evaluate convolutions which can be written as integrals of the form $\int_{R^n} F(X)P(X)dX$ [5]. In the case that $P(X) \geqq 0$ for $X \varepsilon R^n$ and $\int_{R^n} P(X)dX = 1$, then P(X) is a probability density function. If we think of X as an n-dimensional random variable with probability density function P(X), then the value of the integral $\int_{R^n} F(X)P(X)dX$ is the expected value of F which is written E(F(X)).

Rather than estimating this integral using traditional numerical techniques, we propose a statistical estimate. Let $X_1, X_2, \ldots, X_N$ be independent identically distributed random variables with density function P(X). Let

$$F_N = (1/N) \sum_{i=1}^{N} F(X_i).$$

If E(F(X)) exists, then by the strong law of large numbers [8],

$$\lim_{N\to\infty} F_N = E(F(X)) \quad \text{with probability one,}$$

and so for sufficiently large N, F_N is a good estimate of E(F(X)). We also observe that

$$E(F_N) = E(F(X))$$

which means that F_N is an unbiased estimator for E(F(X)).

A statistical measure of the difference between F_N and E(F(X)) is the variance of F_N which is defined to be

$$E\ (F_N - E(F(X)))^2 = (E(F^2(X))-E(F(X))^2)/N$$
$$= VAR(F(X))/N.$$

We observe that not only does F_N converge to E(F(X)) as N gets large but that the variance of F_N about E(F(X)) is VAR(F(X))/N.

Two points of this formulation should be emphasized. First, just as in more traditional numerical schemes for evaluating integrals, the error can be made arbitrarily small by evaluating the function at sufficiently many points. In our problem, these points are chosen randomly with density P(X). The second important point is that the error in our estimate of E(F(X)) is not intrinsically a function of the dimension of the space but depends only on the variation of F over those dimensions. The number of samples to be thrown is not the product of the number of samples required for each dimension; instead, the number of samples depends directly on the variability of F.

It is reasonable to try to construct a sampling scheme so that the variances of our estimates throughout the scene are approximately equal. Ideally, we would determine a threshold, T, and sample until the variance of our estimate was less than T. However, since the variance at each point is not known a priori, we construct the following statistical test. Let

$$S_N^2 = (1/N) \sum_{i=1}^{N} (F(X_i)-F_N)^2.$$

S_N^2 is an approximation of VAR(F(X)) from the generated data. Define the number $\chi_\beta^2(N-1)$ so that, under a normal sampling theory,

$$\text{Probability } (N{*}S_N^2/VAR(F(X)) < \chi_\beta^2(N-1)) = \beta.$$

This notation has been chosen because the distribution of $N{*}S_N^2/VAR(F(X))$ under the normal theory is the χ^2 distribution with N-1 degrees of freedom [8]. Compute $S_N^2 / \chi_\beta^2(N-1)$. If $S_N^2 / \chi_\beta^2(N-1) < T$, then stop sampling, otherwise, throw more samples. This test is constructed so that the

$$\text{Probability } (VAR(F(X))/N < S_N^2/\chi_\beta^2(N-1)) = 1-\beta$$

and so the probability of stopping when VAR(F(X))/N > T is less than β. The threshold, T, is chosen sufficiently small so that the variance of the computed values is small enough to provide a good estimate of the true color values. Since this is a statistical setting, one can not guarantee that a good estimate will be calculated at every pixel. In order to assure that a good estimate is obtained most of the time, the parameter β, the failure rate, is chosen to be small.

Further consideration of the error of our estimate shows that for certain geometries, the variance of the estimate can be reduced by the use of stratified sampling. Stratification denotes selection from several subregions into which a region is divided [12]. In particular, n dimensional Euclidean space can be partitioned and samples thrown into each region. Consider the case that R^n is divided into m regions and that N_j samples $\{X_{ij}\}_{i=1}^{N_j}$ $j=1,\ldots,m$ are thrown into each region where the numbers N_j/N are proportional to the probability of the jth region. The estimate of the average value of F is the same as before and by using a stratified sample of this type, the variation of F_N can be reduced. Let

$$N = \sum_{j=1}^{m} N_j \text{ and } F_j = (1/N_j) \sum_{i=1}^{N_j} F(X_{ij}).$$

Now

$$F_N = (1/N) \sum_{j=1}^{m} N_j F_j .$$

The variance of this estimate is the weighted average of the variances over each region of R^n [12], that is,

$$VAR(F_N) = (1/N) \sum_{j=1}^{m} N_j VAR(F_j).$$

If the variances of F over the individual regions are sufficiently small compared to the variance over the whole space, then there will be a reduction in variance.

Consider a linear boundary between two homogenous regions in the plane and a density function which is radially symmetric about a point on the boundary in the plane (Figure 1). If the plane is divided into four equal quadrants about this point and stratified sampling is performed, then it can be seen that in the worst case, the variance of the estimate is reduced by a factor of 2 and in the best case (the boundary is aligned with the partition), the variance is (almost magically) equal to zero.

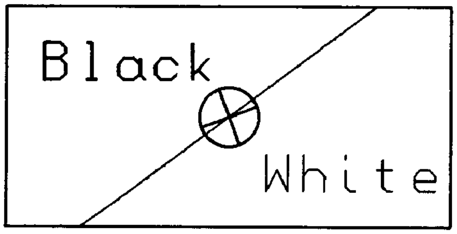


Figure 1

Implementation

To compute the pixel values in a distributed ray tracing implementation, samples are drawn. As the samples are created, incremental sums of $F(X_i)$ and $F^2(X_i)$ are kept. The sampling continues until the exit criterion $S_N^2/\chi_\beta^2(N-1) < T$ is met.

An important issue is how to choose the values for T and β. Let M be the maximum or worst case variance that will be tolerated for a scene. Any estimate to the variance that is greater than M should never pass the early exit test and should force the maximum amount of samples to be used. To force this to happen, T should be set so that, at the maximum number of samples, the correct balance between the variance and the χ^2 test will be met and the test will pass. Let Z be the maximum number of samples to be allowed. This value is usually determined by computer run time constraints. T can be calculated from the maximum variance and the maximum number of samples by the following formula

$$T = M/\chi_\beta^2(Z-1).$$

Now, when the variance of the samples is greater than or equal to the maximum variance allowed, the maximum number of samples are guaranteed to be used. Define Δ to be the minimum color difference that can be represented by the display medium. For a raster device, the minimum color difference is usually $1/(2^b)$ where b is the number of bits in the frame buffer for a primary color. If the difference between two colors is less than Δ, then the difference cannot be displayed. A value for the maximum variance, M, should be of the same magnitude as the minimum difference in color, Δ.

A table containing the values of $T*\chi_\beta^2(N-1)$ for $N=1,\ldots,Z$ can be calculated in a preprocessing step for maximum efficiency. The early exit test becomes a table lookup and a comparison. The value of β now serves to spread the values of the table between zero and M. The larger the value for β, the larger the spread of values becomes and the earlier the variance can pass the test. A tradeoff exists between a lower β value and more accuracy or a higher β value and less overall sampling and computing time. The value chosen for β is a consequence of the quality required for a particular application and the amount of resources available.

Stratified sampling allows for sampling a distribution with a good approximation to the true mean with fewer samples. The region to be sampled is broken into several smaller regions such that the combination of the distributions of the smaller regions is the same as distribution of the original region. If the region is broken into smaller regions in a sensible manner, a very few samples will serve to properly sample the region of interest. The key is to set up the division of the region so that the samples are as well distributed as possible while still following the distribution function accurately. This prevents using many samples that lie near each other and yields a better estimate of E(F(X)).

Each distributed ray is n-dimensional in information content. Several of the dimensions include pixel area sampling, realistic light source sampling, and surface smoothness sampling for reflection and refraction. The ray definition is implicit instead of explicit, however. Instead of carrying an n-dimensional ray through the ray tracer, an ordinary 3-space ray is used along with the implicit n-dimensional information for calculating an n-dimensional ray from an ordinary 3-dimensional ray. The ray is extended to higher dimensions by perturbing the ray using the implicit information in the proper fashion. For anti-aliasing, generate an artificial two-dimensional axis system in the window containing the pixel. Draw a sample from the appropriate distribution for anti-aliasing. Use the samples as offsets from the origin along each axis and make this point the new endpoint of the ray. The new ray has been extended to sample from the proper distribution for anti-aliasing. For modeling realistic light sources, first project the light source onto the plane perpendicular to a ray arriving at the center of the light source. Sample the distribution of the light source in the plane. The value of this random variable specifies the endpoint of the sample ray. Variable surface smoothness can be sampled by perturbing each reflected and refracted ray by using the plane that is perpendicular to the end of the ray and sampling as above. After all perturbations have taken place, the ray now contains n-dimensional information and properly samples each dimension.

Results

Three figures are provided to demonstrate the quality of the results. In each figure, part (a) is the actual image computed. Part (b) is the two-dimensional histogram showing the number of samples used for the corresponding pixels of part (a). The histogram in part (c) shows the relative quantity of pixels for each number of samples used. All three images are sampled in the dimensions required to model solid light sources and variable degrees of surface smoothness and to perform spatial anti-aliasing. All three images use the shading model of Lee and Uselton described in [13], and are done at a resolution of 512 by 512 pixels.

Parameter settings for all three pictures are the same. Eight subregions are used for the stratified sampling and one new sample is chosen from each region when the need for additional samples is indicated. The value for β used is .05 and the value for T is 0.000105. The maximum variance, M, is 1/128, and the maximum number of samples, Z, is 96.

Figure 2(a) shows nine wedges lit from the right, casting shadows onto a checkered backdrop. The shadows show penumbras caused by the sampling of the light source. Figure 2(b) shows that large numbers of rays are used only when smaller numbers will not suffice. Notice that fewer rays are needed in the area of the blackest wedge because all the light is absorbed and the shade becomes constant. Figure 2(c) demonstrates two interesting facts: first that most pixels require only the minimum number of samples; and second that the high frequency of the checkerboard pattern does (as expected) cause a large number of pixels requiring the maximum number of samples.

Three effects of this algorithm can be seen in Figure 3(a) especially well. The shadows cast by the metallic spheres have penumbras, showing the sampling in the light source dimension. The reflection of the checkered backdrop onto the table becomes less precise the further out from the backdrop it is, showing a less than perfectly smooth surface. The highlights on the two spheres have different areas and intensities, indicating a difference in polish between the two. Figure 3(b) shows the variation in the number of samples per pixel required, especially in the reflection of the backdrop on the table. The lack of variation in the area of the backdrop is due to the extremely matte finish of the backdrop. Figure 3(c) shows that the larger squares and matte finish of the backdrop leads to a larger number of pixels requiring only the minimum number of samples.

In figure 4(a) both the table and the backdrop are smoother than in figure 3(a) and the backdrop is more reflective. This is shown by the visibility of the reflection of the table on the backdrop and the visibility of the light source reflection on the backdrop. The reflection of the backdrop on the table is slightly more precise. This difference can actually be seen more easily in figures 3(b) and 4(b) by comparing the smearing of the backdrop's edges in the reflection on the table. Note also the reflection of the shadow of the wine glass.

Future work

This technique of generating sample rays can easily be extended to additional dimensions to model effects such as motion blur and depth of field. Wavelength sampling for improved color modeling as well as wavelength dependent effects such as refraction can also be included. The difficulties in these extensions lie mainly in determining the appropriate distribution to sample for each dimension and computing the ray paths in higher dimensions. Computing the position of moving objects at arbitrary times, for example, will be required for motion blur.

Additional work should consider the interaction between the number of rays per pixel, the number of pixels in the image and the size at which the image is to be displayed. It is intuitively expected that an increase in spatial resolution will decrease the average number of samples needed per pixel. This intuition is dependent on the assumption that the overall image size remains constant. The size of the pixel, in terms of the portion of the field of view occupied, becomes a relevant parameter.

Acknowledgements

The authors would like to express gratitude to the Tulsa Research Center of Amoco Production Company for the use of their computing facilities, to the referees for their helpful comments, and to Debra Redner for her support of this research effort.

References

[1] Amanatides, J. Ray tracing with cones. Computer Graphics 18,3 (July 1984), pp. 129-135.

[2] Catmull, E. Computer display of curved surfaces. Proceedings IEEE Conference on Computer Graphics, Pattern Recognition and Data Structures (May 1975).

[3] Catmull, E. A hidden-surface algorithm with anti-aliasing. Computer Graphics 12,3 (Aug. 1978), pp. 1-5.

[4] Cook, R. L., Porter, T. and Carpenter, L. Distributed ray tracing. Computer Graphics 18,3 (July 1984), pp. 137-145.

[5] Crow, F. C. The aliasing problem in computer-generated shaded images. Comm. ACM 20,11 (Nov. 1977), pp. 799-805.

[6] Crow, F. C. The use of grayscale for improved raster display of vectors and characters. Computer Graphics 12,3 (Aug. 1978), pp. 1-5.

[7] Crow, F. C. A comparison of anti-aliasing techniques. IEEE Computer Graphics and Applications 1,1 (Jan. 1981), pp. 40-49.

[8] Feller, W. An Introduction to Probability Theory and Its Applications. John Wiley and Sons, 1971.

[9] Fiume, E., Fournier, A. and Rudolph, L. A parallel scan conversion algorithm with anti-aliasing for a general-purpose ultracomputer. Computer Graphics 17,3 (July 1983), pp. 141-150.

[10] Fuchs, H. and Barros, J. Generating smooth 2-d monocolor line drawings on video displays. Computer Graphics 13,2 (July 1979), pp. 260-269.

[11] Heckbert, P. S. and Hanrahan, P. Beam tracing polygonal objects. Computer Graphics 18,3 (July 1984), pp. 119-127.

[12] Kish, L. Survey Sampling. John Wiley and Sons, 1965.

[13] Lee, M. E. and Uselton, S. P. A shading model for rendering objects with body color. Technical Report F85-C-5, Amoco Production Company - Tulsa Research Center, Tulsa OK (1985).

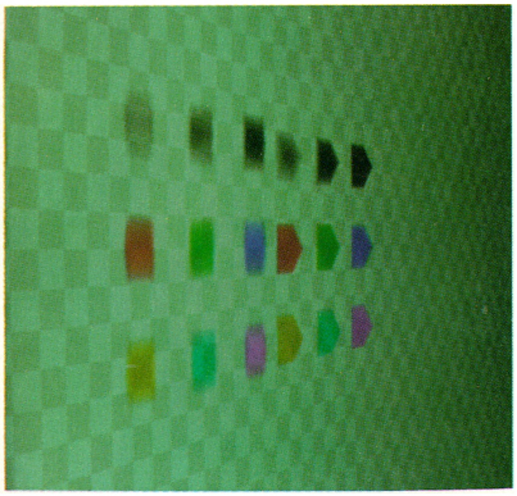
[14] Turkowski, K. Anti-aliasing through the use of coordinate transformations. ACM Transactions on Graphics 1,3 (July 1982), pp. 215-234.

[15] Whitted, T. An improved illumination model for shaded display. Comm. ACM 23,6 (June 1980), pp. 343-349.

[16] Whitted, T. Anti-aliased line drawing using brush extrusion. Computer Graphics 17,3 (July 1983), pp. 151-156.

Figure 2

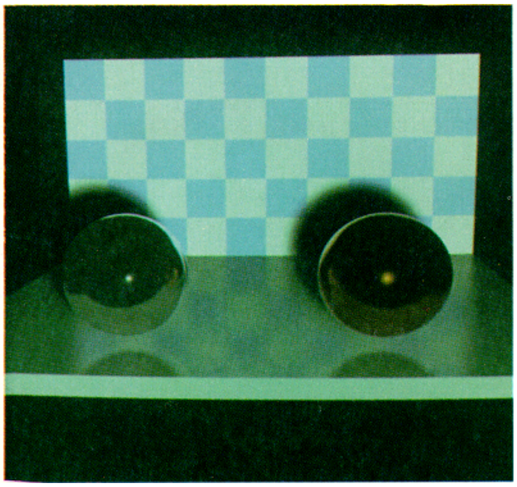
9 wedges



(a)

Figure 3

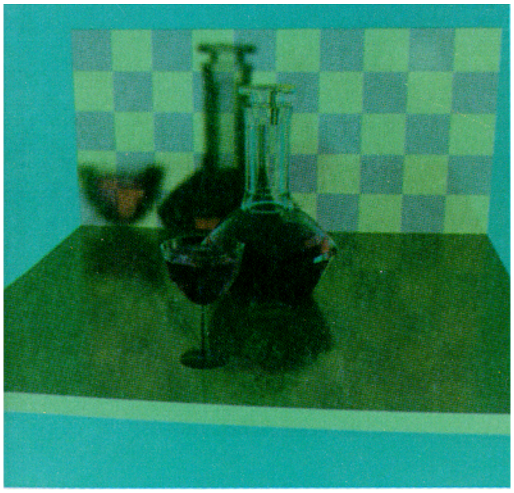
2 metallic spheres



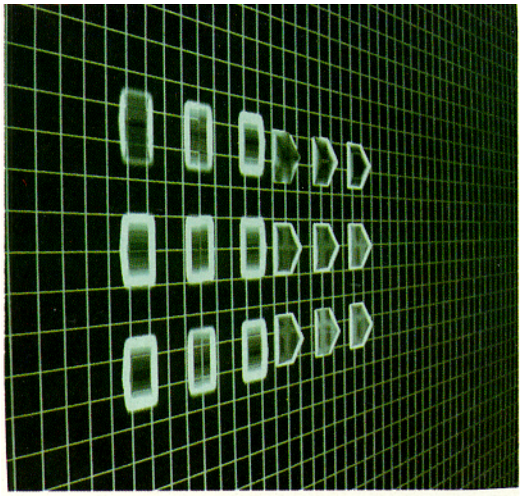
(a)

Figure 4

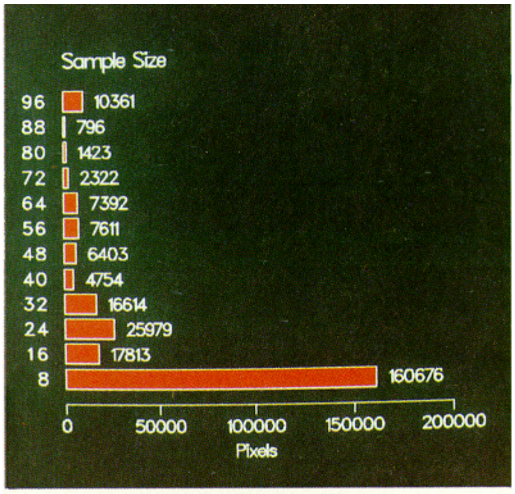
wine glass and decanter



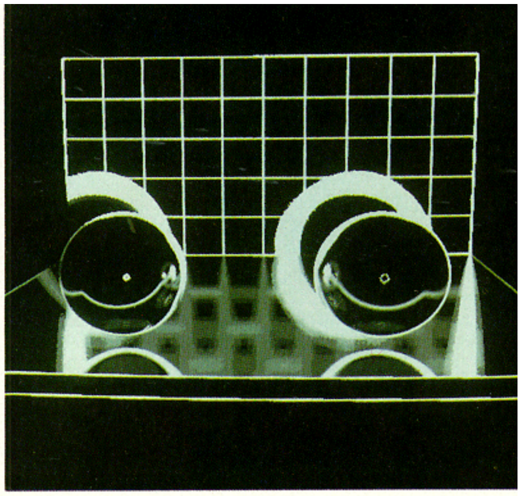
(a)



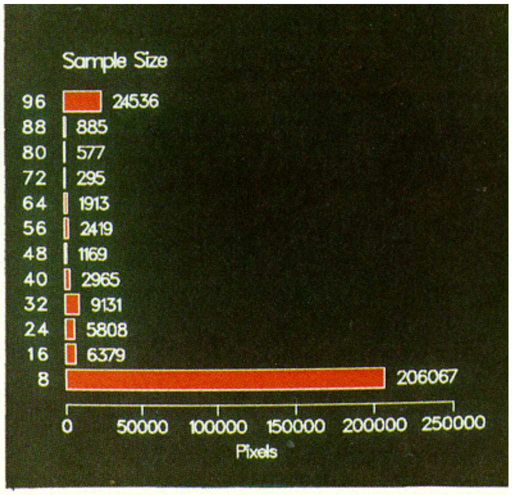
(b)


Sample Size
96
10361
88
796
80
1423
72
2322
64
7392
56
7611
48
6403
40
4754
32
16614
24
25979
16
17813
8
160676
0
50000
100000
150000
200000
Pixels

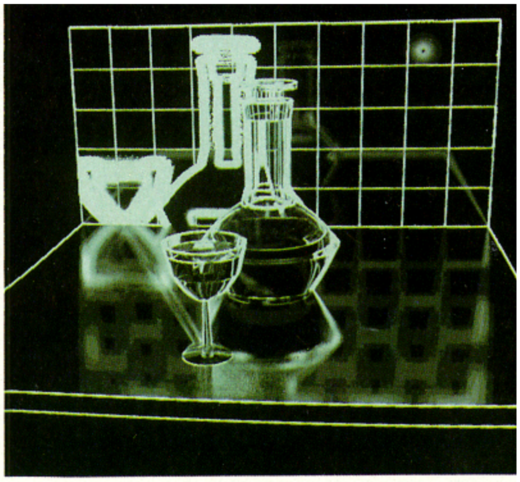
(c)



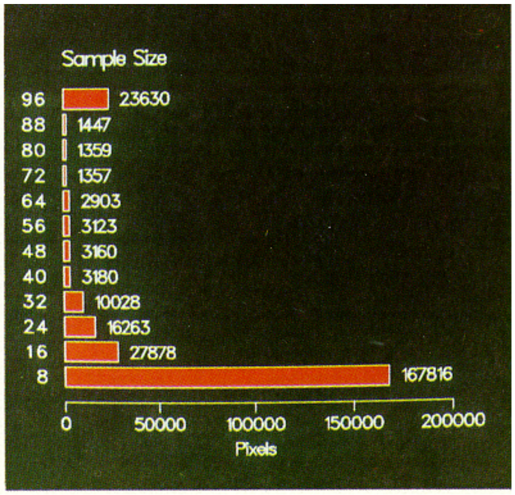
(b)


Sample Size
96
24536
88
885
80
577
72
295
64
1913
56
2419
48
1169
40
2965
32
9131
24
5808
16
6379
8
206067
0
50000
100000
150000
200000
250000
Pixels

(c)



(b)


Sample Size
96
23630
88
1447
80
1359
72
1357
64
2903
56
3123
48
3160
40
3180
32
10028
24
16263
16
27878
8
167816
0
50000
100000
150000
200000
Pixels

(c)