Figure 5.12: Room with one shadow ray per viewing ray.

5.5 Strategies for Non-adaptive Sampling

In the last section, the integrals used to calculate pixel brightness were phrased in terms of stratified Monte Carlo sampling. This sampling occurred on the two dimensional spaces of pixel area, lens area, light source area, and reflection ray direction. Each of these two dimensional spaces was sampled using uncorrelated jittering, where each of the spaces was fully stratified, and the stratum of each space was paired with a stratum of each other space in an irregular manner.

A crucial part of this sampling process is intelligently selecting the sampling points. The easiest way to select sample points is to set a predetermined number of samples for each pixel, and select a pattern of that many points in a two dimensional probability space for each of the

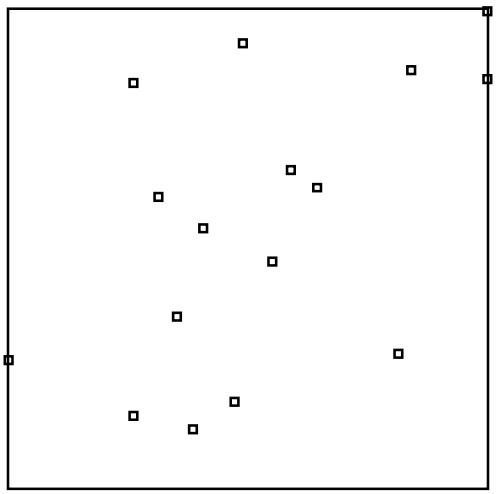

Figure 5.13: 16 random sample points

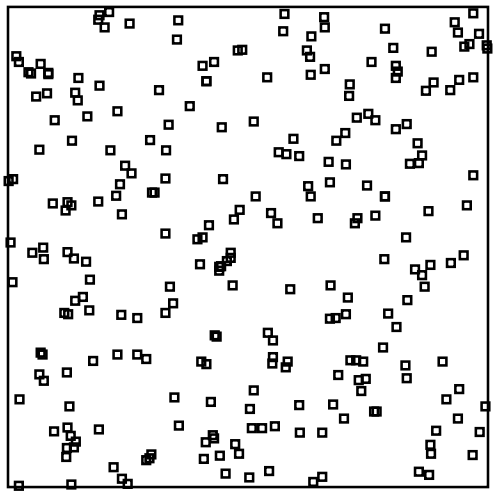

Figure 5.14: 256 random sample points

two dimensional spaces of the integral. This section examines how to best arrange these points once their number is known. The next section discussed adaptive methods.

5.5.1 Random Sampling

The simplest way to choose N points (x_i, y_i) from the 'canonical' two dimensional probability space is to pick every x_i and y_i independently by setting them to canonical pairs (ξ_i, ξ_i'). A set of 16 random pairs is shown in Figure 5.13, and a set of 256 random pairs is shown in

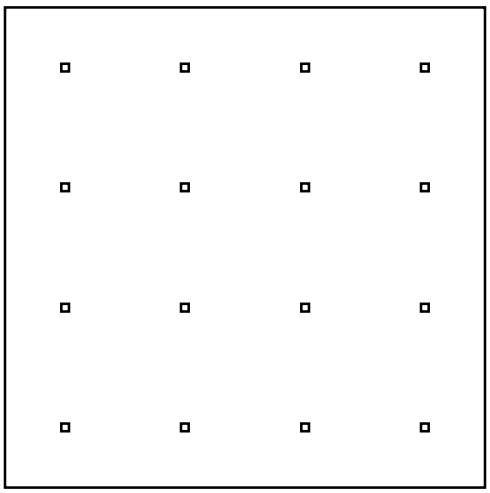

Figure 5.15: 16 regular sample points

Figure 5.14. As can be seen in these figures, random sampling allows some areas to be sparsely sampled, and others to be densely sampled. This implies that the variance in estimates using random sampling will be high.

5.5.2 Regular Sampling

We could simply place the pairs evenly in a grid pattern (as is done in traditional quadrature methods), as shown in Figure 5.15. This will prevent clumping of samples, but may introduce spatial regularity into the error across many pixels.

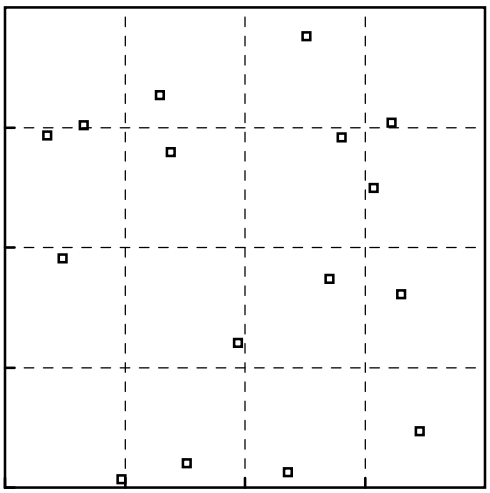

Figure 5.16: 16 jittered sample points with one sample per square

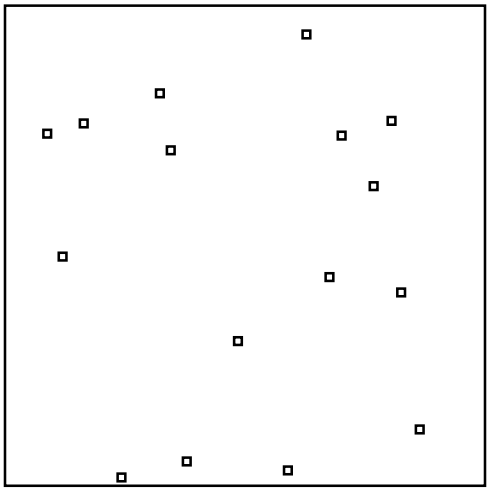

Figure 5.17: 16 jittered sample points

5.5.3 Jittered Sampling

Jittered sampling is another name for classical Monte Carlo stratified sampling[26]. Typically, the canonical probability space is partitioned into an n by n set of equal area squares, and a point is chosen uniformly from each square. A jittered pattern of 16 samples is shown in Figure 5.16, and without the square boundaries in Figure 5.17. A pattern of 256 points is shown in Figure 5.18.

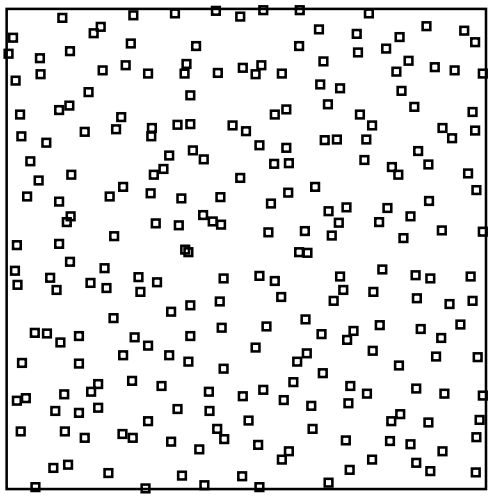

Figure 5.18: 256 jittered sample points

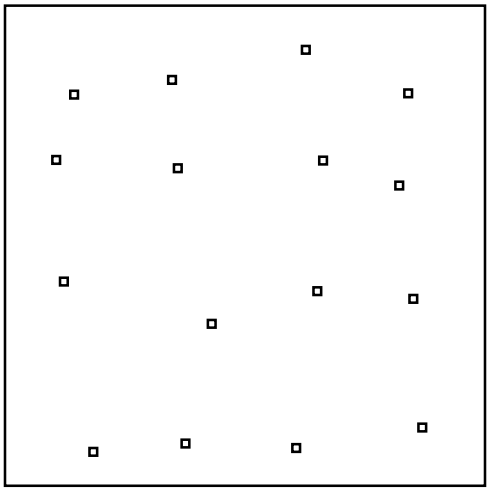

Figure 5.19: 16 semijittered (0.5 maximum shift) sample points

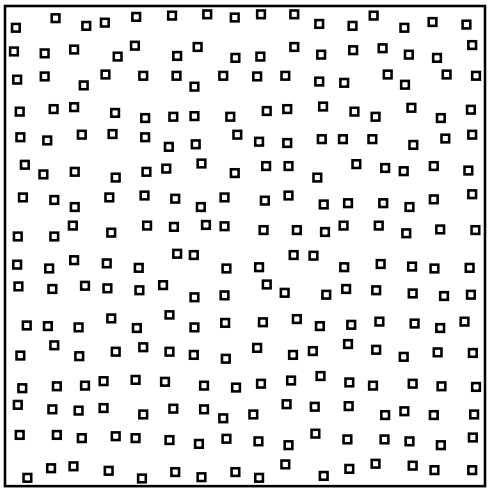

Figure 5.20: 256 semijittered (0.5 maximum shift) sample points

One problem with jittering is that limited clumping can take place. This problem can be lessened by choosing points nearer to the center of each square. Cook did this by using a Gaussian distribution to choose from each square. A simpler version of this is shown in Figure 5.19, where the samples are *half-jittered*: points are chosen uniformly within the square, half the width of the full square. A set of 256 half-jittered sample points is shown in Figure 5.20. As can be seen, there is less clumping, but more regularity than with simple jittering.

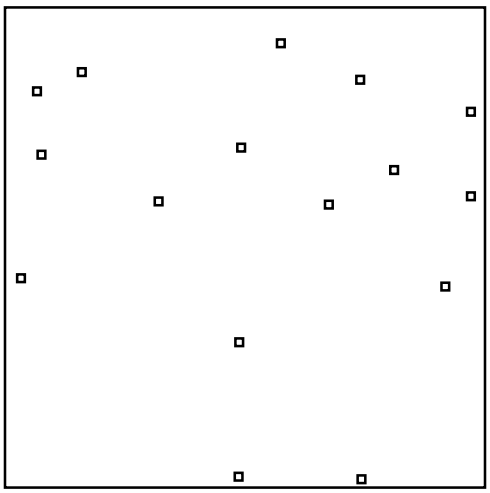

Figure 5.21: 16 Poisson disk sample points (minimum separation 0.1 pixel width)

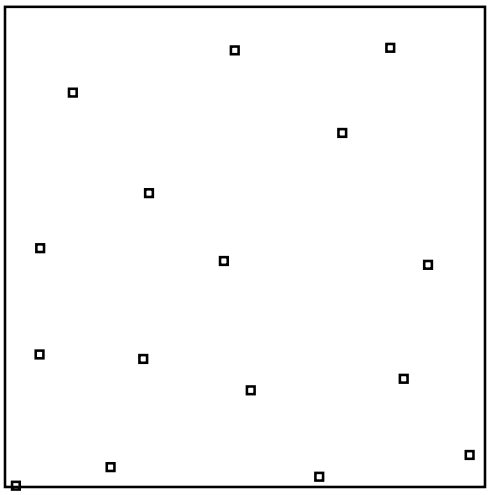

Figure 5.22: 16 Poisson disk sample points (minimum separation 0.2 pixel width)

5.5.4 Poisson Disk Sampling

A simple way to avoid clumping of sample points is to generate a sequence of samples, and reject a new sample if it is too close to an existing sample. This method, called Poisson disk sampling, has often been used in ray tracing applications for this reason[30, 25]. A set of 16 samples with minimum separation 0.1 is shown in Figure 5.21, and with minimum separation 0.2 in Figure 5.22. Poisson disk distributions of 256 samples are shown in Figures 5.23 and 5.24.

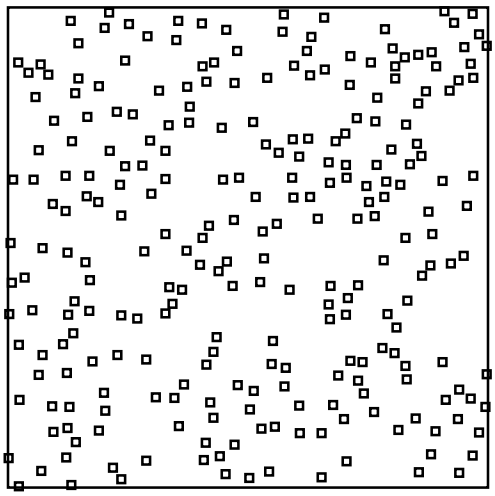

Figure 5.23: 256 Poisson disk sample points (minimum separation 0.025 pixel width)

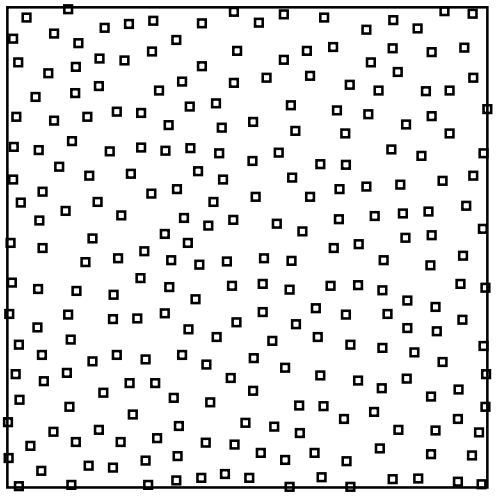

Figure 5.24: 256 Poisson disk sample points (minimum separation 0.05 pixel width)

As can be seen, there is less clumping for large minimum separation, but too large a separation can cause ghosting or even make sampling impossible.

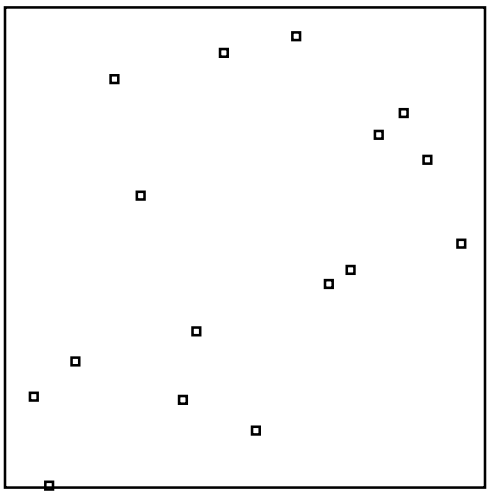

Figure 5.25: 16 separately (uncorrelated jitter) generated sample points

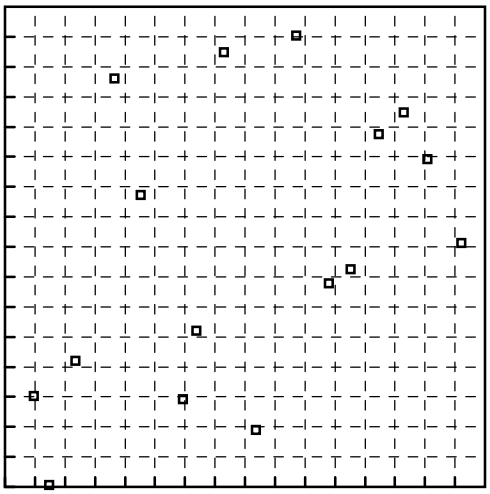

Figure 5.26: 16 separately generated sample points with guidelines

5.5.5 N-rooks Sampling

As discussed in Section 5.4.1, each of the dimensions of a sampling space can be separately partitioned and the strata of each dimension can be randomly associated so that each row and column of a grid will have one sample. A set of 16 samples of this type are shown in Figure 5.25. The underlying pattern of the samples is hard to see, unless a grid is superimposed as shown in Figure 5.26. A particularly descriptive name for this strategy is *N-rooks sampling*, because a acceptable set of sample cells will be the squares of an N by N chessboard with N rooks that

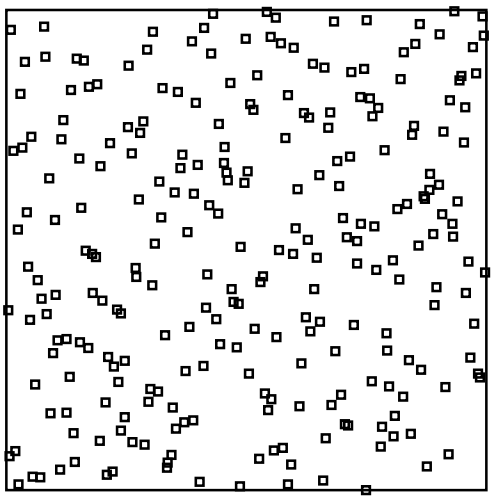

Figure 5.27: 256 separately (uncorrelated jitter) generated sample points

cannot capture each other in one move. There are two advantages of this type of sampling over conventional jitter. The first is that any number of samples can be taken with uncorrelated jitter, while conventional jitter usually requires a $m \times m$ or $m \times n$ pattern. The second is that if the sampled field varies only in one dimension, that dimension is fully stratified (just as in traditional distributed ray tracing).

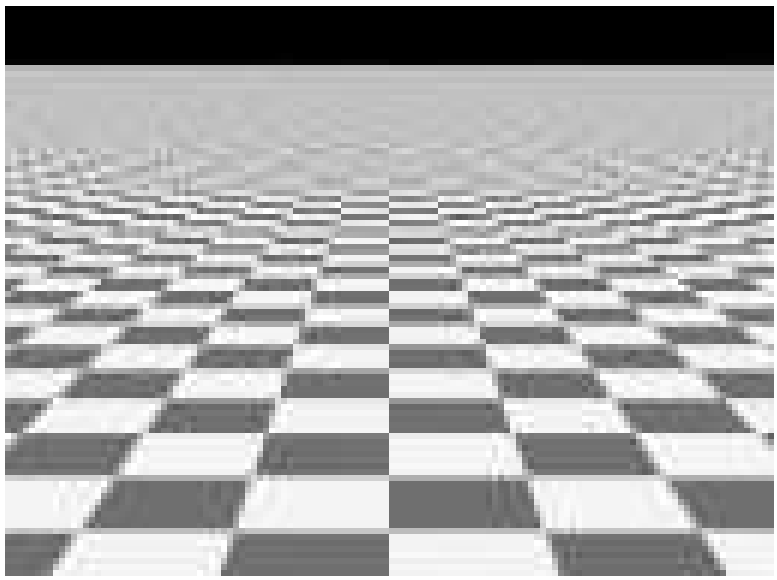

Figure 5.28: Test figure CHECKER

5.5.6 Empirical Behavior of Sampling Strategies

Each of the previous sampling methods was tested with 16 samples per pixel on each of four 128 by 96 pixel test images. All of the images were sampled with a one pixel width box filter. The first image, called CHECKER, shown in Figure 5.28, is that of an infinite checkerboard. Since the uncorrelated jitter is well suited to horizontal and vertical lines, the second test figure, CHECKER-45 (Figure 5.29), is the same checkboard rotated 45° to avoid such lines. The third figure, BALL (Figure 5.30), is a ball lit by two area light sources. The final figure, ALL (Figure 5.31), has a ball, a specular mirror, a glossy mirror, and a finite aperture lens. All four of the figures shown were sampled with 400 jittered samples, which is sufficient to produce an image with relatively small error.

Some of the sampling methods produced regular errors that were visually disturbing. On the checkerboards this was particularly true; the regular and semijittered sampling had visible aliasing artifacts. Figure 5.32 shows the regular sampling of CHECKER-45. The banding

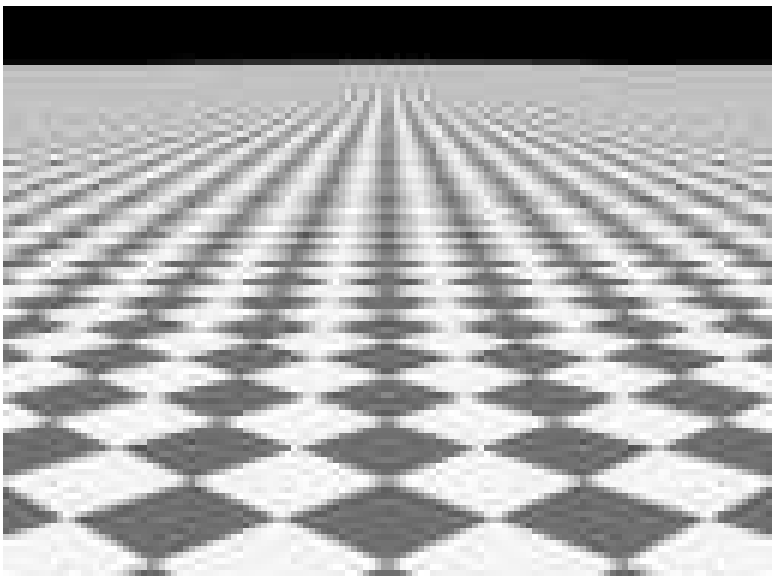

Figure 5.29: Test figure CHECKER-45

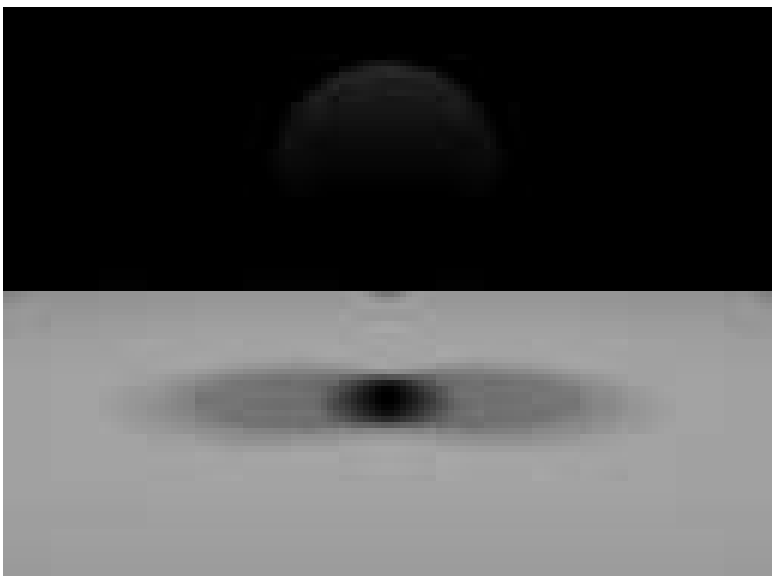

Figure 5.30: Test figure BALL

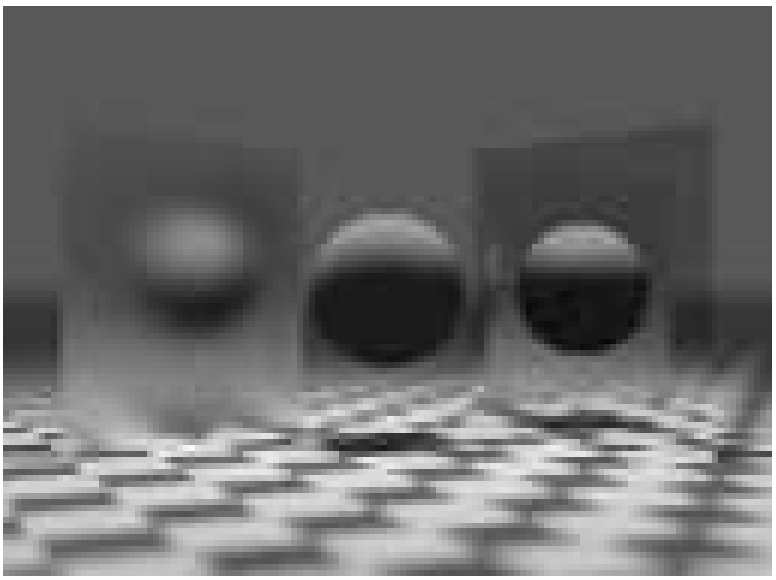

Figure 5.31: Test figure ALL

near the horizon is not present in the same image in the separately sampled image shown in Figure 5.33. This is an example of the common problem that it is the regularity of the error, rather than its magnitude, that is objectionable.

The average absolute error in luminance for each sampling strategy is listed for each test image in Tables 5.1 through 5.4. Surprisingly, the separate (uncorrelated jitter) sampling performed best by the average error metric on three of the four images, and on test image ALL was only outperformed by regular and half-jittered strategies, both of which are prone to aliasing. The standard deviation and maximum error caused by separate sampling also performs well relative to the other sampling strategies.

5.5.7 Theoretical Behavior of Sampling Strategies

In the last section it was demonstrated that for some images the separate sampling strategy performs quite well, even compared to Poisson disk sampling. Overall, the performance rankings of the strategies (from best to worst) was approximately separate, jittered, half-jittered, regular,

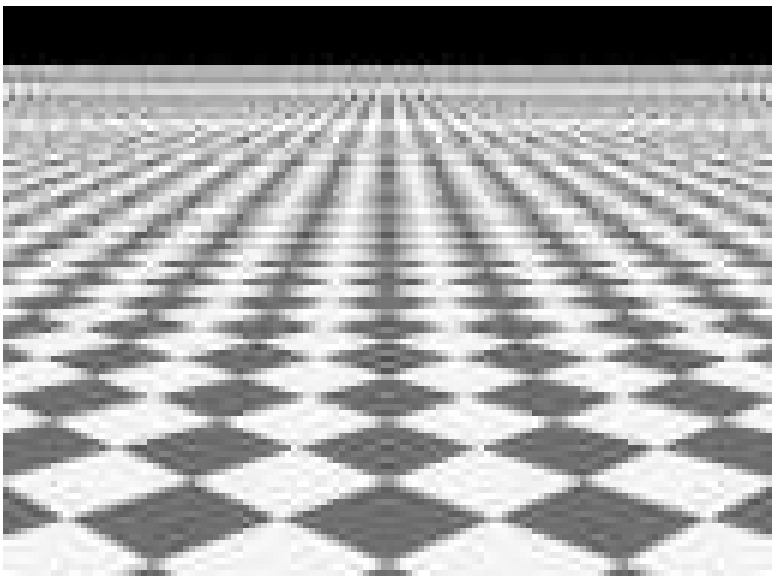

Figure 5.32: Test figure CHECKER-45 with 16 regular samples per pixel

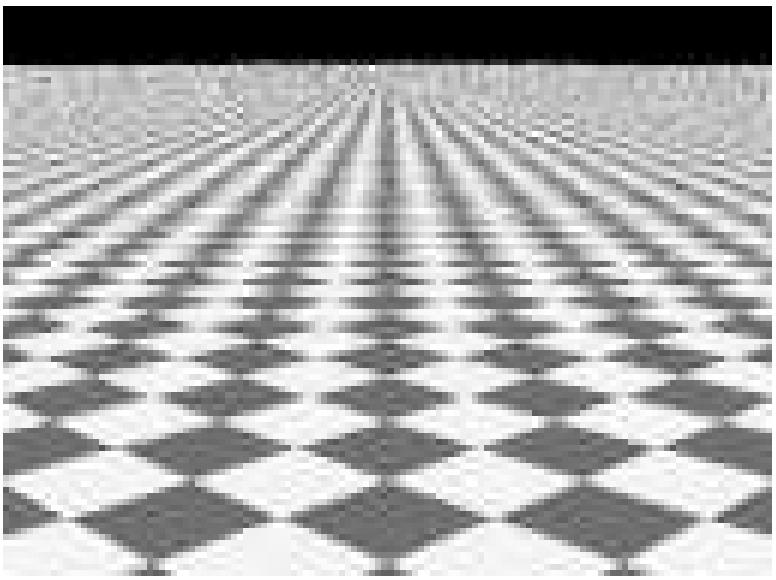

Figure 5.33: Test figure CHECKER-45 with 16 separate samples per pixel

sampling method	*ave(|E|)*	*SD(|E|)*	*max(|E|)*
separate	0.0163	0.0303	0.308
jittered	0.0216	0.0368	0.394
poisson ($d = 0.2$)	0.0221	0.0362	0.334
poisson ($d = 0.1$)	0.0259	0.0413	0.308
half-jittered	0.0263	0.0437	0.368
random	0.0303	0.0479	0.331
regular	0.0312	0.0526	0.390

Table 5.1: Pixel errors in luminance for CHECKER

sampling method	*ave(|E|)*	*SD(|E|)*	*max(|E|)*
separate	0.0190	0.0296	0.286
jittered	0.0221	0.0340	0.291
poisson ($d = 0.2$)	0.0225	0.0343	0.333
poisson ($d = 0.1$)	0.0281	0.0416	0.304
half-jittered	0.0237	0.0445	0.410
regular	0.0273	0.0532	0.450
random	0.0315	0.0466	0.294

Table 5.2: Pixel errors in luminance for CHECKER-45

sampling method	*ave(|E|)*	*SD(|E|)*	*max(|E|)*
separate	0.00324	0.0099	0.179
regular	0.00363	0.0113	0.177
half-jittered	0.00365	0.0114	0.151
jittered	0.00370	0.0110	0.160
poisson ($d = 0.2$)	0.00404	0.0123	0.170
poisson ($d = 0.1$)	0.00526	0.0180	0.226
random	0.00607	0.0214	0.266

Table 5.3: Pixel errors in luminance for BALL

sampling method	*ave(|E|)*	*SD(|E|)*	*max(|E|)*
regular	0.0137	0.0235	0.242
half-jittered	0.0139	0.0234	0.205
separate	0.0148	0.0246	0.245
jittered	0.0150	0.0251	0.259
poisson ($d = 0.2$)	0.0156	0.0256	0.247
poisson ($d = 0.1$)	0.0172	0.0284	0.236
random	0.0190	0.0315	0.287

Table 5.4: Pixel errors in luminance for ALL

sampling method	*ave(D)*	*SD(D)*	*max(D)*
separate	0.162	0.0237	0.229
half-jittered	0.184	0.0187	0.243
jittered	0.193	0.0288	0.291
poisson ($d = 0.2$)	0.196	0.0332	0.290
regular	0.234	0.0000	0.234
poisson ($d = 0.1$)	0.245	0.0447	0.357
random	0.282	0.0557	0.428

Table 5.5: Discrepancies of different sampling strategies

sampling method	*ave(S)*	*SD(S)*	*max(S)*
half-jittered	0.0463	0.00290	0.0537
separate	0.0467	0.00847	0.0812
jittered	0.0495	0.00192	0.0678
poisson ($d = 0.2$)	0.0540	0.00891	0.0844
regular	0.0600	0.00000	0.0600
poisson ($d = 0.1$)	0.0743	0.02140	0.1740
random	0.0877	0.02390	0.2080

Table 5.6: Root mean square discrepancies of different sampling strategies

Poisson disk, and finally random. It is possible that these results are closely tied with filter choice, and that idea merits further investigation. The poor performance of Poisson disk goes against conventional wisdom. It would be nice to establish a quantitative metric for predicting the value of a particular strategy. It would also be a good idea to understand what we have done by using sample sets that are not strictly random, since presumably we are doing a Monte Carlo integration. In numerical integration theory these non-random sample sets are called *quasi-random*, because they have some statistical qualities that make them acceptable substitutes for true random samples. Zeremba developed a theory to bound the error of an integration based on equidistribution properties of the sample set (assuming certain continuity properties of the integrand)[126]. In this section, Zeremba's equidistribution metric, *discrepancy*, is discussed in the context of the sampling strategies from the last section.

sampling method	*ave(T)*	*SD(T)*	*max(T)*
separate	0.2132	0.0236	0.275
jittered	0.2555	0.0397	0.428
poisson ($d = 0.2$)	0.2613	0.0459	0.390
half-jittered	0.2608	0.0282	0.338
poisson ($d = 0.1$)	0.2921	0.0503	0.513
random	0.3434	0.0540	0.485
regular	0.3600	0.0000	0.360

Table 5.7: Stroud's discrepancies of different sampling strategies

The concept behind discrepancy is that we'd like a number that is small for very equidistributed sample sets, and large for poorly distributed sets. Imagine a set of N sample points, (x_i, y_i) on the unit square. Given a point (a, b) on the square, the set of points (x, y) such that $x < a$ and $y < b$ will define a smaller square (with lower left corner $(0, 0)$ and upper right corner (a, b)) with area ab. Let n be the number of the N sample points that falls within that smaller square. If the sample points are reasonably equidistributed, we would expect n/N to be about ab. Zeremba uses this observation to define the discrepancy, D, as the lowest upper bound of $|n/N - ab|$ for all (a, b). The average discrepancies of 100 sets of 16 samples for the various strategies are shown in Table 5.5. The table shows that the discrepancy of the separate samples is lowest, and the other sampling strategies are ranked in an order reasonably consistent with empirical behavior.

Zeremba points out that instead of taking the lowest upper bound of $|n/N - ab|$, we could take its root mean square value. The root mean square discrepancy, S, is shown in Table 5.6. Under this metric, the half-jittered sampling strategy slightly outperforms separate sampling. Again, the ordering is reasonably consistent with the observed behavior of the strategies.

Stroud has a slightly different definition of discrepancy: the discrepancy, T, is the lowest upper bound of $|n/N - (a - c)(b - d)|$, where c and d are the lower corner of the square, and n is the number of points within the square[109]. In other words, all squares are used, rather than

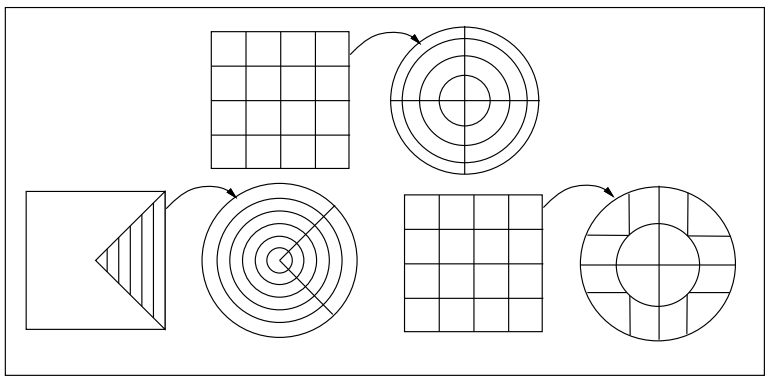

Figure 5.34: Two different ways to partition a circle

just squares with one corner at the origin. This makes the discrepancy of a 90° rotation of a set of points invariant. Stroud's discrepancy for 100 sets of samples is shown in Table 5.7. Applying this definition of discrepancy leads to an evaluation of sampling strategies that accords closely with the observable degree of error.

These tests indicate that discrepancy may be a useful tool in evaluating sampling strategies. One shortcoming of Zeremba's definitions is that it assumes a square domain. As shown in the top of Figure 5.34, a straight transformation from a evenly partitioned square to polar coordinates on a circle can stretch the strata. This implies that a good discrepancy in the canonical probability space does not guarantee good equidistribution in the actual domain (such as the lens area). A special purpose transformation is shown in the bottom of Figure 5.34. This keeps the strata from distorting too much. Unfortunately, the definition of discrepancy does not extend to non-square domains, so the only justification for preferring the bottom stratification of the circle is visual inspection and intuition.

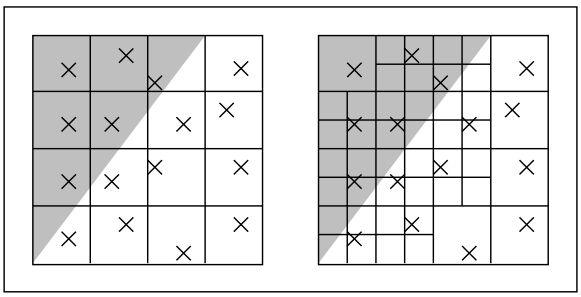

Figure 5.35: Adaptive subdivision applied when a sample has a different value than any of its 8-connected neighbors.

5.6 Strategies for Adaptive Sampling

One problem with taking a constant number of samples in each pixel is that different pixels can have vastly different variance. One technique is to take an initial sampling of the pixel with N samples, and apply more sampling if the initial samples vary[32, 30]. The tricky part of this technique is that if N is too small, a pixel with variance could have N samples that are the same (the classic feature detection problem[121]).

Kajiya tried to improve on adaptive sampling using a stratification of the samples[61]. This idea, which Kajiya called adaptive hierarchical sampling, is illustrated in Figure 5.35, where new samples are placed in strata adjacent to strata with different sample values. This is sort of a jittered Warnock subdivision. This basic idea has also been explored by Painter and Sloan[81].

Adaptive hierarchical sampling has not been successfully applied to distributed ray tracing. Kajiya[61] writes:

> *So far our experiments in finding adaptive criteria have not been terribly successful.*
>
> *We have not used adaption in computing the final images.*

I believe that the lack of success stems from the uncorrelated jitter used in distributed ray tracing; samples that are adjacent in the pixel probability space may not be adjacent in the

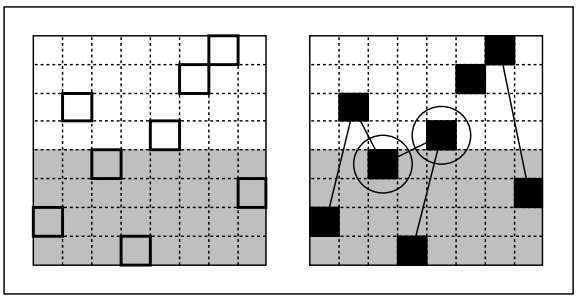

Figure 5.36: Adaptive subdivision applied to separately sampled space has many horizontal neighbors subdividing.

reflection or other probability space. As a simple example, consider the two dimensional separately sampled space shown on the left of Figure 5.36. If we want to sample hierarchically, we should refine in the two strata shown circled on the left of the figure. However, if we simply look at all the horizontal neighbor strata that are different, we will also subdivide in all the cells shown in black.

To investigate the feasibility of selectively subdividing only in the dimensions where there is actual variance, correlated jittering (where all probability spaces have the same connectivity) can be used with strict hierarchical subdivision. Figure 5.37 shows a test figure with 16 samples per pixel. Figures 5.38 and 5.39 show a detail of the figure before and after subdivision. The subdivision generated an average of about 5 extra rays per pixel over the whole image.

The subdivision technique was also applied to a path tracing application. Figure 5.40 shows indirect lighting coming off a specular block with 441 rays per pixel. Figure 5.41 shows the same picture with an initial set of 400 rays per pixel, and an average of 22 extra rays per pixel. The large number of initial samples is needed because at least one of the initial rays must hit the light source before extra sampling will take place.

These examples indicate that adaptive hierarchical sampling should be useful if the sampling spaces are subdivided in a reasonable way.

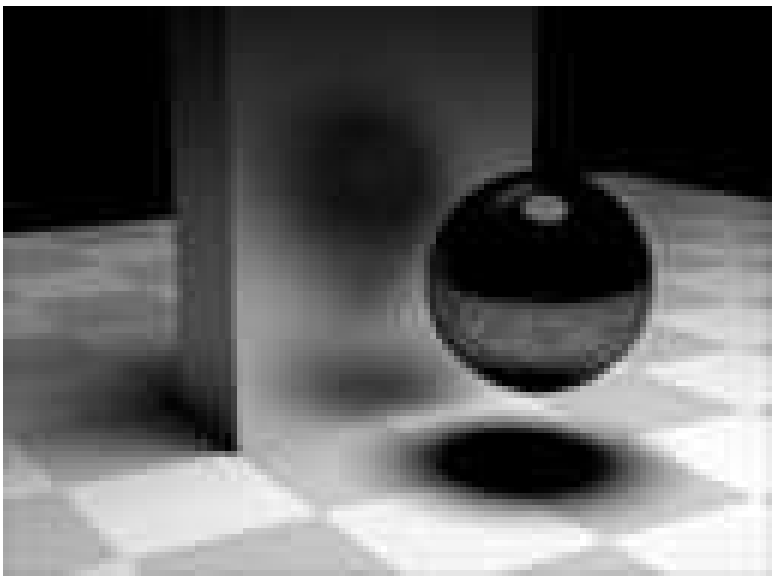

Figure 5.37: Test figure with glossy reflection

Figure 5.38: Detail of test figure with 16 samples per pixel

Figure 5.39: Detail of of test figure after two levels of subdivision

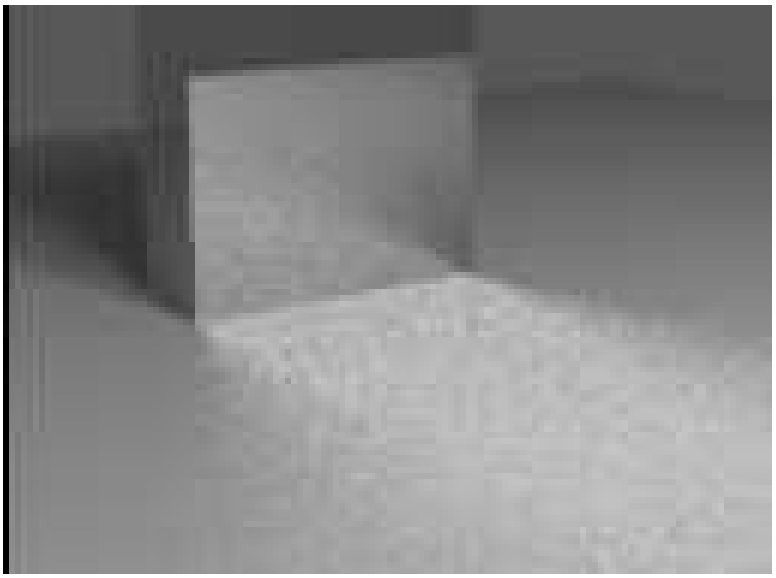

Figure 5.40: Test figure of light bouncing off mirrored block onto the ground

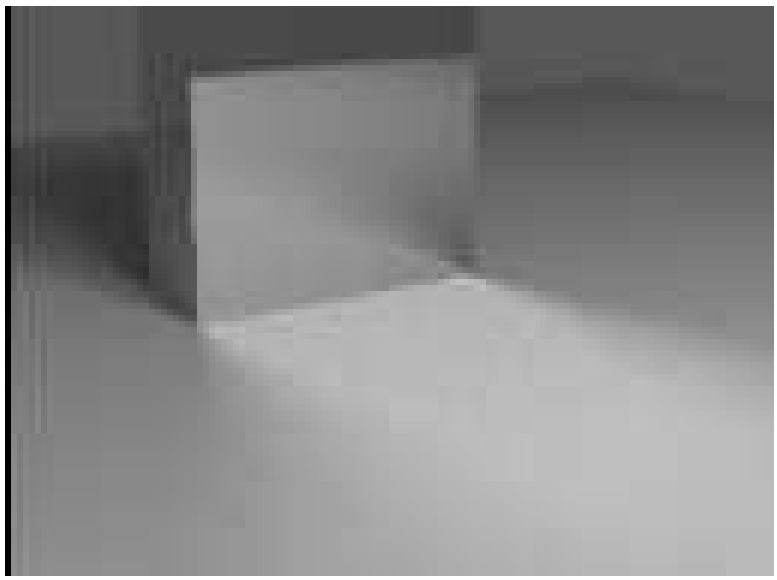

Figure 5.41: Test figure after one level of subdivision

5.7 Summary

Image-based methods calculate the colors for each pixel independently. A finite-aperture camera model fits nicely into this scheme. The color of each pixel is usually found by taking a weighted average of colors around the pixel center. The weighting function has certain features that restrict allowable functions. A new symmetric separable function was shown to satisfy these restrictions, and was also shown to have better filtering characteristics than the box filter.

Utah models assume only direct lighting plus an ambient term. Whitted-style ray tracing allows, in addition to direct lighting, ideal specular reflection and transmission, and gives a convenient way to test for shadowing.

Stochastic methods view the rendering problem as a Monte Carlo solution to a multidimensional integral. Cook et al.'s distributed ray tracing adds soft shadows, lens effects, and glossy reflection to Whitted Style ray tracing. The concept of uncorrelated jittering is central to distributed ray tracing, and can be understood as a general multidimensional sampling technique.

Kajiya generalized distributed ray tracing to allow for general light transport. Ward et al. modified Kajiya's method by saving information that can be used later on. A new stochastic method of restricting the number of shadow rays was discussed that makes creating images of scenes containing multiple light sources more practical.

If a predetermined number of samples is sent through each pixel, then the spatial distribution of samples should be chosen that minimizes error without introducing coherence to the distribution of error in the image. Separate sampling is shown to be superior to jittering or to Poisson Disk sampling for some cases. The notion of discrepancy, an error prediction metric used in numerical integration, is presented as a method of predicting the success of a sampling strategy. Using discrepancy seems to have some advantages over traditional signal processing approaches.

Kajiya's hierarchical sampling cannot be correctly used in stochastic rendering because of the uncorrelated jitter used to reduce noise. A modification to this sampling method was shown to allow an adaptive hierarchical sampling scheme for the full multidimensional integral.