Variance Reduction II

Lecture #9: Tuesday, 28 October 1997
Lecturer: Eric Veach
Scribe: David Hoffman
Reviewer: James Davis

1 Transformation of the Unit Square into the Unit Disk

Often it is useful to be able to sample a circular domain with uniform density. For instance, a cosine-weighted sampling of the hemisphere can be obtained by first generating samples uniformly within a unit disc, and then projecting them vertically up onto the hemisphere (as we saw in the last lecture). This is useful for sampling reflections from diffuse surfaces.

To achieve uniform sampling on the unit disk, one might use a simple parameterization such as

$$r = u$$
$$\theta = 2\pi v$$

and sample (u, v) on a regular grid in the unit square. But the strata produced by this method do not have uniform (or even similar) areas, as figure 1 shows.

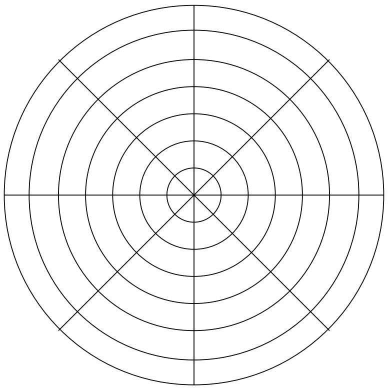

Figure 1: Image of the unit square under the mapping $(r, \theta) = (u, 2\pi v)$

One way to correct this problem is to re-parameterize as

$$r = \sqrt{u}$$
$$\theta = 2\pi v$$

which produces strata of equal area (figure 2). But these strata are less and less compact away from the center of the disk.

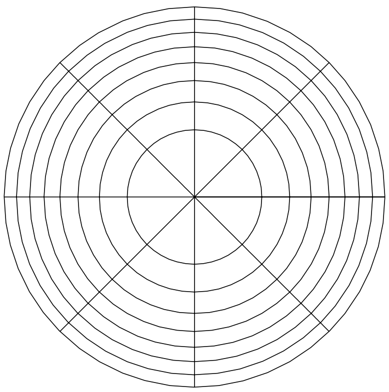

Figure 2: Image of the unit square under the mapping $(r, \theta) = (\sqrt{u}, 2\pi v)$

There is a better mapping that transforms compact areas of the square of side 2 ($[-1, 1] \times [-1, 1]$) into compact areas of the unit disk and also preserves their area (or rather, their fractional area, since the area of the unit disk is $\pi/4$ times that of the side-2 square). The triangular region below $v = u$ and above $v = 0$ maps to the region $\theta \in [0, \pi/4]$ using the map

$$r = u$$
$$\theta = \tfrac{\pi}{4}\tfrac{v}{u}$$

as shown in figure 3.

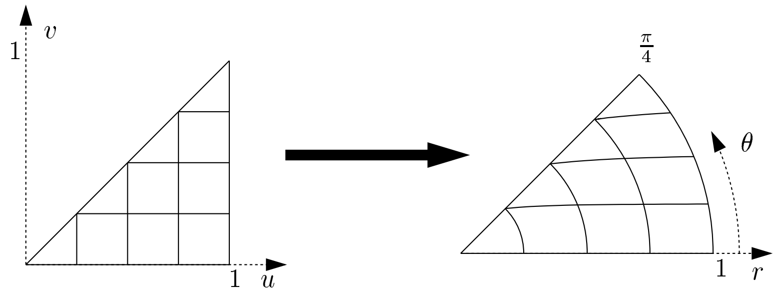

Figure 3: Transformation of part of the first quadrant by the mapping $(r, \theta) = (u, \frac{\pi}{4}\frac{v}{u})$

The mapping can then be reflected and rotated to cover the remaining 7 regions of the disk as in figure 4. The fact that this transformation preserves areas can be verified by computing its Jacobian determinant, as we did in the previous lecture.

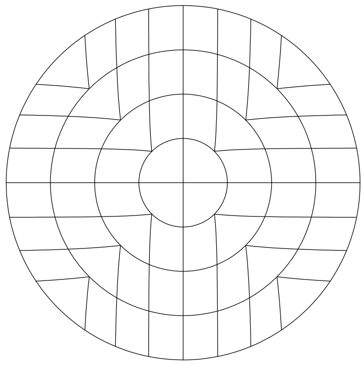

Figure 4: Completion of the $(r, \theta) = (u, \frac{\pi}{4}\frac{v}{u})$ mapping by rotation and reflection

2 Why (and when) Stratified Sampling reduces variance

Since mappings like the above are readily available for specific domains, most discussion of stratified sampling assumes that the domain is an s-dimensional unit cube. This domain is split into *strata* labeled $S_1 \ldots S_k$ which are disjoint and completely cover the domain:

$$\begin{aligned} S_i \bigcap S_j &= \emptyset \qquad \text{for all } i \neq j, \\ \bigcup_{i=1}^{k} S_i &= [0,1]^s \,. \end{aligned}$$

In general there are no other shape restrictions on the strata, but in practice they are most often rectangular regions. We denote the volume of a stratum as v_i, and clearly $\sum_i v_i = 1$. There will be a certain number of samples within each stratum, and we denote this number as n_i. Also, the samples are uniformly distributed, so

$$p_i(x) = \begin{cases} 1/v_i & \text{if } x \in S_i \\ 0 & \text{otherwise} \end{cases}$$

By taking n_i samples in stratum S_i, we get an estimated average value of

$$\hat{I}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} f(X_{i,j})$$

for the integrand in S_i, where $X_{i,j}$ is the jth sample in that stratum (which has the density function p_i defined above). The overall estimate will be the sum of the averages, each weighted by the volume of its stratum:

$$\hat{I} = \sum_{i=1}^{k} v_i \hat{I}_i$$

The actual average value of the integrand within each stratum is

$$\mu_i = E[f(X_{i,j})] = \frac{1}{v_i}\int_{S_i} f(x)dx$$

and this is the expected value of the estimator for the stratum, so $E[\hat{I}_i] = \mu_i$. The variance of the function within each stratum is

$$\sigma_i^2 = V[f(X_{i,j})] = \frac{1}{v_i}\int_{S_i} (f(x) - \mu_i)^2 dx \ ,$$

so the variance of the estimator for a stratum with n_i averaged samples is

$$V[\hat{I}_i] = \sigma_i^2 / n_i \ ,$$

and the variance of the overall stratified estimator is

$$V[\hat{I}] = V\left[\sum_{i=1}^{k} v_i \hat{I}_i\right] = \sum_{i=1}^{k} V[v_i \hat{I}_i] = \sum_{i=1}^{k} v_i^2 V[\hat{I}_i] = \sum_{i=1}^{k} \frac{v_i^2 \sigma_i^2}{n_i}$$

If we assume that the number of samples in each stratum is proportional to its volume, that is, $n_i = v_i N$ where N in the total number of samples, then this simplifies to

$$V[\hat{I}_s] = \frac{1}{N}\sum_{i=1}^{k} v_i \sigma_i^2 \tag{1}$$

(We have added the subscript s to indicate that this estimator is stratified.)

If the same number of samples is taken without stratification (i.e. where each sample is chosen independently and uniformly over $[0,1]^s$), what would the variance be? We have already defined the mean and variance within each stratum. The variance across the entire domain can be expressed as the mean of the individual variances plus the variance of the means, or[1]

$$V[f(X)] = \sum_{i=1}^{k} v_i \sigma_i^2 + \sum_{i=1}^{k} v_i (\mu_i - I)^2 \tag{2}$$

where X is uniformly distributed on $[0,1]^s$, and where I is the expected value of the means, which is the sought-after integral

$$I = \int_{[0,1]^s} f(x)\, dx \ .$$

[1]This formula can be proven by expanding the right-hand side directly, or observing that an unstratified sample X is equivalent to first choosing a random stratum I according to the discrete probabilities v_i, and then choosing choosing X uniformly within S_I. From this point of view, X is chosen conditionally on I, and we can obtain the given formula using the standard identity $V[F] = E_I V_{X_I} F + V_I E_{X_I} F$ where $F = f(X_I)$.

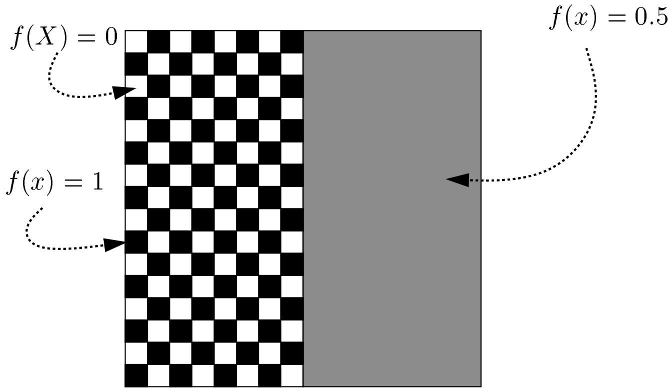

Figure 5: In both strata of this 2-dimensional function, the mean is 0.5

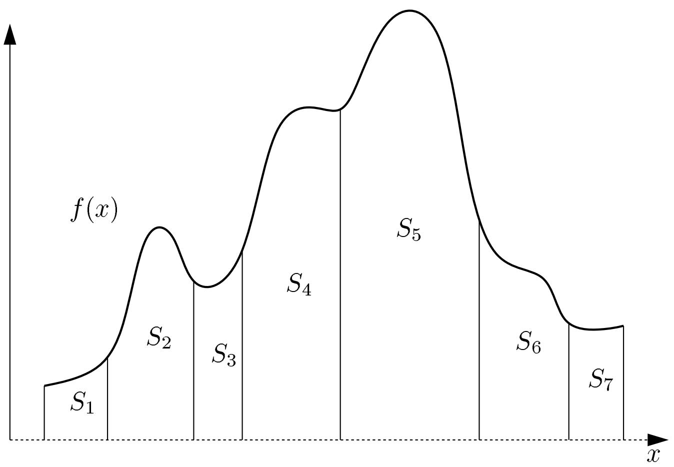

Figure 6: Well chosen strata can have widely varying means

Then the variance of the non-stratified estimator that averages N samples is

$$V[\hat{I}_{ns}] = \frac{1}{N}\left[\sum_{i=1}^{k} v_i \sigma_i^2 + \sum_{i=1}^{k} v_i (\mu_i - I)^2\right] \tag{3}$$

From equations 1 and 3 we can see that the non-stratified estimator's variance is larger than that of the stratified estimator by $\frac{1}{N}\sum_{i=1}^{k} v_i(\mu_i - I)^2$. Thus we only get a reduction in variance when the means μ_i of the strata differ from the overall mean I. For example, in figure 5, the mean in both strata is 0.5, so the variance of the mean is zero and there is no advantage in stratification (at least when the number of samples in each stratum is proportional to its volume). But in figure 6, the means vary widely between strata, so the variance reduction will be large if stratified sampling is used.

The need for widely varying means explains why compact strata are more desirable when little is known of the underlying function to be integrated. If the strata have

c	b	d	a
a	d	b	c
b	c	a	d
d	a	c	b

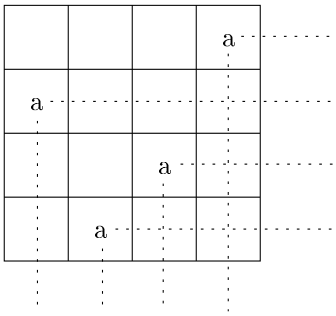

Figure 7: A single symbol from a 4×4 Latin square stratifies both the x and y coordinates

wide support in any direction, there is likely to be more variation within the strata and therefore less variation of the means between strata. By using compact strata, the variance of the means is likely to be increased, thereby increasing the variance reduction achieved by stratified sampling.

3 Latin Hypercube Sampling

In graphics, a typical application of stratified sampling is the selection of ray directions for ray tracing. For example, when there are multiple light sources or area light sources, the strata can be chosen to direct rays toward the light. Similarly, anti-aliasing can be performed by subdividing each pixel into several strata. Depth of field effects are achieved by sampling over the area of the aperture. Surface reflections are handled by sampling a hemisphere of directions according to a bidirectional reflectance distribution function, and motion blur by sampling over time. A ray tracer that includes all of these effects will be sampling in at least 9 dimensions (2 for pixels, 2 for the aperture, 2 for the reflected direction, 2 for the direction to the light source, and 1 for time). If each dimension is split into just 2 strata, there are 512 samples required per pixel! As the dimensionality of the space increases, the rate of convergence of the estimator decreases with respect to any particular dimension, just as in numerical integration. One technique to circumvent this problem of dimensionality is *Latin Hypercube Sampling*.

A "Latin square" is an $n \times n$ array of symbols – drawn from an alphabet of size n – in which each symbol appears exactly once in each row and once in each column. Figure 7 shows a 4×4 example. If we consider this a square 2-dimensional domain divided into 16 square strata, we could take a sample in each square. But we could also take samples just in those squares labeled "**a**". With only 4 samples, we would still have a sample in each row and a sample in each column.

This sampling strategy is easy to extend to s dimensions without the need to construct a full Latin hypercube. In each dimension the range $[0, 1]$ is divided into N intervals, resulting in a domain with N^s subcubes. By choosing s permutations Π_j of the integer range $\{1,\ldots,N\}$ and using the ith number in Π_j as the jth coordinate of sample X_i (where $1 \le i \le N$), we guarantee that in each dimension we have sampled all N intervals of $[0,1]$ – in other words, the samples are stratified on each of the dimensions individually. We write the jth coordinate of the ith sample as

$$X_i^j = \frac{\Pi_j(i) - U_{ij}}{N}$$

where U_{ij} is a uniform random number in $[0,1]$, making the probability distribution uniform in each one-dimensional interval and therefore in each s-dimensional subcube. So in the right side of figure 7, we have $\Pi_1 = \{1,2,3,4\}$ and $\Pi_2 = \{3,1,2,4\}$.

To analyze the convergence of this method, let us first suppose that $f(x)$ is an *additive function*, i.e. a function of the form

$$f(x) = \sum_{j=1}^{s} f_j(x^j) \tag{4}$$

where x^j denotes the jth coordinate of x. In other words, an additive function is a sum of functions f_j that each depend on only one coordinate of the vector x. To get an idea of how Latin hypercube sample converges for this type of function, let

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} f(X_i)$$

and suppose that we fix every sample at the center of its cell by setting $U_{ij} = 1/2$. This yields

$$\hat{I} = \sum_{j=1}^{s} \frac{1}{N} \sum_{i=1}^{N} f_j\left(\frac{i-\frac{1}{2}}{N}\right) , \tag{5}$$

noting that under summation over i we can replace $\Pi_j(i)$ with i. Equation (5) is simply the sum of s midpoint integration rules, one for each f_j, which has an error of $O(N^{-2})$ provided that the f_j are smooth. It is also possible to show that if U_{ij} is a uniform random number in $[0,1]$ (rather than being fixed at $1/2$), the error is $O(N^{-3/2})$ with high probability (for smooth functions). In either case, the convergence is better than that of standard Monte Carlo,which is $O(N^{-1/2})$.

To generalize this result, suppose that f is decomposed into a sum

$$f(x) = f_{add}(x) + f_{res}(x)$$

where $f_{add}(x)$ is an additive function, and $f_{res}(x)$ is the "residual" which is orthogonal to all additive functions, i.e.

$$\int_{[0,1]^s} f_{res}(x)\, g_{add}(x)\, dx = 0$$

for any additive function $g_{add}(x)$. This decomposition is unique, and we will give an explicit formula for $f_{add}(x)$ in terms of $f(x)$ below.

Given this decomposition, the variance of Latin hypercube sampling is

$$V[\hat{I}] = \frac{1}{N} \int f_{res}^2(x) dx + o\left(\frac{1}{N}\right)$$

for any square-integrable function $f(x)$. The second term is a function that decreases more rapidly that c/N for any constant c. Thus, Latin hypercube sampling increases the rate of convergence on the components of the function that depend on just one coordinate at a time. The closer that f is to being a purely additive function (i.e. the smaller the residual part $f_{res}(x)$ is), the better Latin hypercube sampling will do.

3.1 Analysis of variance decompositions

We can determine an explicit formula for decomposing f into its components f_{add} and f_{res}, using an *analysis of variance decomposition* (usually abbreviated *anova*). Let S be the set $\{1, \ldots, s\}$ of all coordinate indices, and consider all the possible subsets $U \subseteq S$ of these coordinates (there are 2^s possible subsets in all). In the anova framework, an arbitrary function f is expressed as a sum

$$f(x) = \sum_{U \subseteq S} f_U(x^U)$$

where each function f_U depends only on the coordinates x^U, and all the f_U are orthogonal. The function when $U = \emptyset$ does not depend on any variables, and is called the *grand mean*:

$$\mu = f_\emptyset = \int_{[0,1]^s} f(x) dx$$

The other $2^s - 1$ subsets of U are called *sources of variation*. For example, the components of f that depend on just one variable are called the *main effects* and are defined as

$$f_j(x^j) = \int (f(x) - \mu) \prod_{i \neq j} dx^i$$

Notice that all of these functions are orthogonal to the constant function $f_\emptyset = \mu$. The best additive approximation to f can then be written as

$$f_{add}(x) = \mu + \sum_{j=1}^{s} f_j(x^j) \ ,$$

where the residual $f_{res} = f - f_{add}$ is orthogonal to all additive functions.

Similarly, one can define the functions

$$f_{j,k}(x^{j,k}) = \int \Big(f(x) - \mu - f_j(x^j) - f_k(x^k)\Big) \prod_{i \neq j,k} dx^i$$

which represent the parts of f that depend on two particular variables together (the *two-factor interactions*). These functions are orthogonal to $f_\emptyset$ and to all the f_j.

In general, f_U is defined by

$$f_U(x^U) \;=\; \int \left(f(x) - \sum_{V \subset U} f_V(x^V) \right) dx^{S-U} \tag{6}$$

where the sum is over all proper subsets of U $(V \neq U)$. This yields a set of function f_U that are all orthogonal, i.e.

$$\int f_U(x^U)\, f_V(x^V)\, dx \;=\; 0$$

whenever $U \neq V$. This implies that

$$\int f^2(x)\, dx \;=\; \sum_{U \subset S} \int f_U^2(x^U)\, dx$$

so that the variance of f can be written as

$$\int (f(x) - \mu)^2\, dx \;=\; \sum_{|U|>0} \int f_U^2(x^U)\, dx \;\; .$$

With Latin hypercube sampling, the components of the variance where $|U| = 1$ converge at a rate faster than $1/N$, leaving only the residual component

$$\int f_{res}^2(x)\, dx \;=\; \sum_{|U|>1} \int f_U^2(x^U)\, dx \;\; .$$

4 Orthogonal Arrays

This above analysis suggests that one might try to select samples that are stratified in every *pair* of variables, in addition to every single variable. This is, in fact, possible using a version of Latin hypercube sampling based on *orthogonal arrays*.

An orthogonal array is an $N \times s$ matrix whose entries – from an alphabet of b symbols – are such that every $N \times t$ submatrix contains exactly the same number of copies of all b^t possible rows.[2] If we let λ be the number of times that each row appears in this submatrix, then clearly the total number of rows is $N = \lambda b^t$. The parameter t is called the *strength* of the orthogonal array. The following table shows an example of an orthogonal array A_{ij} whose parameters are $(N, s, b, t) = (9, 4, 3, 2)$.

[2] The submatrix is not necessarily contiguous; it can consist of any subset of the columns.

0	0	0	0
0	1	1	2
0	2	2	1
1	0	1	1
1	1	2	0
1	2	0	2
2	0	2	2
2	1	0	1
2	2	1	0

This array was constructed by the rule

$$\begin{aligned} A_{i1} &= x \\ A_{i2} &= y \\ A_{i(j+2)} &= x + jy \qquad \text{for } j = 1 \dots b-1, \end{aligned}$$

where $i = bx+y+1$, and $0 \le x, y \le b-1$. This technique works in general for constructing an orthogonal array with $N = b^2$, $s = b+1$, $t = 2$, and b symbols, for any prime number b. There are a variety of other techniques known for constructing such arrays.

So, how do we use such an array for sampling? Suppose that we divide the domain into b^s cells, by splitting each axis into b intervals of equal size. Also, assume that the symbols in the orthogonal array are $\{0, 1, \dots, b-1\}$. Then each $1 \times s$ row of the array can be considered an index that specifies one of the b^s possible cells. The overall strategy is to generate one sample in each of the N cells specified by the rows of the orthogonal array. Specifically, the j-th coordinate of sample X_i is given by

$$X_i^j = \frac{A_{ij} + U_{ij}}{b}$$

where U_{ij} is a uniform random number in $[0, 1]$.

Next, consider the projection of these samples onto any set of t coordinates. (There are $\binom{s}{t}$ possible projections in all.) Then from the orthogonal array property, these samples will be uniformly distributed among the b^t cells obtained by stratifying these axes into b equal-sized intervals. To see this, observe that the coordinates of the projected samples are specified by the rows of a particular $N \times t$ submatrix. That is, each $1 \times t$ row specifies one of the b^t cells in the t-dimensional projection of the domain. By the definition of orthogonal arrays, every possible $1 \times t$ row appears λ times in this submatrix, so each of the b^t cells occurs exactly λ times.

Therefore, orthogonal array sampling produces samples that are stratified with respect to *every* possible subset of t coordinates. We should note here that given an orthogonal array, one can create another with the same parameters simply by permuting the columns, or by permuting the alphabet within each column. This is significant because we don't want to always sample the same set of N cells out of the $\binom{b^s}{N}$ sets that are possible.

The variance of orthogonal array sampling can be written as

$$V[\hat{I}] = \frac{1}{N} \sum_{|U|>t} \int f_U^2(x^U)\, dx \,+\, o\left(\frac{1}{N}\right) .$$

Thus, it improves the rate of convergence with respect to all components of the integrand that depend on t coordinates or less. The case $t = 2$ is particularly interesting for graphics. For example, if we apply this technique to distribution ray tracing, it ensures that all the two dimensional projections are well stratified (e.g. over the pixel, lens aperture, light source, etc.) This also includes combinations of dimensions such as the x-coordinate of the current pixel and the x-coordinate of the lens aperture.

4.1 Orthogonal array-based Latin hypercube sampling

Notice that because we get such good stratification with respect to t-dimensional projections, we also get some stratification with respect to w-dimensional projections for $w < t$. For simplicity, assume $\lambda = 1$. Then in a w-dimensional projection, each cell receives b^{t-w} samples. For example, this says that in each one-dimensional interval of width $1/b$ along any axis j, there are exactly b^{t-1} samples. Notice that this is not as good as Latin hypercube sampling, which places one sample in each interval of width $1/b^t$.

There is a simple modification to orthogonal array sampling that yields the same one-dimensional stratification properties as Latin hypercube sampling. (The result, logically enough, is called *orthogonal array-based Latin hypercube sampling.*) Again assume that $\lambda = 1$, so that the array has b^t rows. Then observe that within each column, the values $\{0, \ldots, b-1\}$ each appear b^{t-1} times. The technique works by remapping these symbols into a single sequence $\{0, \ldots, b^t - 1\}$, by letting the b^{t-1} identical copies of each value m map to the sequence $(b^{t-1}m, \ldots, b^{t-1}(m+1)-1)$ according to some random permutation. This process is repeated for each column separately. Calling the modified array $\hat{A}$, we then change the coordinate assignment to

$$X_i^j = \frac{\hat{A}_{ij} + U_{ij}}{b^t} .$$

This will ensure that the samples are maximally stratified for each one-dimensional projection, as well as for each t-dimensional projection.