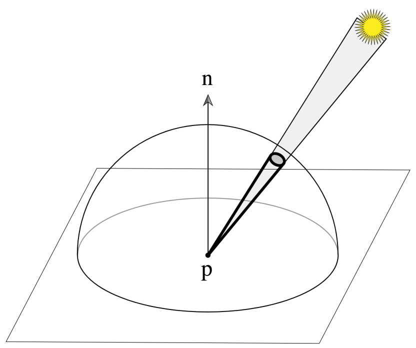
Warp and Effect

Matt Pharr
NVIDIA

Most of the rest of this course is about the general problem of choosing good sample points in the [0,1)^n domain. However, in graphics, we're often not exactly integrating over [0,1)^n. It turns out that there are some interesting implications of that little detail, and that's what we'll talk about in this section.

Theory $$\int_{[0,1)^n} f(x)\,\mathrm{d}x$$

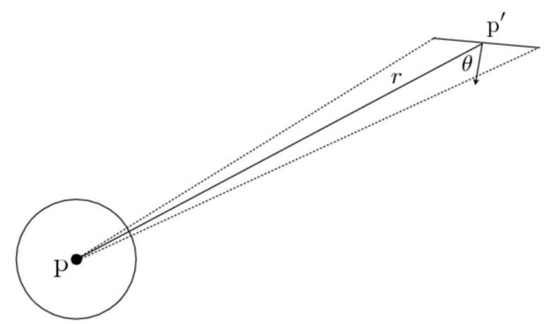
Practice $$\int_{\Omega} L_{\mathrm{i}}(\omega)\, f(\omega \to \omega_{\mathrm{o}})\, \cos\theta \,\mathrm{d}\omega$$



For example, consider computing direct lighting from an area light source: we're integrating over the hemisphere around the point p.

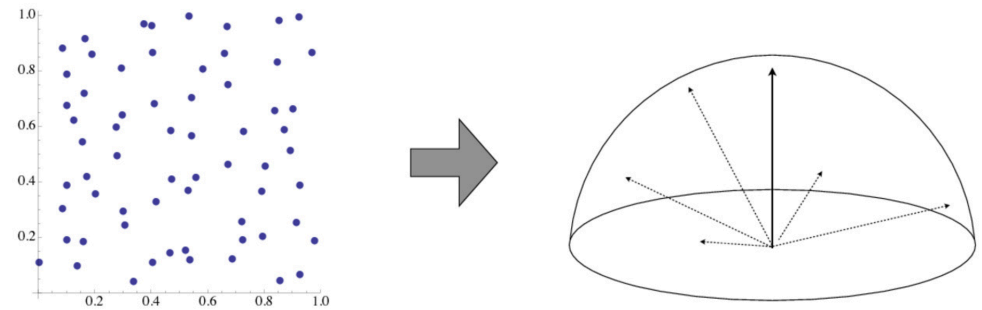
Theory $$\int_{[0,1)^n} f(x)\,\mathrm{d}x$$

Practice $$\int_A \frac{L_\mathrm{e}(\mathrm{p}')\, f(\omega_\mathrm{i} \to \omega_\mathrm{o})\ \cos\theta_\mathrm{i}\ \cos\theta}{r^2}\,\mathrm{d}\mathrm{p}'$$



Another way to formulate direct lighting is as an integral over the surface of the light source. These are just two examples of something that comes up a lot.

Solution: Warp Samples

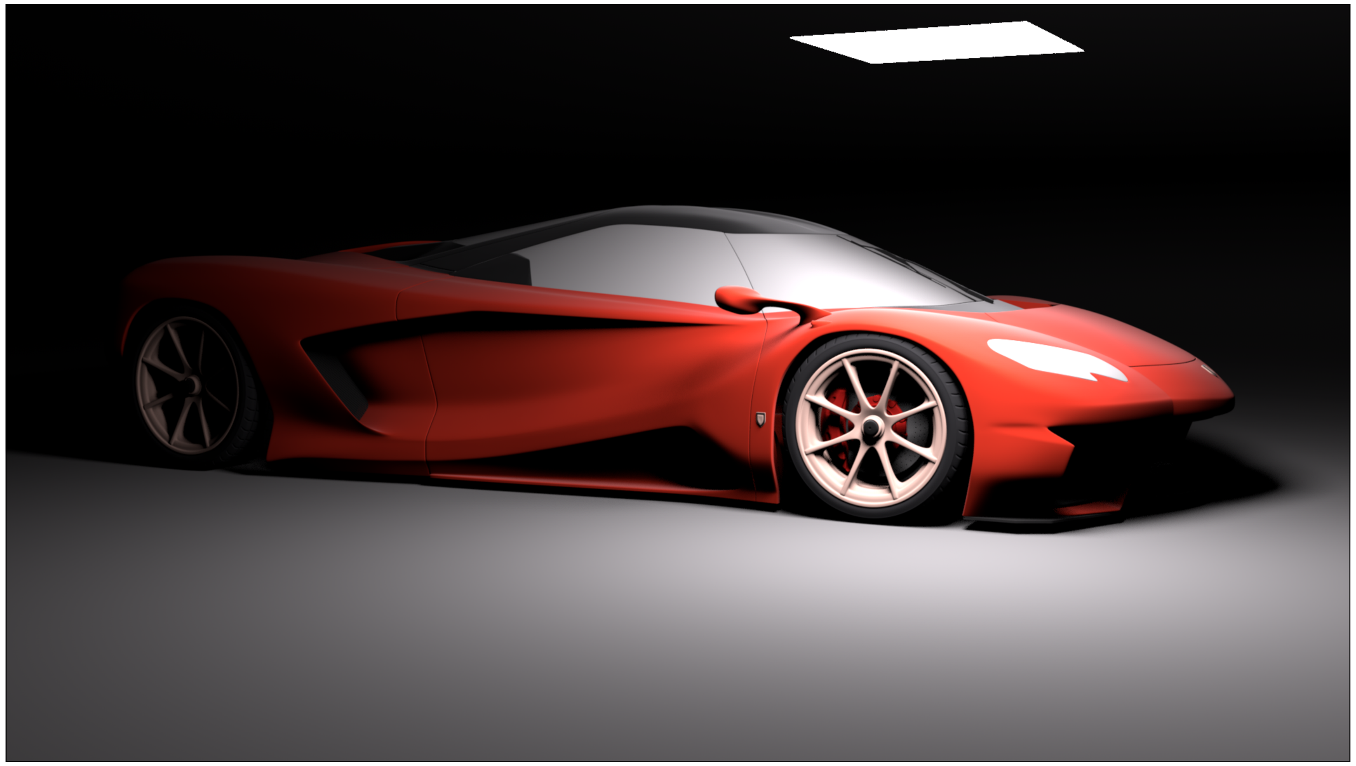


(And account for change-of-variables.)
(Or, more generally, importance sample.)

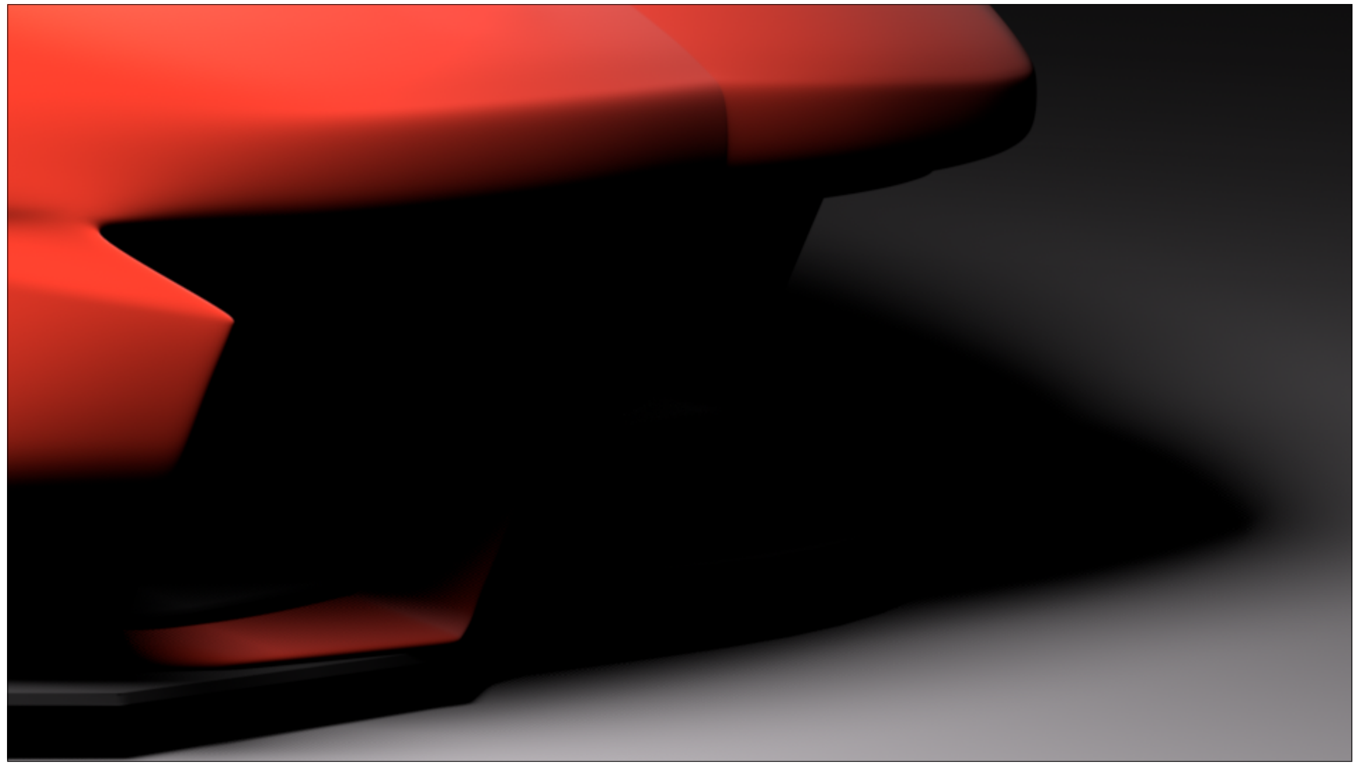
Mathematically, the solution is straightforward: warp the samples from the canonical [0,1)^n domain to the domain that we're integrating over. There are all sorts of recipes to do this.

Importance sampling is in some sense a generalization of that idea, where the samples are warped to a distorted domain to better match the function being integrated.

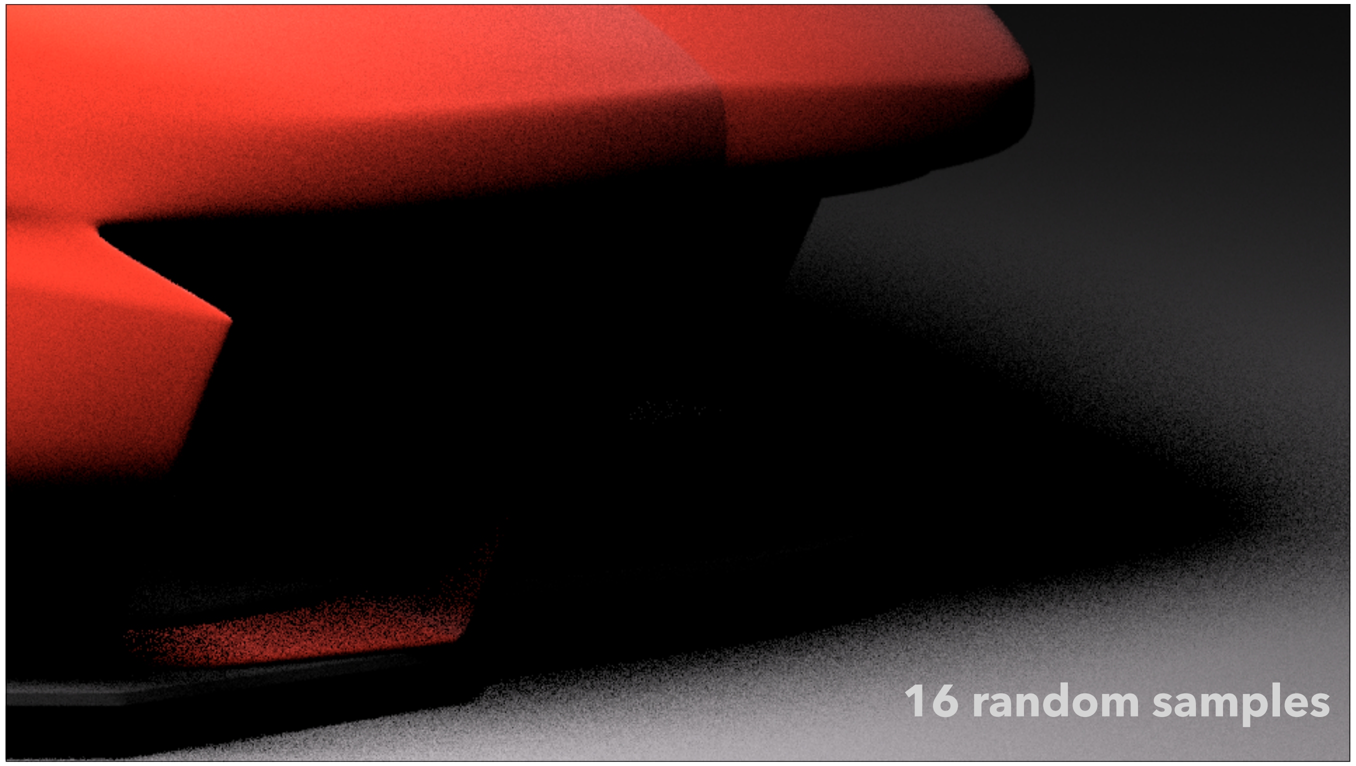
Although the math just works when you do this stuff, these warps have some surprising implications.



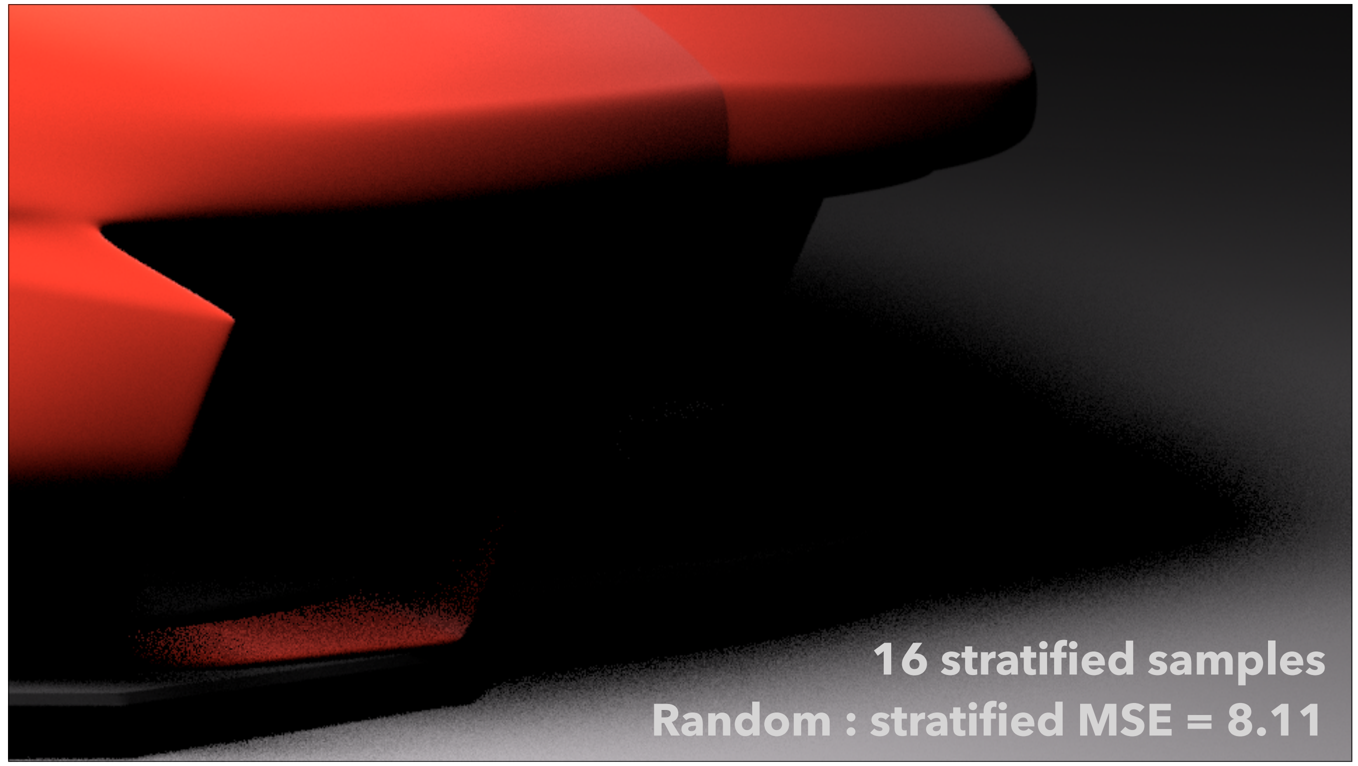
To understand some of these issues, we'll consider this case: a car model lit by a quadrilateral light source, shown here.



And here's a zoom in of the part of the image that we'll look at more closely. This is a high-quality reference rendering with a few thousand samples per pixel.



Here's what happens with random sampling. Random samples definitely shouldn't be your favorite samples, but we'll use them for a baseline (you should be booing at the idea of using them). Look at the penumbra and the red part at the bottom of the car—they're pretty noisy.



And here's what happens if we use regular old stratified samples, along the lines of what people were doing in the mid-1980s. It's a big improvement! Mean squared error is down by a factor of 8x, and so, it would take about 8 times as many random samples to reach this error. Note how much better the penumbra looks.

The great thing about this is that it's essentially no more work to use stratified samples than random samples, yet there's this huge benefit. That's generally the case for good sampling patterns—very little work to generate samples compared to the cost of evaluating the integrand, which usually involves tracing a ray.

Stratified Sampling

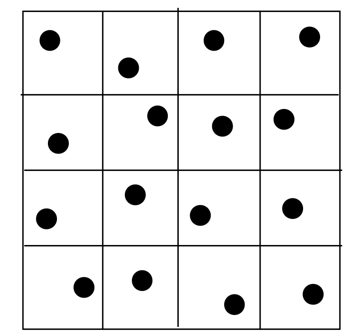
Decompose domain into non-overlapping *strata*

Estimate each stratum individually:

$$\langle F_i \rangle = f(x_i)$$

Overall estimator:

$$\langle F \rangle = \frac{1}{N} \sum_{i=1}^{N} F_i$$



So why did stratification help so much there? Let’s review how it works.

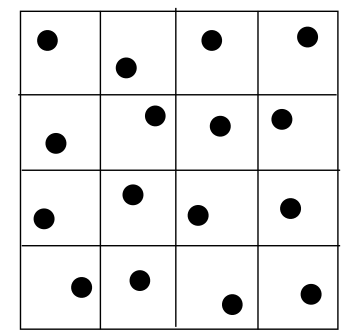
The idea with stratification is to decompose the domain into independent overlapping strata, compute independent estimates in each stratum, and then average the estimates. Here’s the estimator.

Stratified Sampling

Variance

$$V[F] = \frac{1}{N^2} \sum_{i=1}^{N} V[F_i]$$

If the variance in each region is the same,
then total variance goes as $1/N$

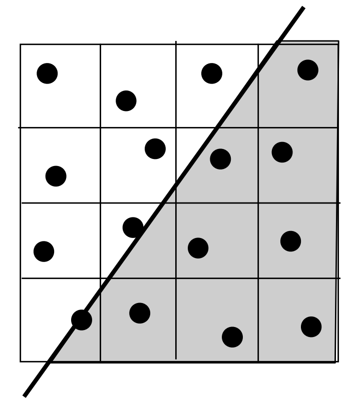


And here’s the variance of that estimator: the sum of the individual variances, divided by the square of the number of strata. That N^2 is a result of the properties of variance.

If each region has the same variance, then the overall variance goes down at a rate of 1/N. That matches regular Monte Carlo. Put another way, there’s no free lunch. If all regions have the same variance, then stratification gives no benefit.

Sampling A Polygon

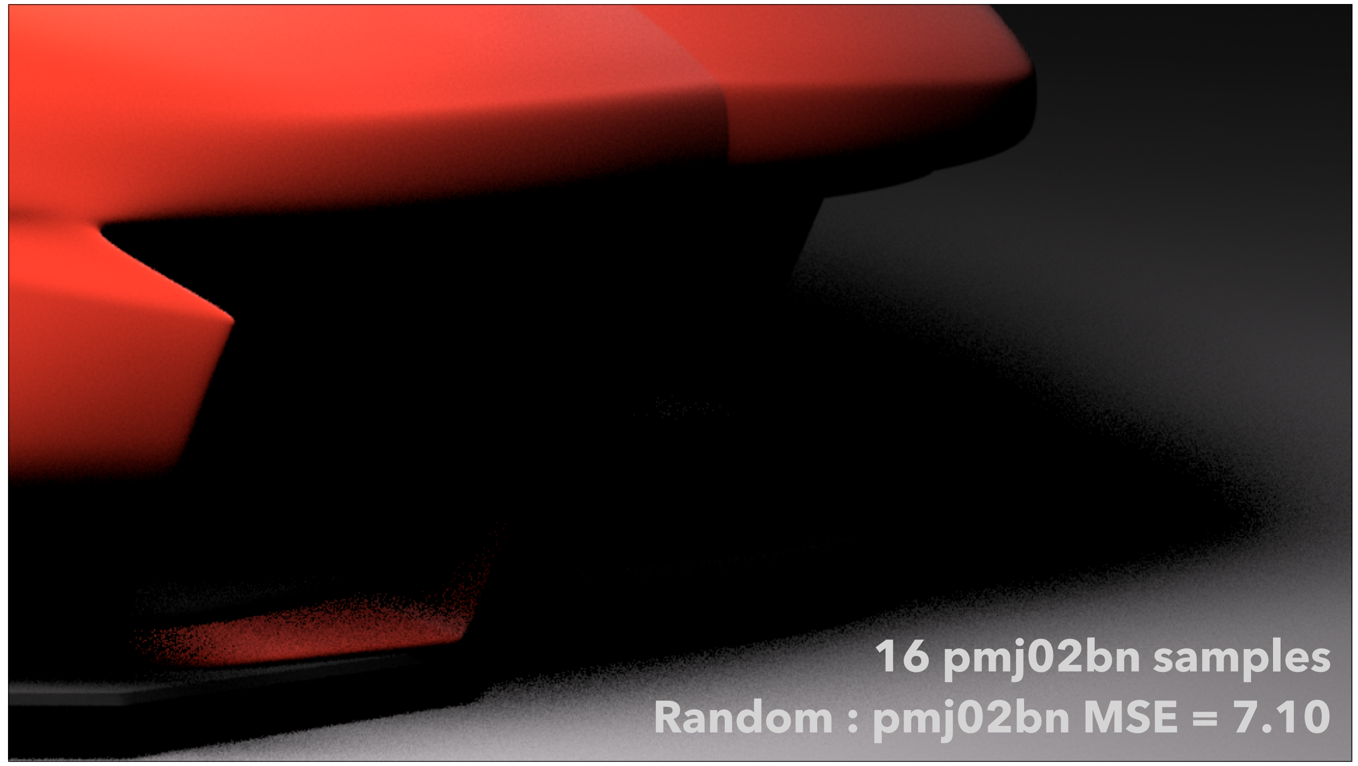
$$V[F] = \frac{1}{N^2} \sum_{i=1}^{\sqrt{N}} V[F_i] = \frac{V[F_\mathrm{E}]}{N^{1.5}}$$



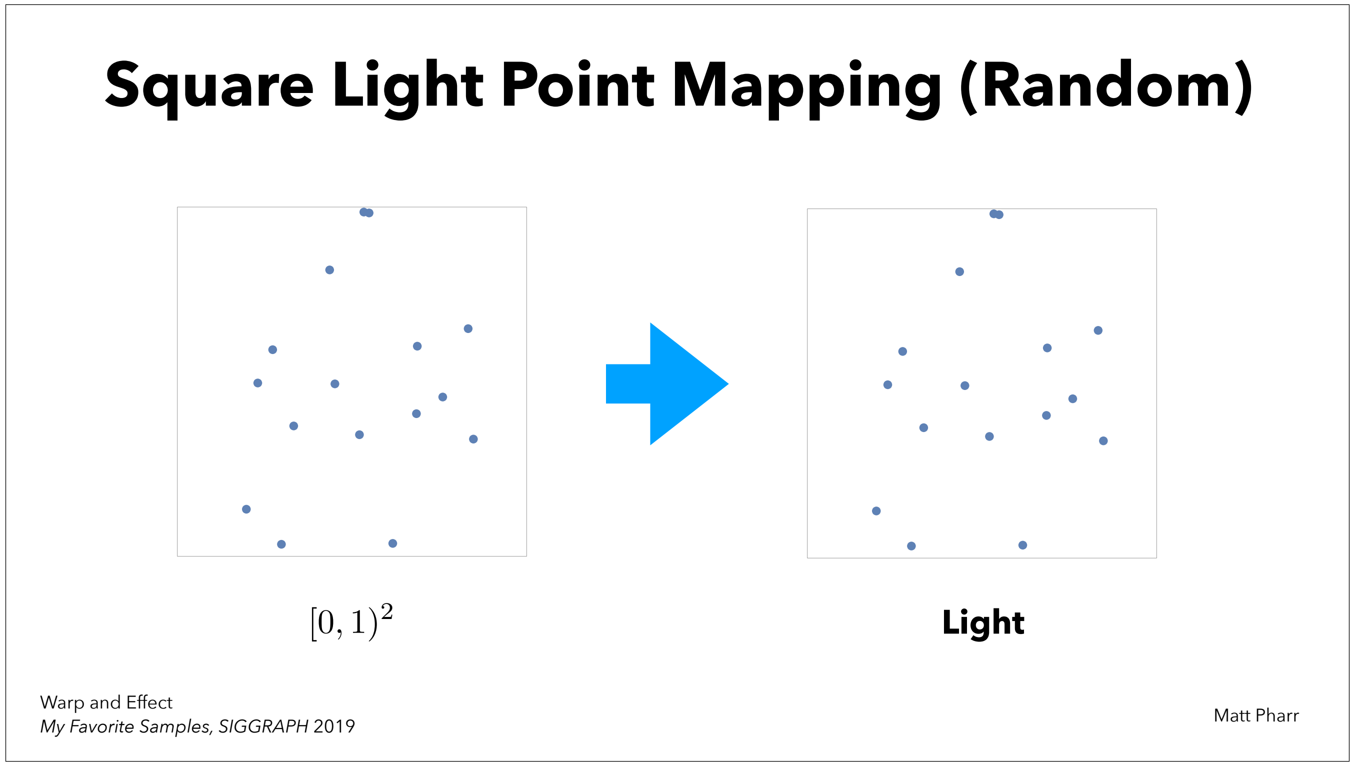
If the variance varies across the regions,
then the overall variance will be reduced.

But what if they don't all have the same variance? Consider the case of computing the area of a single polygon's overlap with a pixel's extent. Many of the strata are either entirely inside or outside of the polygon. In turn, they have constant-valued functions and in turn, zero variance.

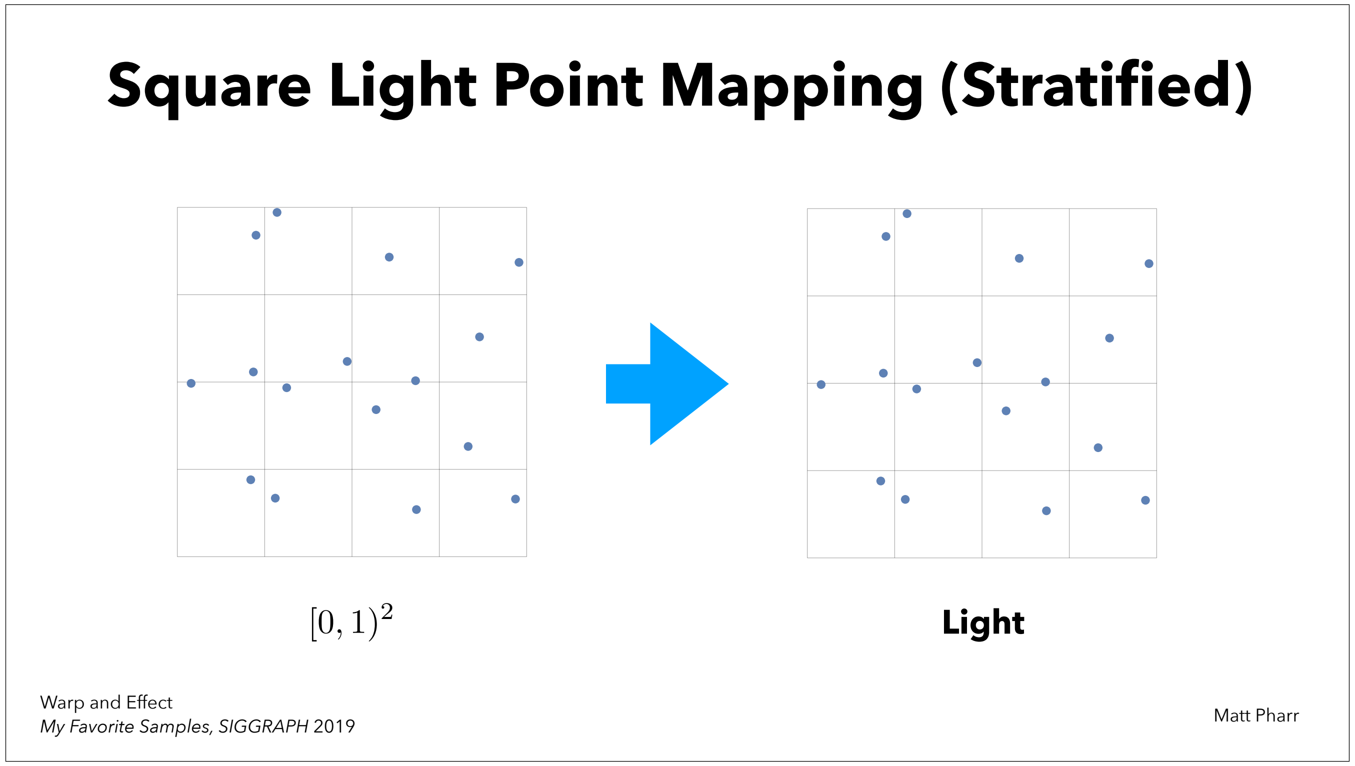
There's only non-zero variance in the strata along the polygon's edge. In turn, stratified sampling gives lower variance and a faster rate of convergence than random sampling in this case.



This course is all about much fancier sampling patterns than regular old stratified. Here's what happens with the pmj02bn points that Per and colleagues have introduced. It's a little surprising that it doesn't do quite as well as the stratified points, but them's the numbers I measured. They're still way better than random points.

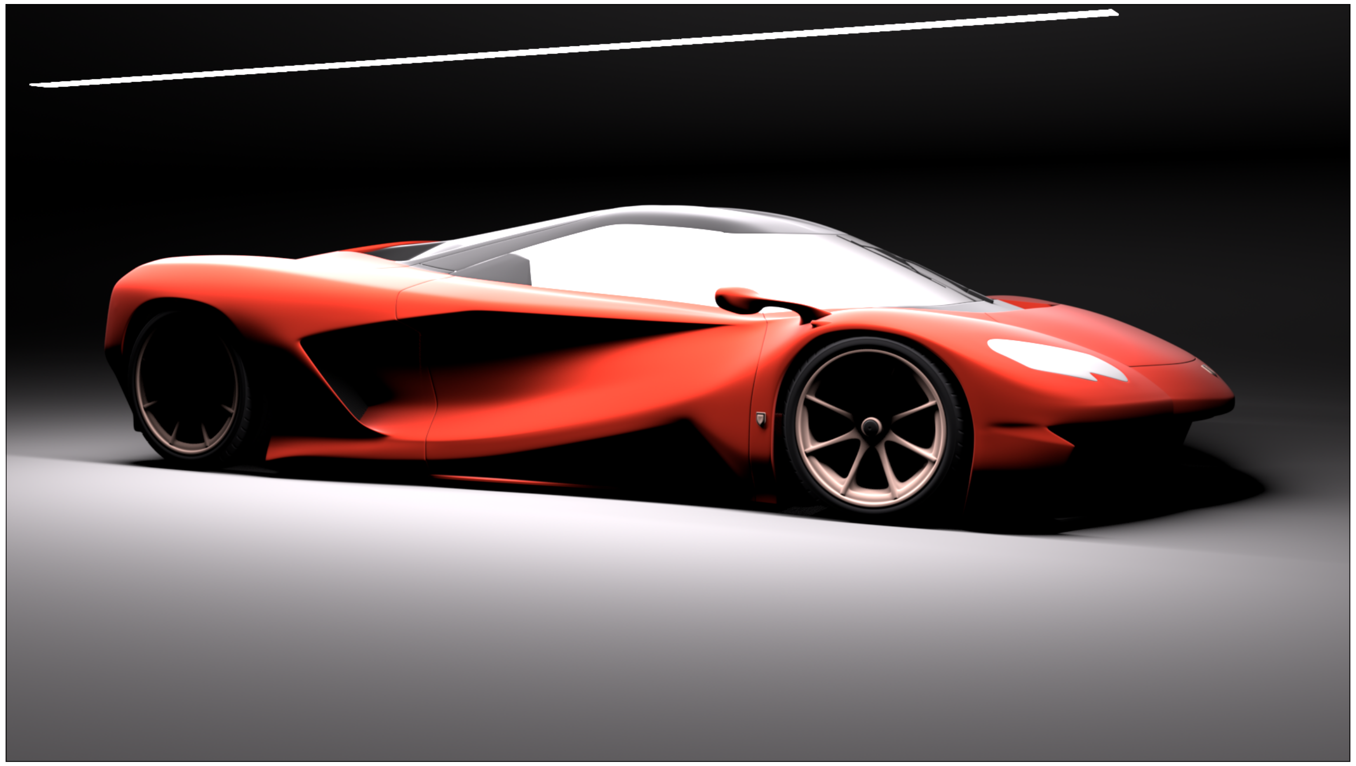


To understand what's going on here, let's consider how the sample points are mapped to the light source's surface. For a square light like we have here, the mapping is essentially the identity (with some scaling). Random points in sample space turn into random points on the light source.

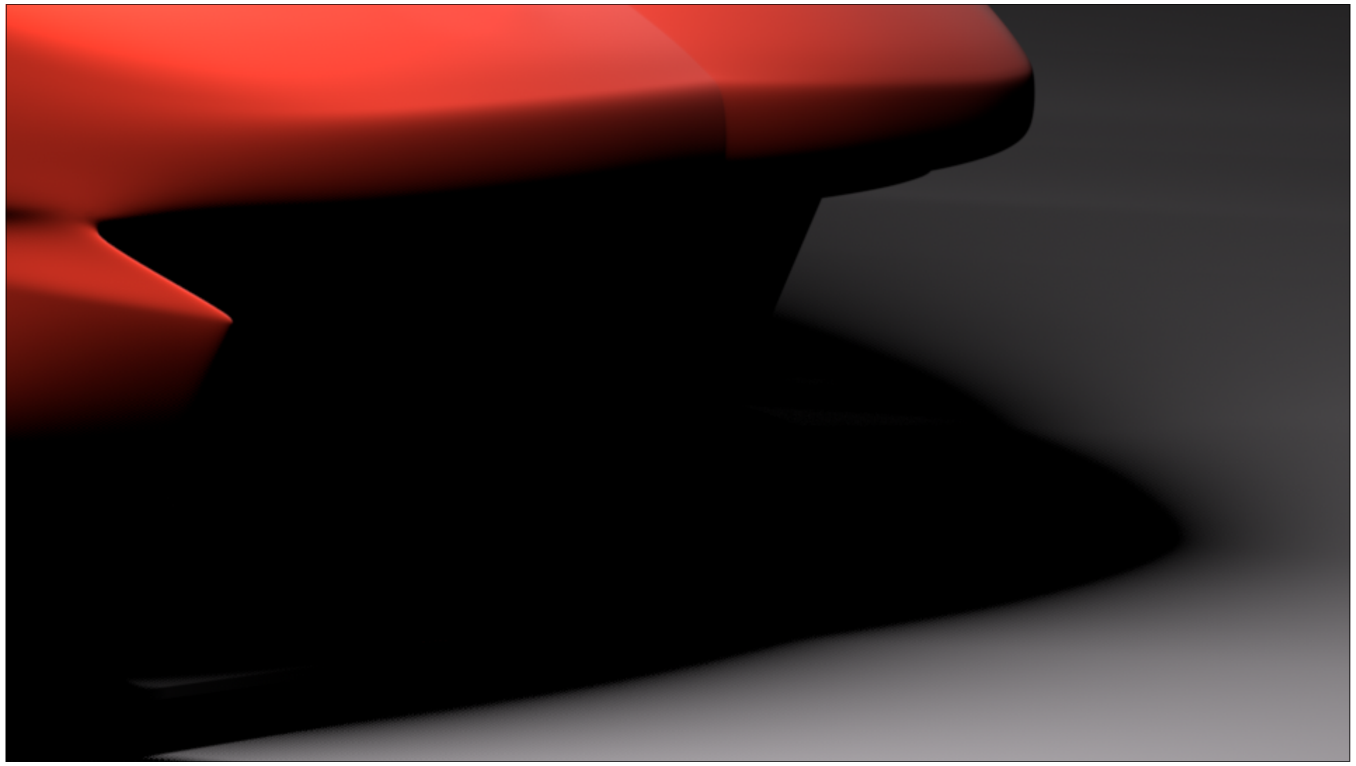


With stratified sampling, we get stratified points on the light source.

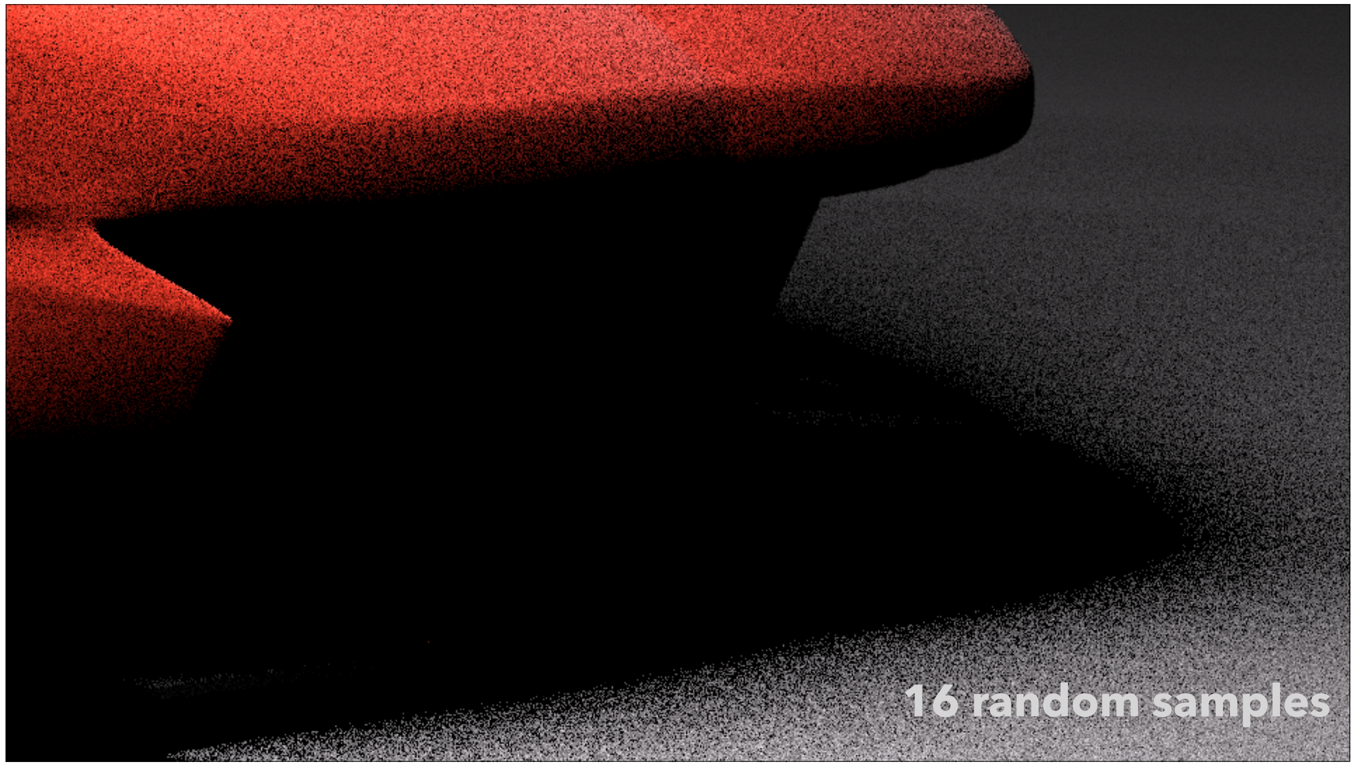
Ignoring visibility, the function's value within each stratum isn't quite constant—there are cosine and 1 over distance squared terms that vary over the strata. But in penumbra pixels, the majority of the variation will be due to the visibility function. And in turn, if that isn't too complex—the edge of a car versus a tree full of leaves—stratification will give a benefit.



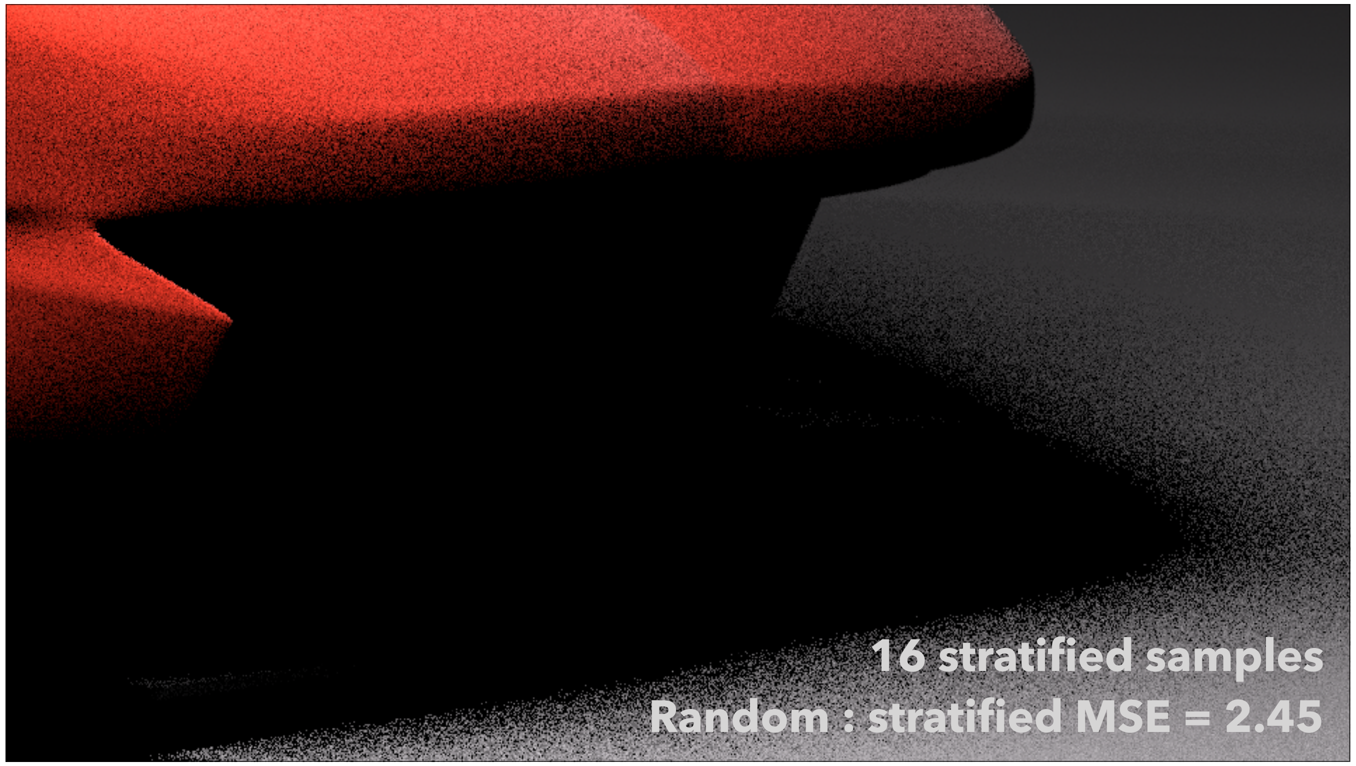
Ok, so let's consider this case. Same car, but now a long and skinny quadrilateral light source.



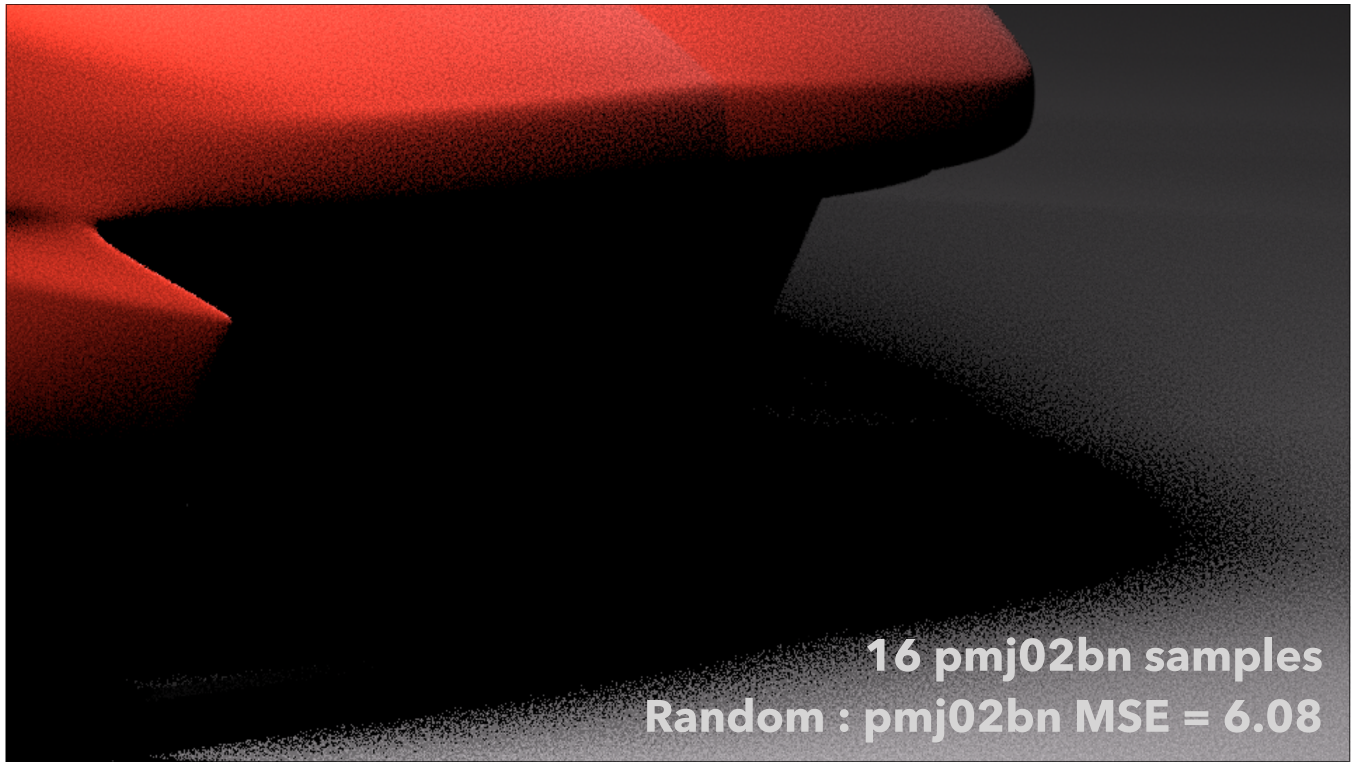
Here’s a reference image of that part we were looking at.



And again, random sampling is a bad idea.

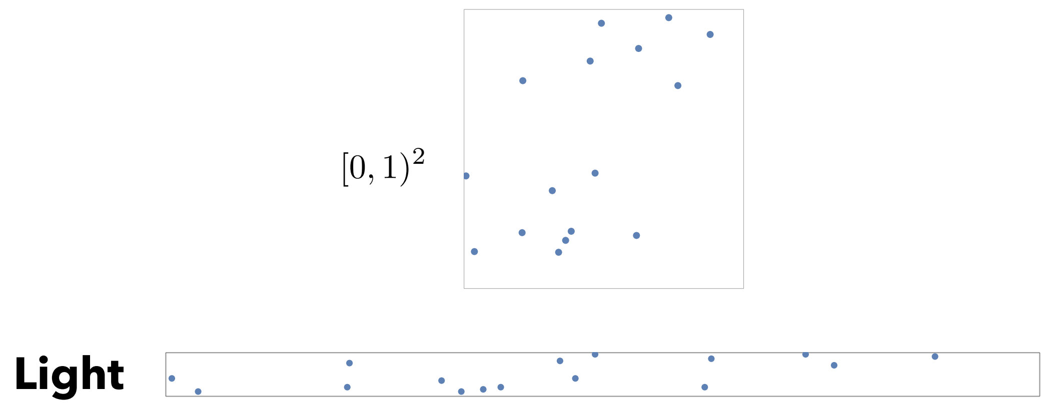


Stratified sampling doesn't help as much as it did with a square light. It's better, but before it was about 8 times better. Now it's about 2.5 times better. That's nice, since it's no more work, but it's kind of disappointing: in a sense, it doesn't seem like a skinny light should be that different than a square light.



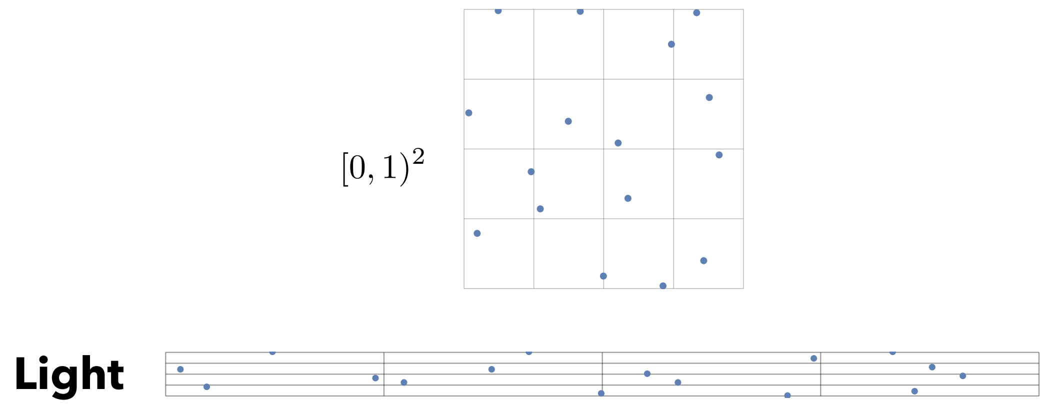
It turns out that the pmj02bn samples do much better in this case—a 6x reduction in error, which is almost as well as with the square light source. That's interesting…

Rectangular Light Point Mapping (Random)



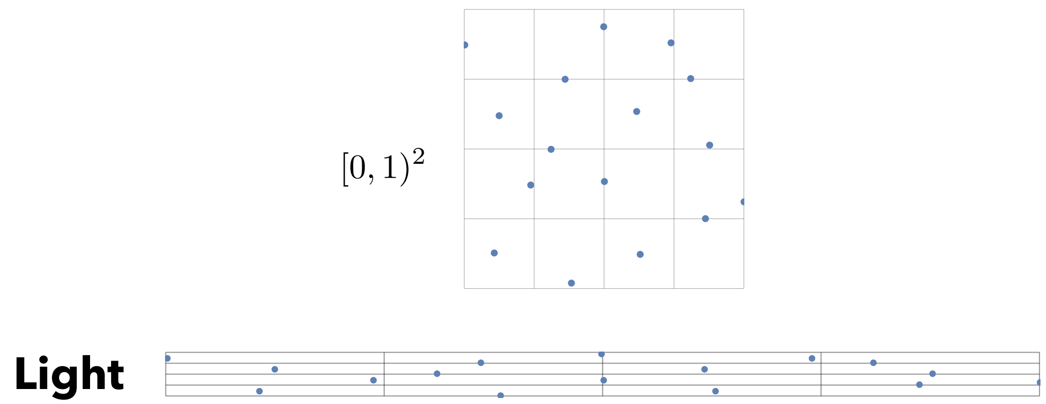
So what happened? Here are random samples mapped to the light source. They end up randomly distributed over it. There's nothing good about their distribution.

Rectangular Light Point Mapping (Stratified)



And here are stratified samples. If we look at the strata on the light source at the bottom, we can see that they're super stretched out. If that seems not great, your intuition is right. What ends up happening is that the samples are pretty much only stratified 4 ways over the light source's area—the stratification in the vertical dimension doesn't matter much. Thus, there's some benefit from stratification, but not as much as we might hope for.

Rectangular Light Point Mapping (pmj02bn)

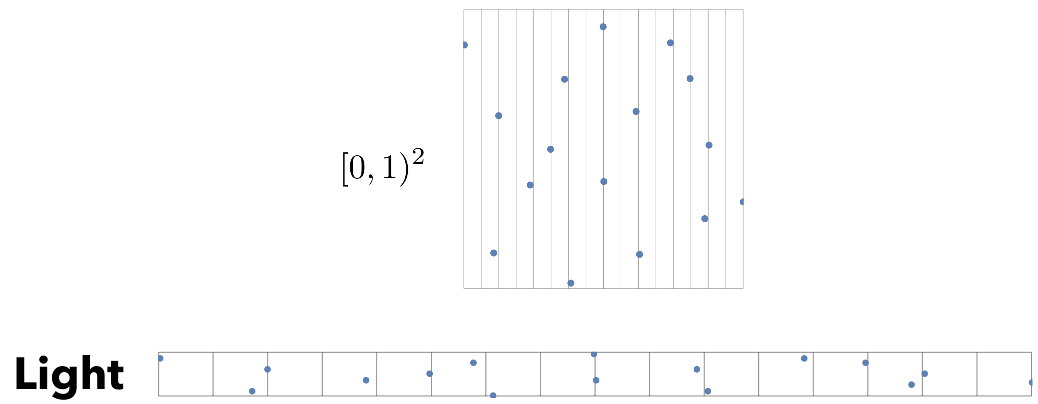


Warp and Effect
My Favorite Samples, SIGGRAPH 2019

Matt Pharr

Here are pmj02bn points, mapped to the light source. They fulfill the 4x4 stratification and so, we see the same ok but not great thing going on with the shapes of those strata on the light.

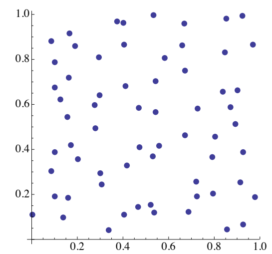
Rectangular Light Point Mapping (pmj02bn)



But here's where the magic happens. The pmj02bn points also fulfill the 16x1 stratification, shown here. In turn, when mapped to the light source, they're well distributed along the length of the light source. The 16x1 stratification turned out to be the one that mattered, given the warping function.

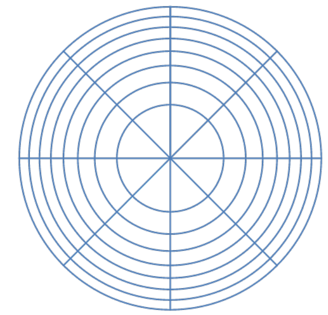
The key point here is that because (0,2)-sequence points like pmj02bn fulfill these very general stratifications, they are in turn resilient to being stretched more in one dimension than another. That's really helpful in practice.

Square to Disk Mappings



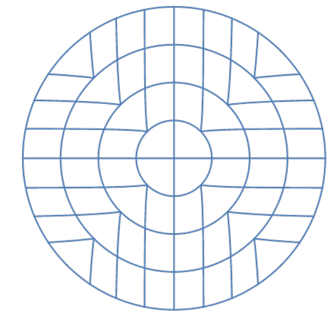
$$\xi_i \in [0,1)^2$$

Polar



$$\theta = 2\pi\xi_1$$
$$r = \sqrt{\xi_2}$$

Concentric



$$r = \xi_1$$
$$\theta = \frac{\pi}{4}\frac{\xi_2}{\xi_1}$$

Warp and Effect
My Favorite Samples, SIGGRAPH 2019

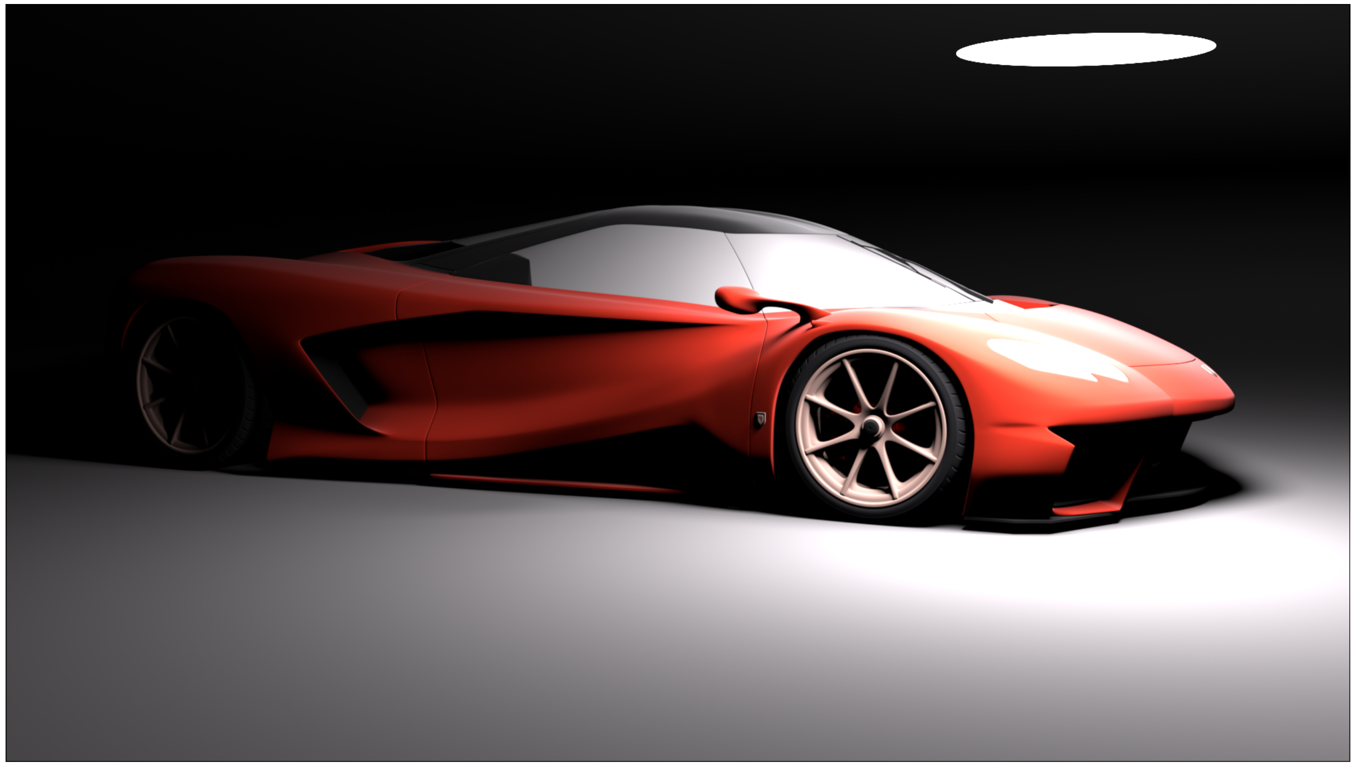
Matt Pharr

So let's move on to consider warping points to the unit disk. You'd think this would be straightforward, but it turns out to have all sorts of surprises.

Here are two warpings that go from the canonical sample domain to the unit disk. Both give uniformly distributed points on the disk. We have the polar mapping, which is the one you can derive if you go and work through the math for how to generate uniform samples given a polar parameterization of the disk.

You can see that the strata on the disk aren't great. In particular, you've got those long and skinny strata by the edges. As we saw with the stretched out quad light, that may be a problem, since it gives more chance for the function we're integrating to vary over a stratum and thus, have higher variance.

Pete Shirley and collaborators saw that problem and came up with the warping on the right, in order to improve the stratification on the disk. The mapping has been adjusted in order to give more compact strata in the target domain.



Once again, we have the car model, illuminated now by a disk light source.

Disk Lighting MSE x 1000 (16 spp)

	Polar	Concentric
Stratified	2.80	2.19
pmj02bn	1.94	2.15

And let's look at error, rendering that part of the image. I won't show the images here, but we'll just look at the error.

First, we can see that for stratified samples, the concentric mapping is much better than the polar mapping. That makes sense, given those better shaped strata we saw earlier.

We can also see that pmj02bn always does better than regular stratified samples, which we'd also expect.

But then there's something interesting and to me, completely unexpected: pmj02bn has about 10% lower error with the polar mapping than with the concentric mapping. What's going on?

Disk Lighting MSE x 1000 (16 spp)

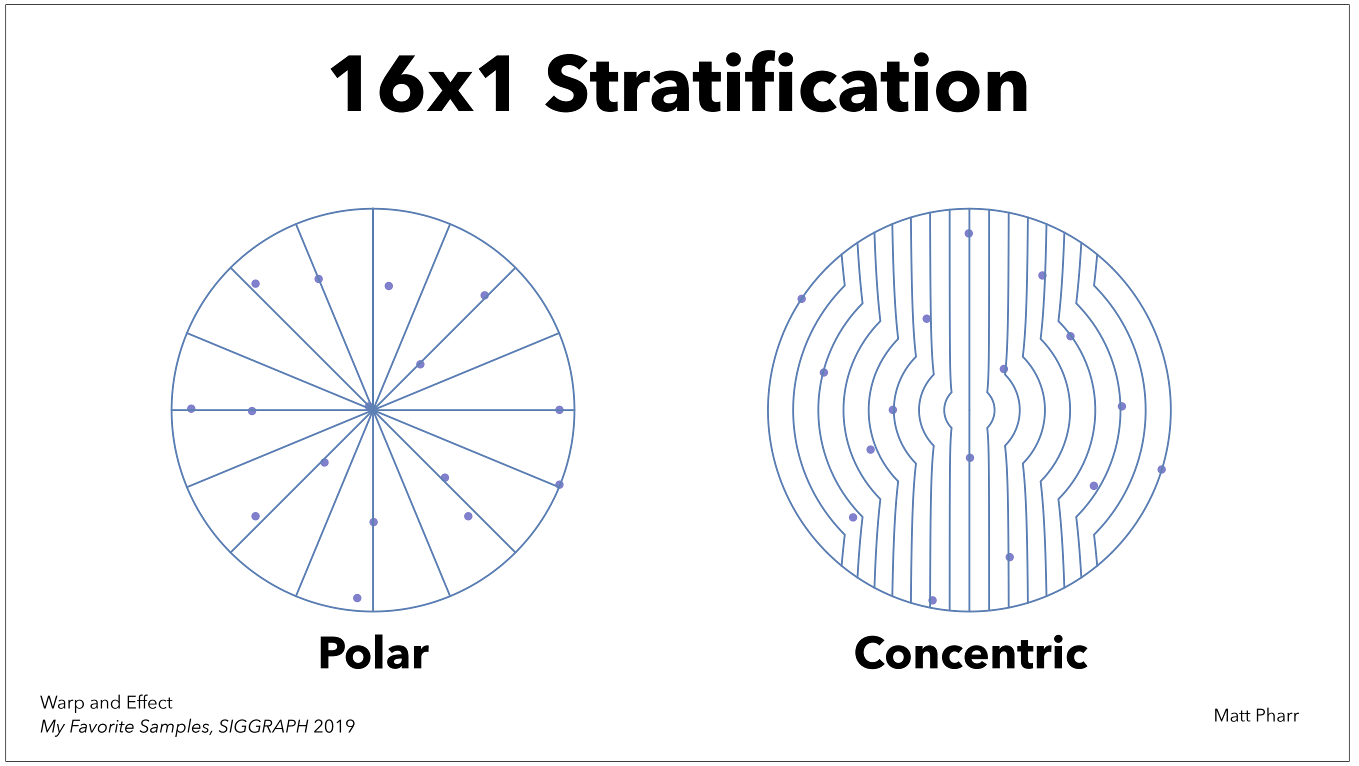
	Polar	Concentric
Stratified	2.80	2.19
pmj02bn	1.94	2.15

Disk Lighting MSE x 1000 (16 spp)

	Polar	Concentric
Stratified	2.80	2.19
pmj02bn	1.94	2.15

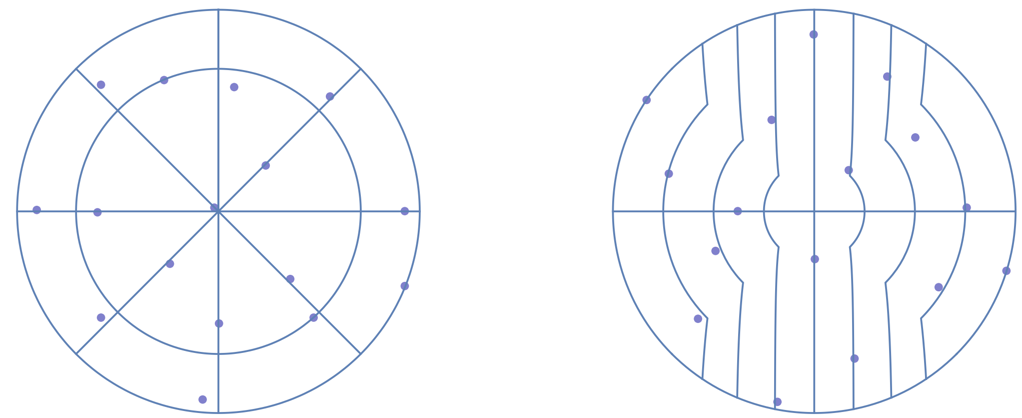
Disk Lighting MSE x 1000 (16 spp)

	Polar	Concentric
Stratified	2.80	2.19
pmj02bn	1.94	2.15



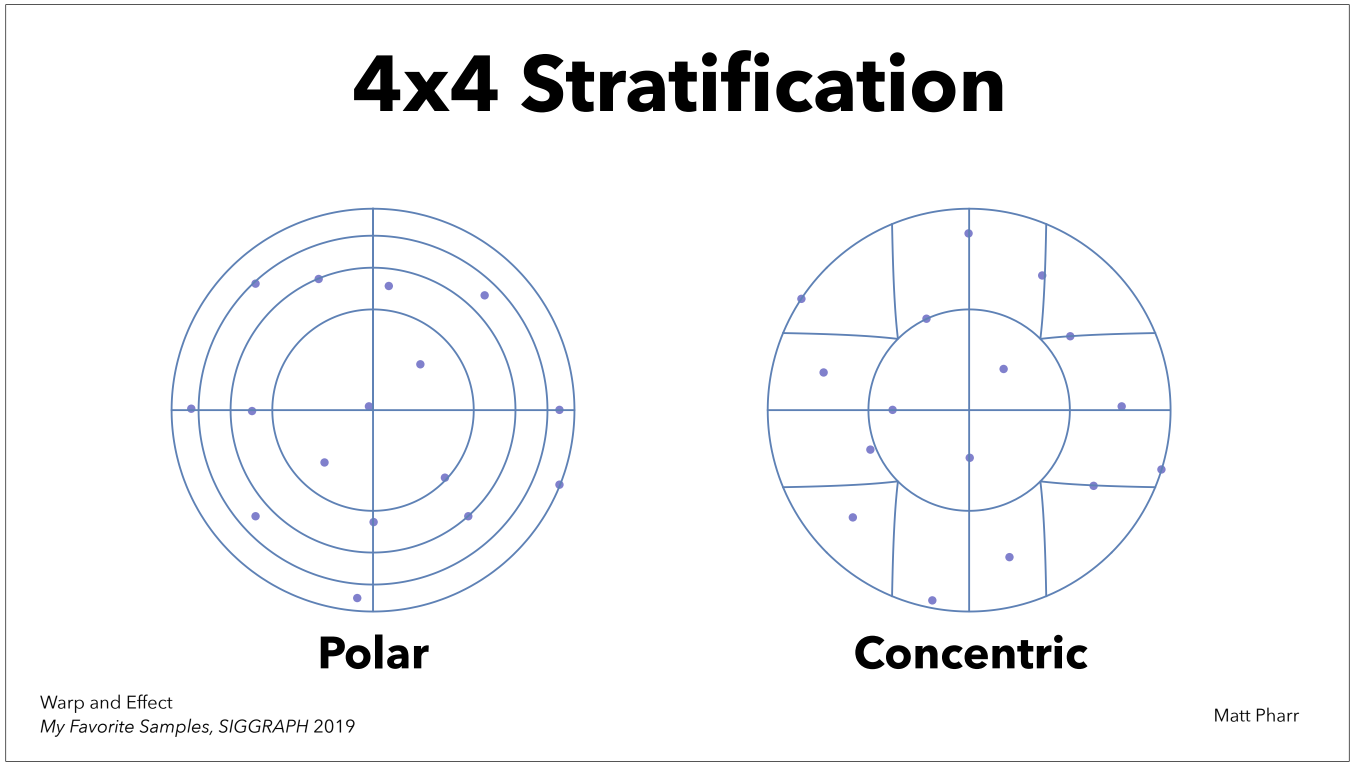
One thing that might give some insight is to look at the shapes of the strata in the target domain, for various stratifications in the source domain. We can see that for 16x1, one of the stratifications that the pmj02bn points satisfy, the polar mapping's strata are more compact than the concentric mapping, where the strata go all the way across the disk.

8x2 Stratification

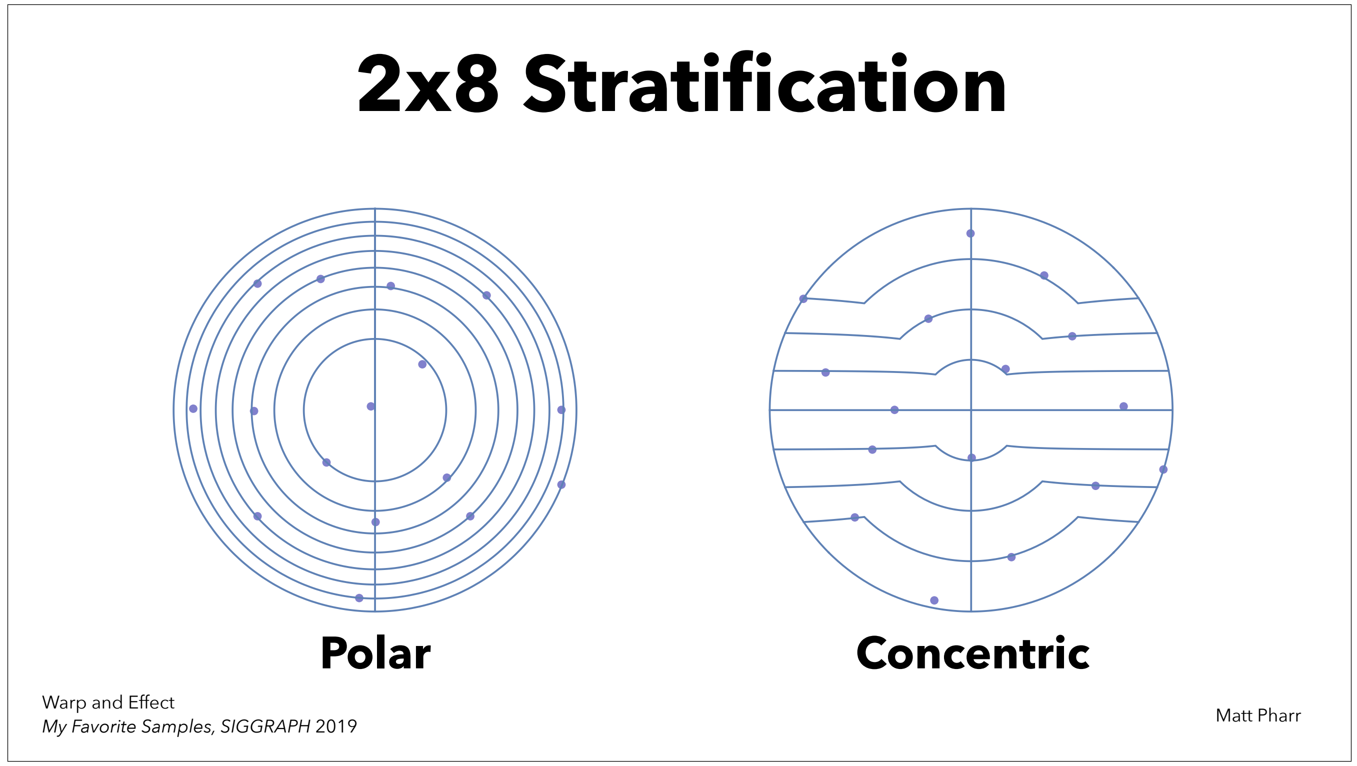


Polar **Concentric**

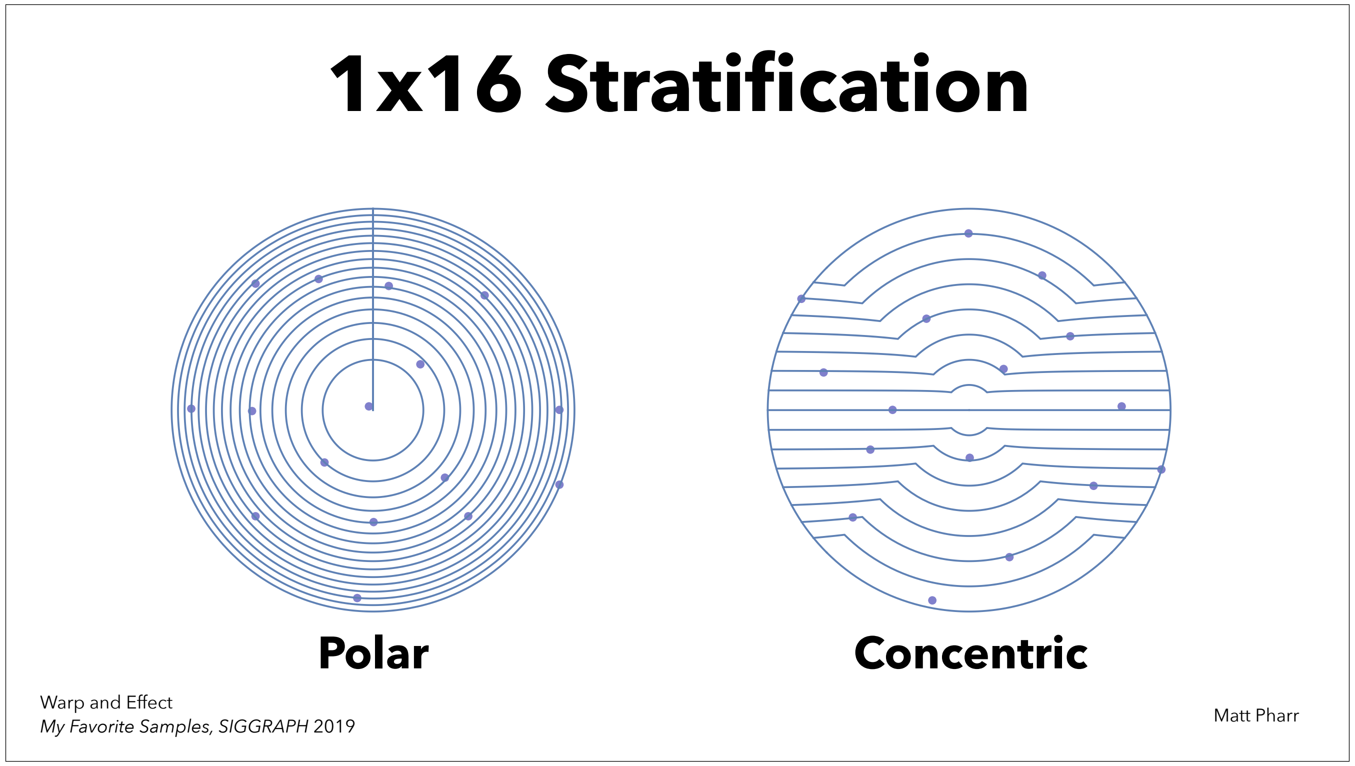
Similarly, with 8x2, to me, the polar strata look better—more compact, less stretched out.



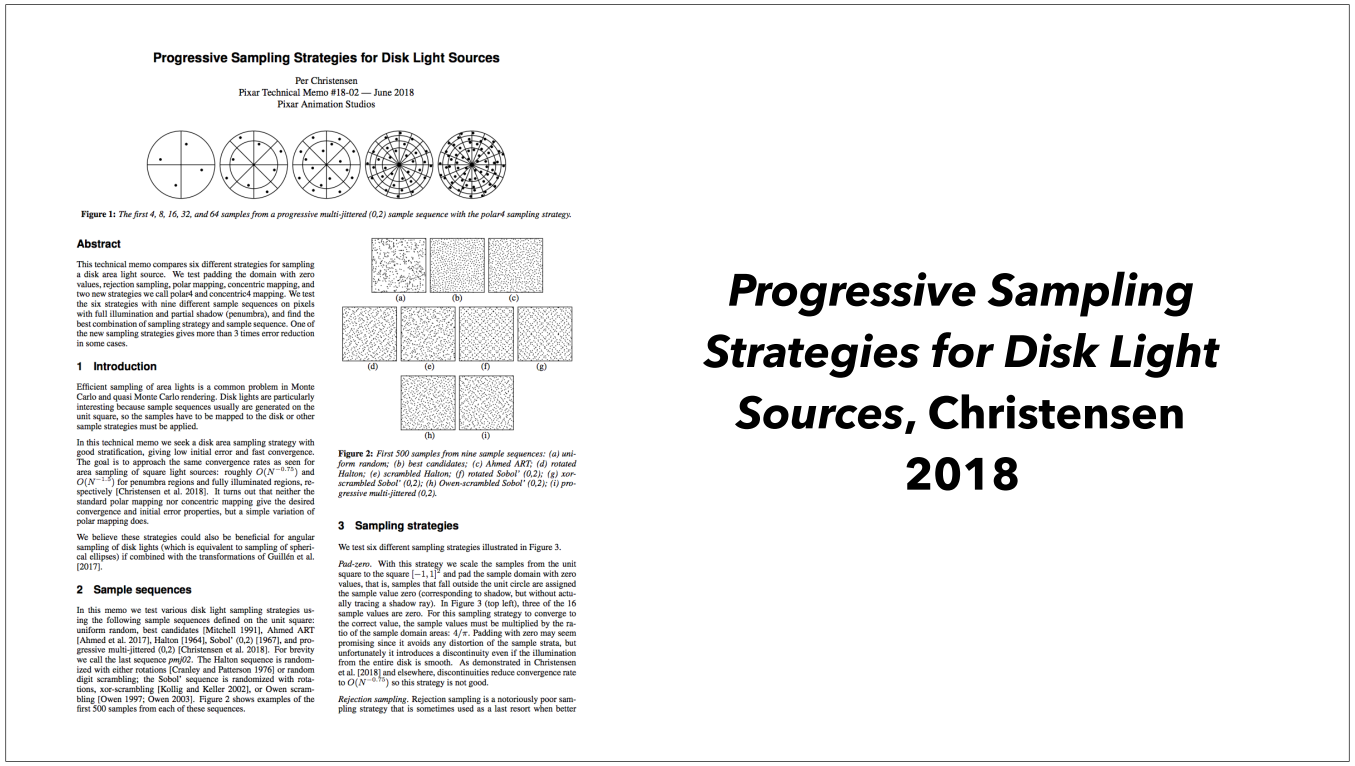
Here’s what we saw before: for equal-sized strata, concentric looks better.



But there's nothing to like about the 2x8 polar strata, wrapping halfway around the disk.



And the 1x16 polar strata are really bad, with the exception, perhaps, of that nice round one in the middle.



Progressive Sampling Strategies for Disk Light Sources

Per Christensen
Pixar Technical Memo #18-02 — June 2018
Pixar Animation Studios

Figure 1: *The first 4, 8, 16, 32, and 64 samples from a progressive multi-jittered (0,2) sample sequence with the polar4 sampling strategy.*

Abstract

This technical memo compares six different strategies for sampling a disk area light source. We test padding the domain with zero values, rejection sampling, polar mapping, concentric mapping, and two new strategies we call polar4 and concentric4 mapping. We test the six strategies with nine different sample sequences on pixels with full illumination and partial shadow (penumbra), and find the best combination of sampling strategy and sample sequence. One of the new sampling strategies gives more than 3 times error reduction in some cases.

1 Introduction

Efficient sampling of area lights is a common problem in Monte Carlo and quasi Monte Carlo rendering. Disk lights are particularly interesting because sample sequences usually are generated on the unit square, so the samples have to be mapped to the disk or other sample strategies must be applied.

In this technical memo we seek a disk area sampling strategy with good stratification, giving low initial error and fast convergence. The goal is to approach the same convergence rates as seen for area sampling of square light sources: roughly $O(N^{-0.75})$ and $O(N^{-1.5})$ for penumbra regions and fully illuminated regions, respectively [Christensen et al. 2018]. It turns out that neither the standard polar mapping nor concentric mapping give the desired convergence and initial error properties, but a simple variation of polar mapping does.

We believe these strategies could also be beneficial for angular sampling of disk lights (which is equivalent to sampling of spherical ellipses) if combined with the transformations of Guillén et al. [2017].

2 Sample sequences

In this memo we test various disk light sampling strategies using the following sample sequences defined on the unit square: uniform random, best candidates [Mitchell 1991], Ahmed ART [Ahmed et al. 2017], Halton [1964], Sobol' (0,2) [1967], and progressive multi-jittered (0,2) [Christensen et al. 2018]. For brevity we call the last sequence *pmj02*. The Halton sequence is randomized with either rotations [Cranley and Patterson 1976] or random digit scrambling; the Sobol' sequence is randomized with rotations, xor-scrambling [Kollig and Keller 2002], or Owen scrambling [Owen 1997; Owen 2003]. Figure 2 shows examples of the first 500 samples from each of these sequences.

(a) (b) (c)
(d) (e) (f) (g)
(h) (i)

Figure 2: *First 500 samples from nine sample sequences: (a) uniform random; (b) best candidates; (c) Ahmed ART; (d) rotated Halton; (e) scrambled Halton; (f) rotated Sobol' (0,2); (g) xor-scrambled Sobol' (0,2); (h) Owen-scrambled Sobol' (0,2); (i) progressive multi-jittered (0,2).*

3 Sampling strategies

We test six different sampling strategies illustrated in Figure 3.

Pad-zero. With this strategy we scale the samples from the unit square to the square $[-1,1]^2$ and pad the sample domain with zero values, that is, samples that fall outside the unit circle are assigned the sample value zero (corresponding to shadow, but without actually tracing a shadow ray). In Figure 3 (top left), three of the 16 sample values are zero. For this sampling strategy to converge to the correct value, the sample values must be multiplied by the ratio of the sample domain areas: $4/\pi$. Padding with zero may seem promising since it avoids any distortion of the sample strata, but unfortunately it introduces a discontinuity even if the illumination from the entire disk is smooth. As demonstrated in Christensen et al. [2018] and elsewhere, discontinuities reduce convergence rate to $O(N^{-0.75})$ so this strategy is not good.

Rejection sampling. Rejection sampling is a notoriously poor sampling strategy that is sometimes used as a last resort when better

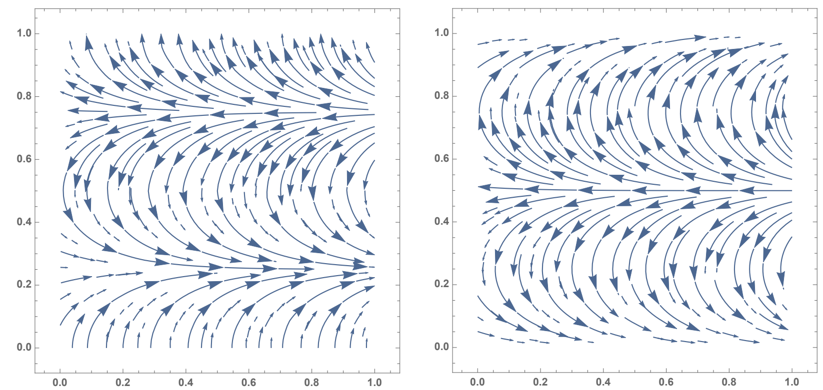
***Progressive Sampling Strategies for Disk Light Sources*, Christensen 2018**

From looking at the strata, it's a little hard to say what's best and what's going on with the pmj02bn samples.

As I was trying to understand these results, I remembered this tech report that Per published last year about sampling disk light sources. Many people have previously encountered unexpected results with sampling disks, and this TR is full of good observations and insight about some of the underlying issues.

Derivative Fields: Polar Mapping

$$f(x_1, x_2) \to (x', y')$$



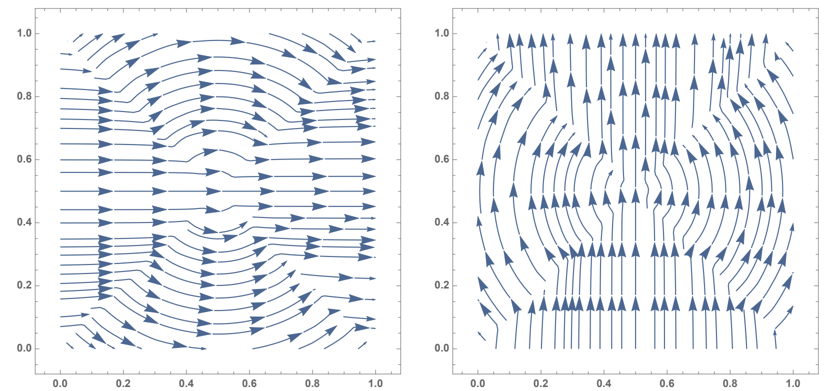
$$\frac{\partial f(x_1, x_2)}{\partial x_1} \qquad \frac{\partial f(x_1, x_2)}{\partial x_2}$$

One of Per's insights was that the polar mapping isn't great in that it pulls different parts of the sample domain in different directions. Here's a visualization of that idea. We're looking at the partial derivative of the mapping from the unit square to the unit disk for the polar mapping. The vectors show the direction that Aat each point in the [0,1)^2 domain will be pulled by the polar mapping.

We can see that across the original sample domain, points are pulled in very different directions. You can think of this as sort of working to break up the points' good stratification properties.

Derivative Fields: Concentric Mapping

$$f(x_1, x_2) \rightarrow (x', y')$$



$$\frac{\partial f(x_1, x_2)}{\partial x_1} \qquad \frac{\partial f(x_1, x_2)}{\partial x_2}$$

And here's a visualization of the derivative fields fo the concentric mapping. These look really good! There obviously has to be some stretching to get from a square to a disk, but these are quite well behaved.

So why doesn't the concentric mapping give better results than the polar mapping for pmj02bn points? I have no idea!

Disk Lighting MSE x 1000 (16 spp)

	Polar	Concentric
Stratified	2.80	2.19
pmj02bn	1.94	2.15

Here again are the error results we saw before, with 16 samples per pixel.

Disk Lighting MSE x 100,000 (256 spp)

	Polar	Concentric
Stratified	7.76	7.20
pmj02bn	7.11	7.14

What happens with 256 samples per pixel? Polar and concentric are essentially the same with pmj02bn points. So maybe all this wondering about what's going on with polar and pmj02bn has been all about nothing.

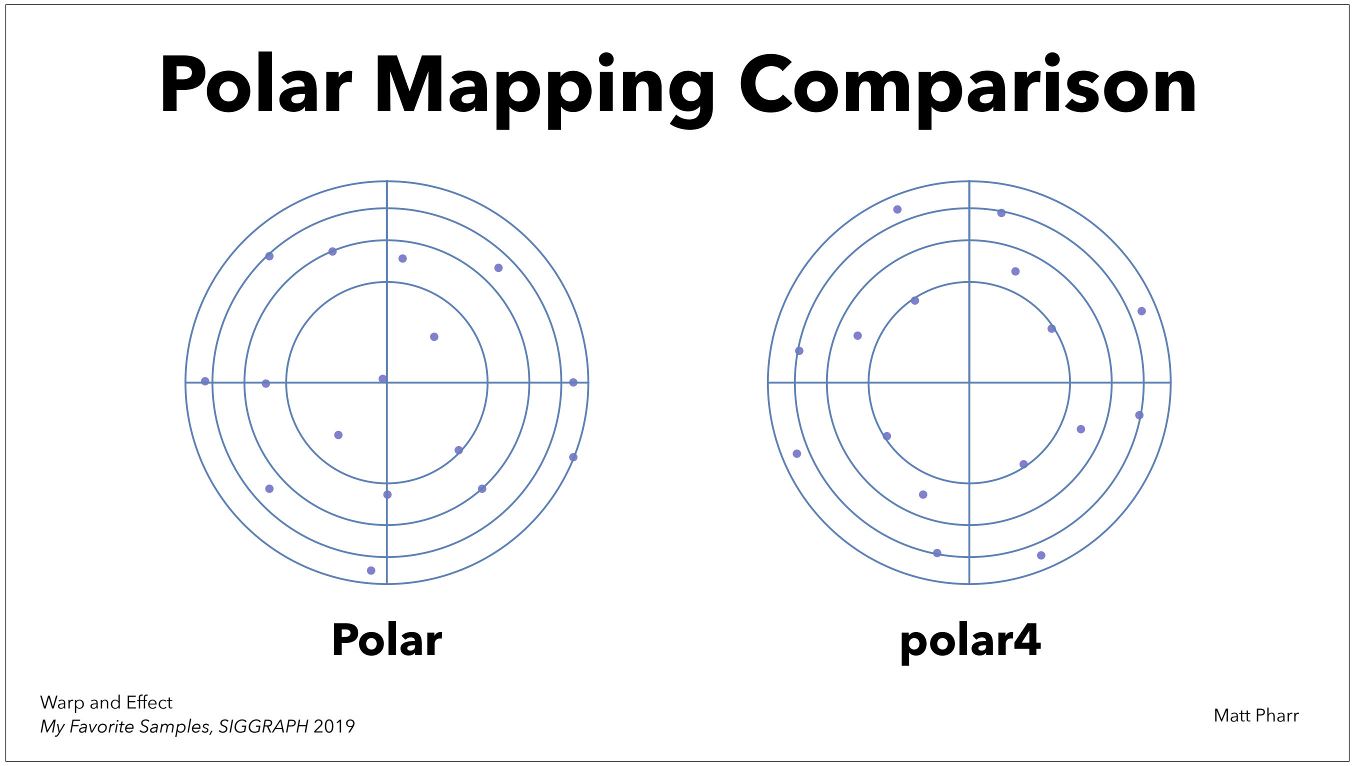
The underlying point here is that the rate of convergence matters, too, and looking at error at just a single sampling rate doesn't give the whole picture. (This point was made nicely in the pmj02bn points paper.)

polar4 mapping

```
polar4(sampleIndex) -> (r, theta)
  u0 = sample[sampleIndex / 4].x
  u1 = sample[sampleIndex / 4].y
  // polar mapping to quarter-disk
  r = sqrt(u0)
  theta = 0.5 * Pi * u1
  // rotate sample
  theta += 0.5 * Pi * (sampleIndex % 4)
```

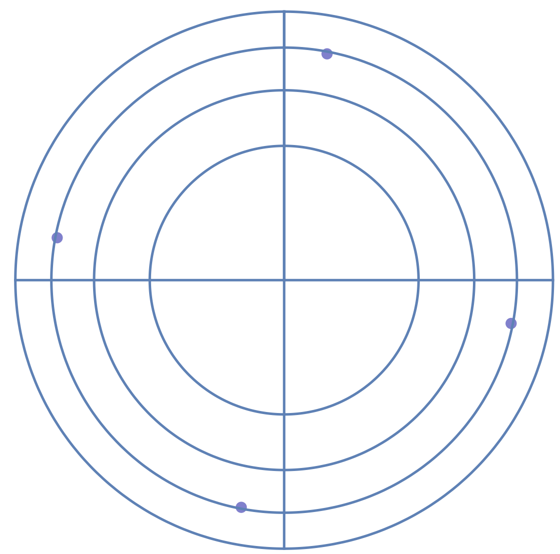
***Progressive Sampling Strategies for Disk Light Sources*, Christensen 2018**

Per's tech report includes an updated polar mapping that tries to counteract that issue of points in different parts of the domain being pulled in different directions. It basically reuses each sample four times, placing a sample on the disk the same place of each of the four quadrants for each one. Here's the algorithm.



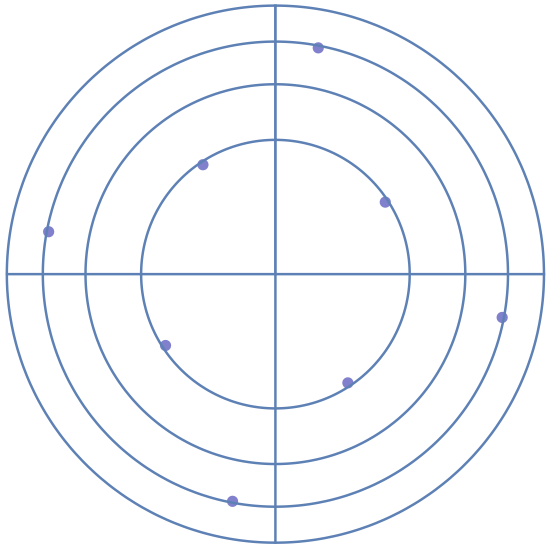
And here are 16 points with the regular polar mapping and the polar4 mapping.

Polar4: First Four Samples



It's easier to see what's going on if you look at the samples four at a time. Here are the first four. Note that they form a square.

Polar4: First Eight Samples

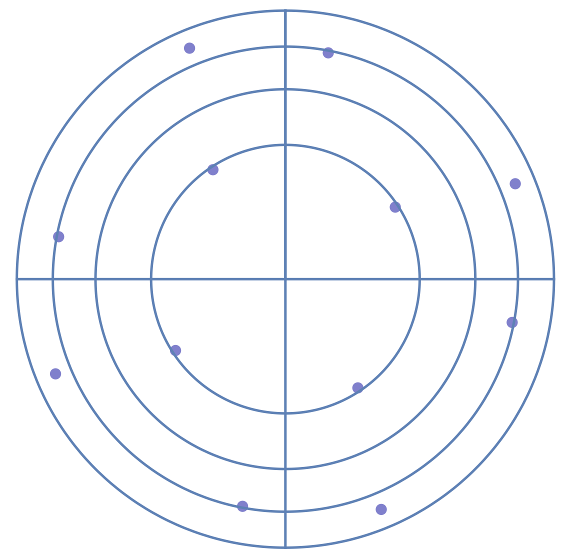


Warp and Effect
My Favorite Samples, SIGGRAPH 2019

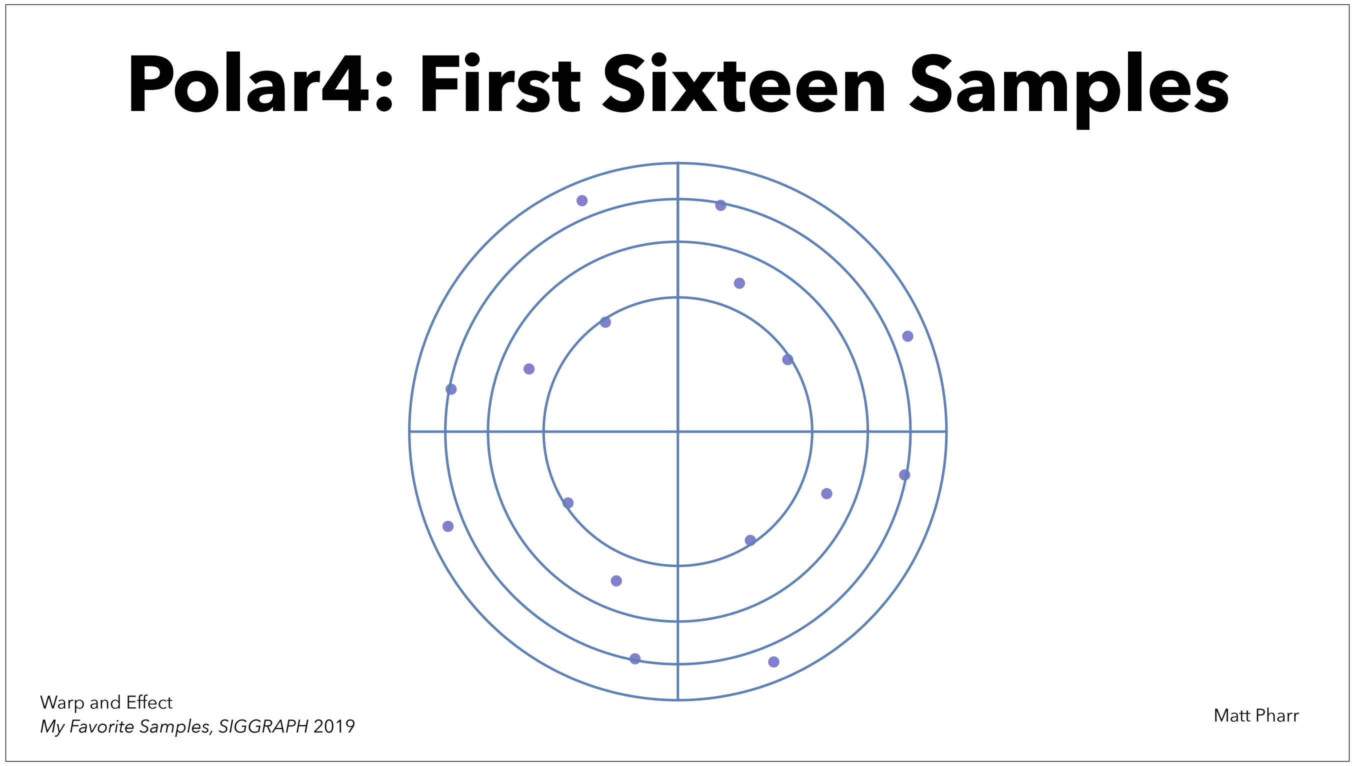
Matt Pharr

And here are the next four: another square, closer to the center.

Polar4: First Twelve Samples



And the next four…



And the last four, getting us to sixteen. Some structure has been added to the sample points.

Disk Lighting MSE x 100,000 (256 spp)

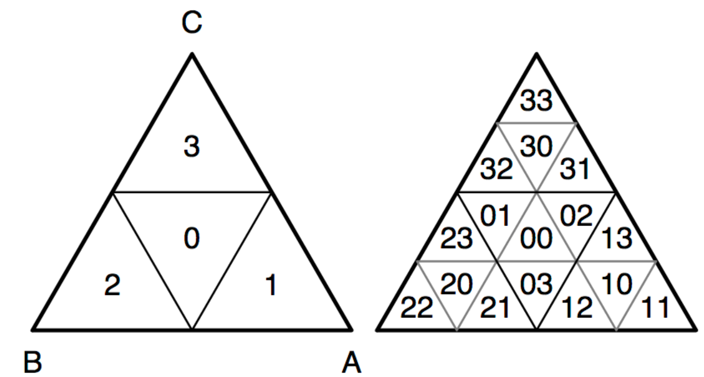
	Polar	Concentric	Polar4
Stratified	7.76	7.20	4.14
pmj02bn	7.11	7.14	3.82

And it really makes a difference—the polar4 mapping gives substantially lower error with both stratified and pmj02bn points.

The one unfortunate thing about this mapping is that it's tricky from a software engineering standpoint: you need to take a multiple of four light samples, and you end up consuming samples in a different way for a disk light source than for other types of light source.

Domain-Specialized Sampling

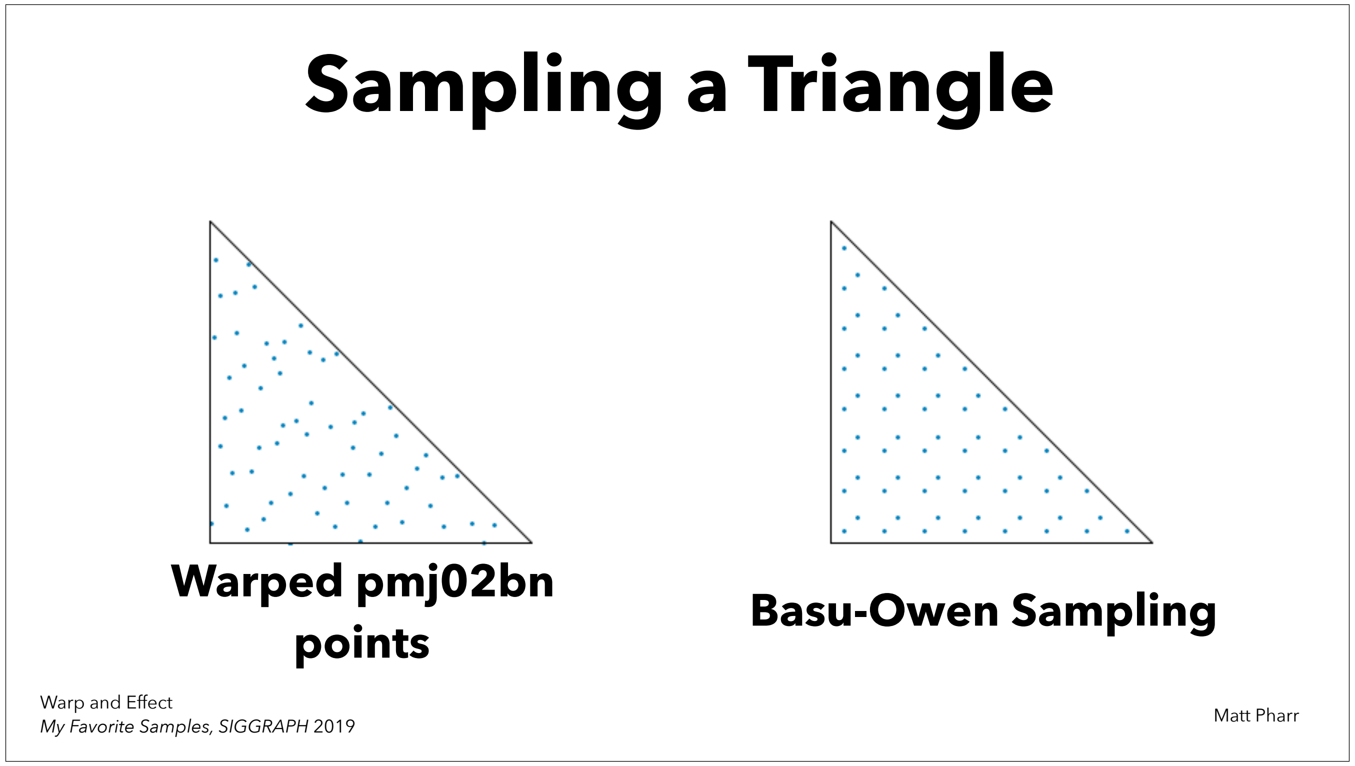
- Take van der Corput points base 4, and index into triangle



***Low discrepancy constructions in the triangle*, Basu and Owen 2014**

Related, there's a method for sampling triangles that Basu and Owen suggested. They use a single 1D sample value, based on the integers starting from zero. They then take the base 4 radical inverse and use the digits of that to walk down a hierarchical decomposition of the triangle, as shown here.

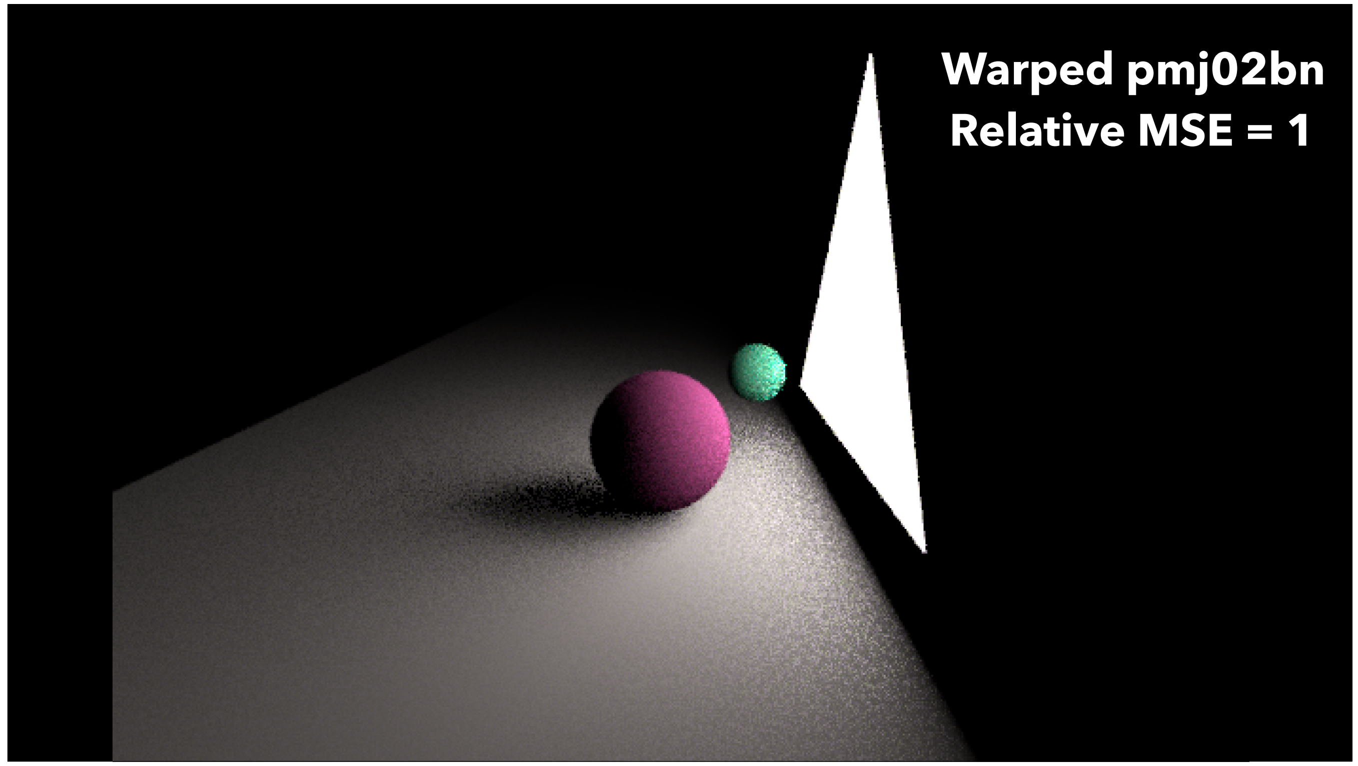
As it turns out, computers are good at base 4 arithmetic as well as base 2: you just need to consider pairs of binary digits…



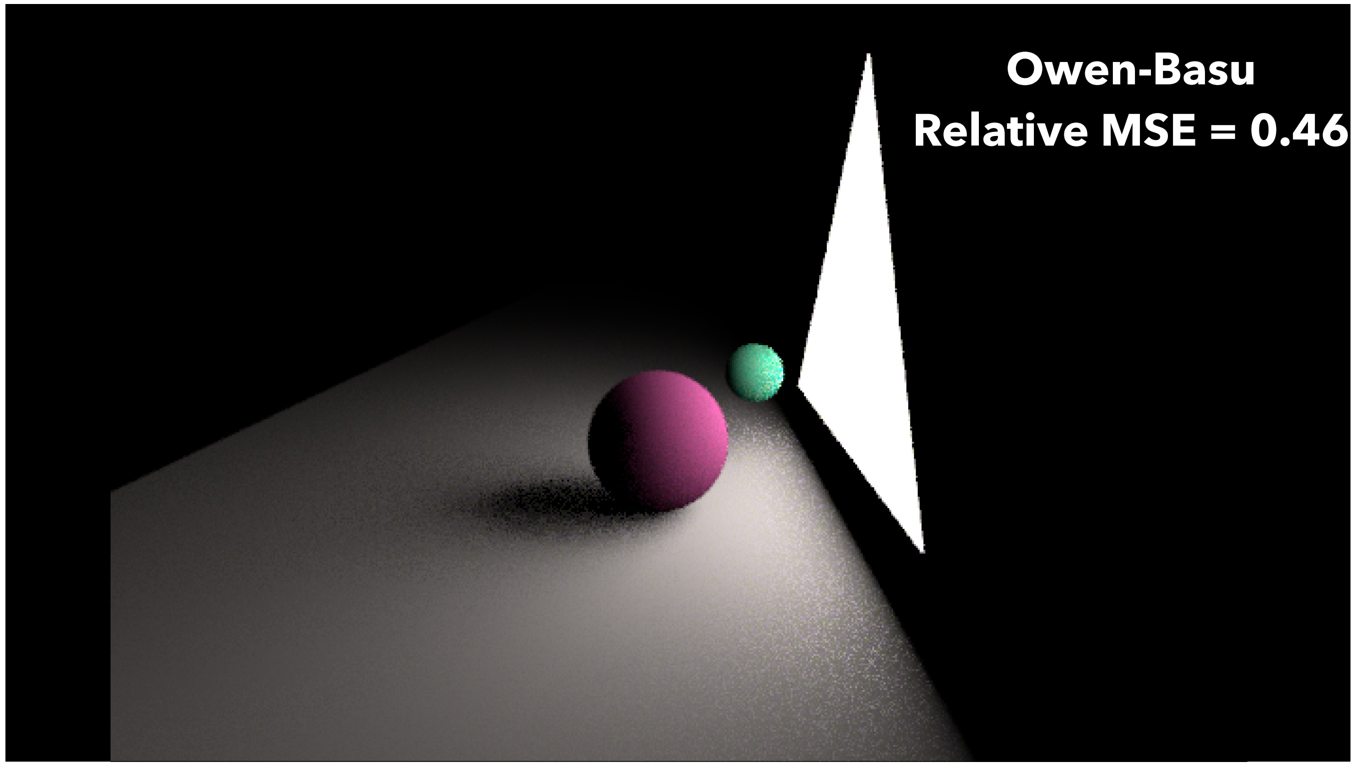
And here are the points you get in a triangle. On the left, we have pmj02bn points warped with the traditional square to triangle mapping, which gives uniformly-distributed points.

On the right, we have Basu and Owen's approach, which also gives uniformly-distributed points. You can see that, like the polar4 mapping, there's some structure in the points.

How does that work out in practice?



Here’s a simple scene with a triangle area light source, rendered with pmj02bn points and the traditional mapping to the triangle.



And here it is using the Basu-Owen mapping. Mean squared error is more than halved—that's quite nice!

So again, just like with the polar4 mapping, we seem to have some evidence that maintaining some structure in the sample point locations has value. I don't know that this idea is well understood in general, but I hope that folks will go forth and figure more of this out…

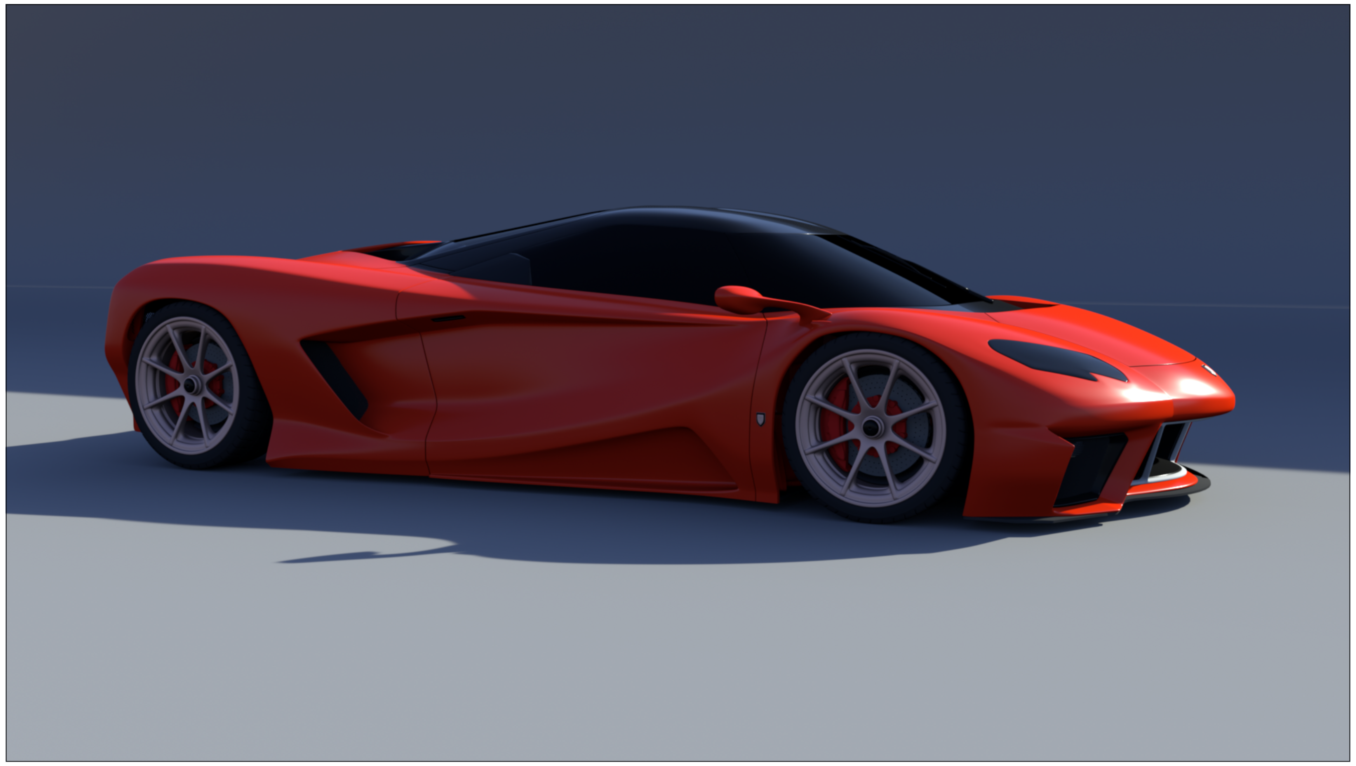
HDRI Haven: Qwantani

For our last topic, we'll consider rendering with HDR environment maps. We have an image that defines a function over the sphere of directions and we'd like to generate samples with probability proportional to the brightness in each direction.

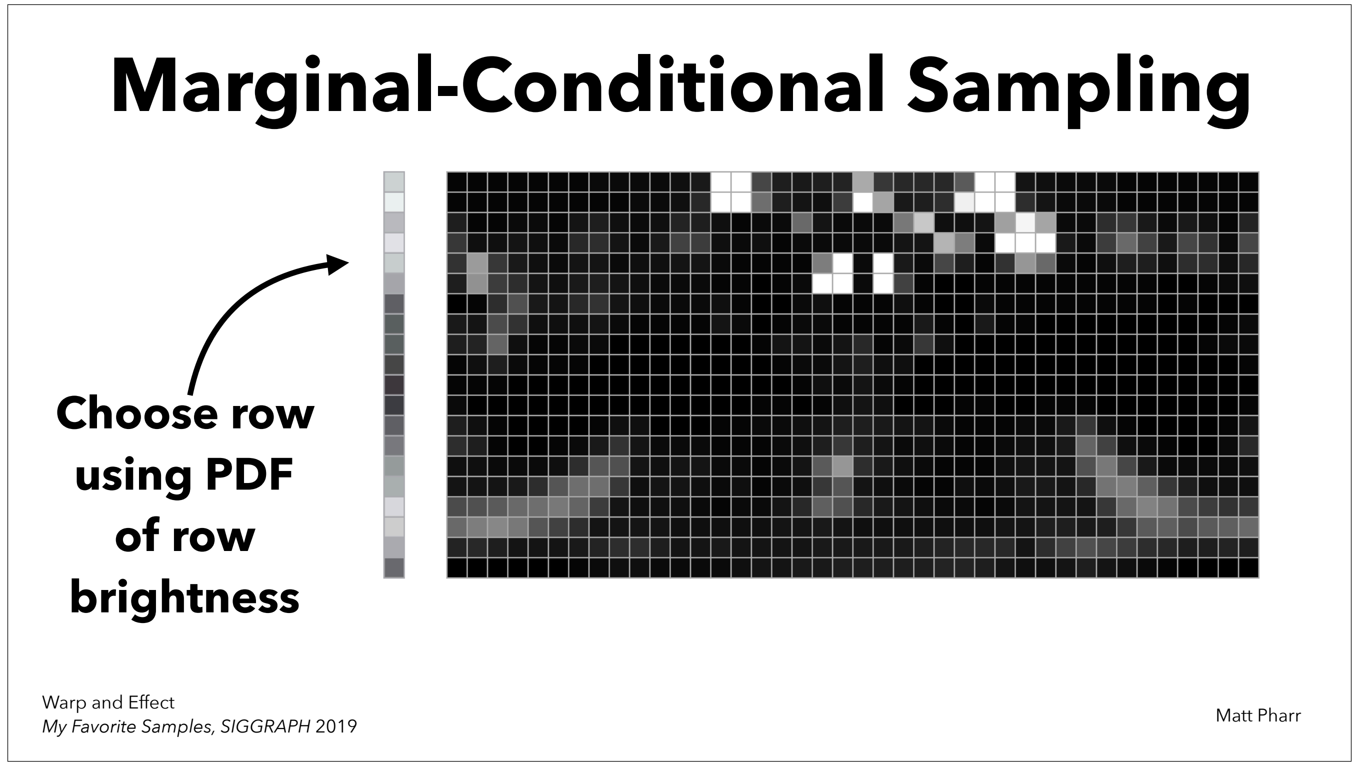
Here's the first environment map that we'll look at: it includes the daytime sky, no clouds, and the sun. The brightest pixels are about 50,000 times brighter than the darkest ones.

HDRI Haven: Surgery

Here’s the second environment map we’ll consider. This interior scene has roughly a 300:1 dynamic range.



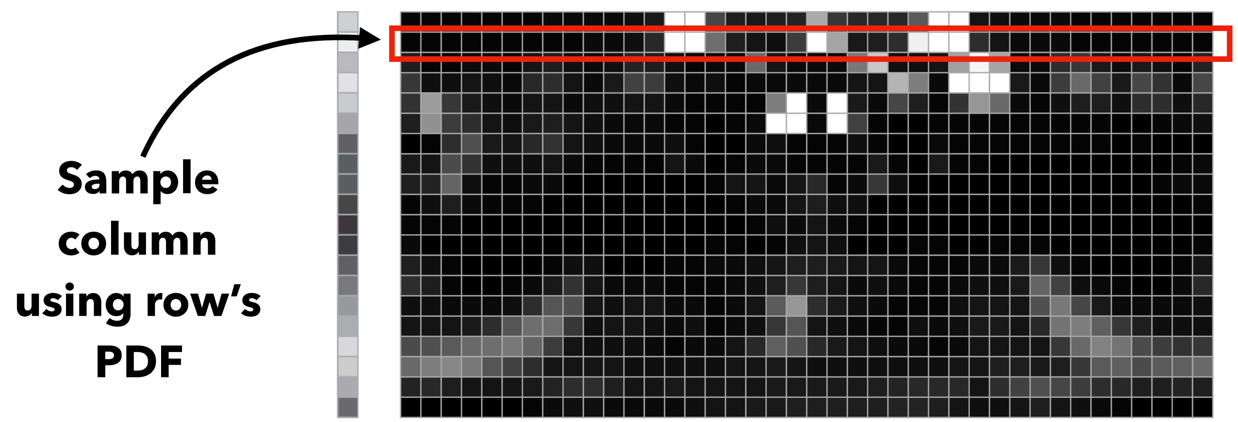
And here’s the trusty sportscar, lit by the outdoor environment map.



There are two widely-used techniques for warping samples to images and, specifically, to environment maps.

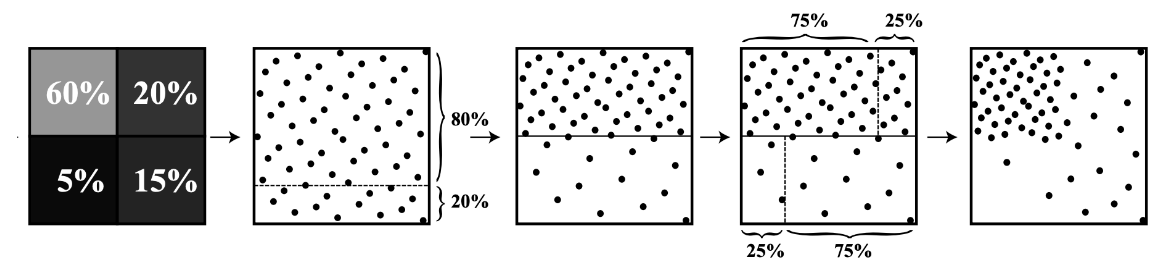
First, you can express the distribution over pixels as the combination of a 1D marginal distribution which leads to a 1D conditional distribution. Essentially, you're summing up the values across each scanline and using those to define a PDF over scanlines. You warp one sample dimension using that PDF to choose a scanline.

Marginal-Conditional Sampling



Each scanline has its own 1D PDF. You warp the second sample dimension using that, which gives you a pixel in the image. That pixel ends up being sampled with probability proportional to its relative brightness.

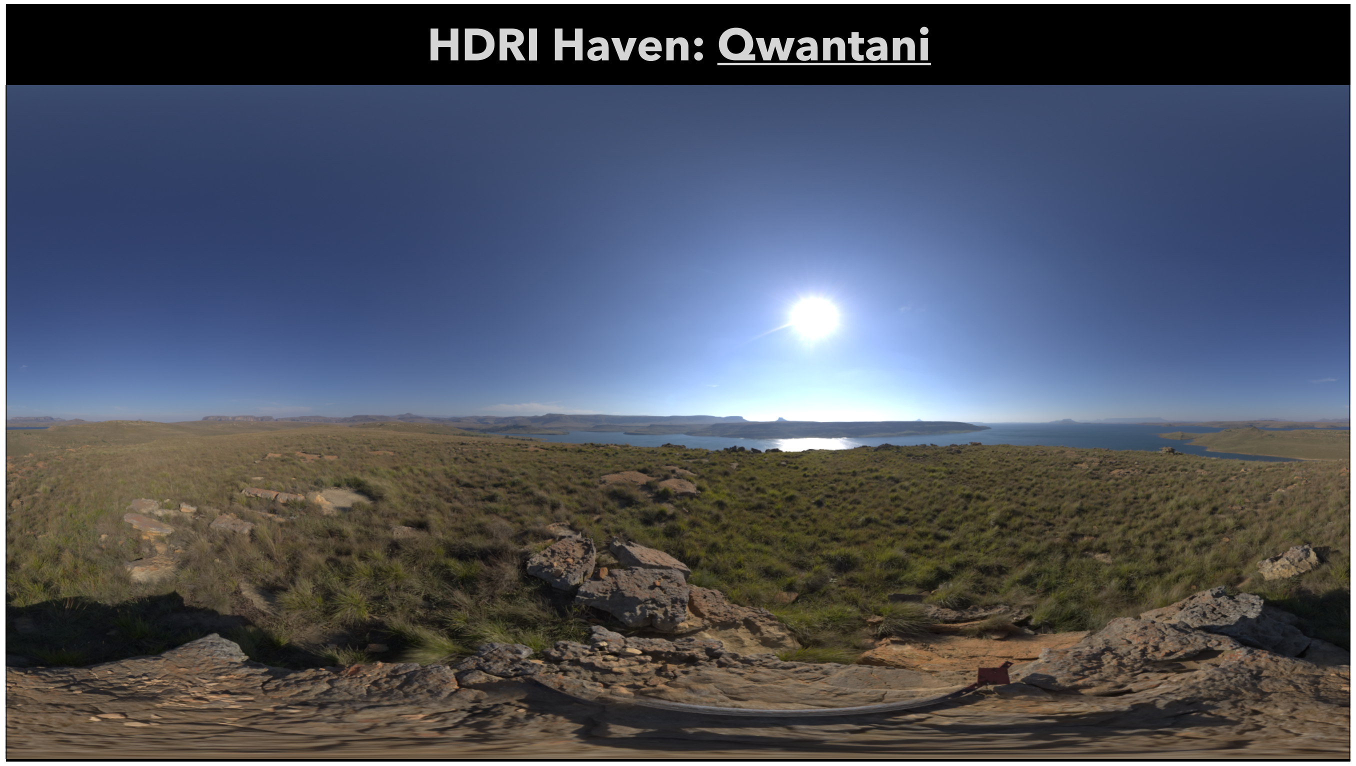
Hierarchical Sample Warping



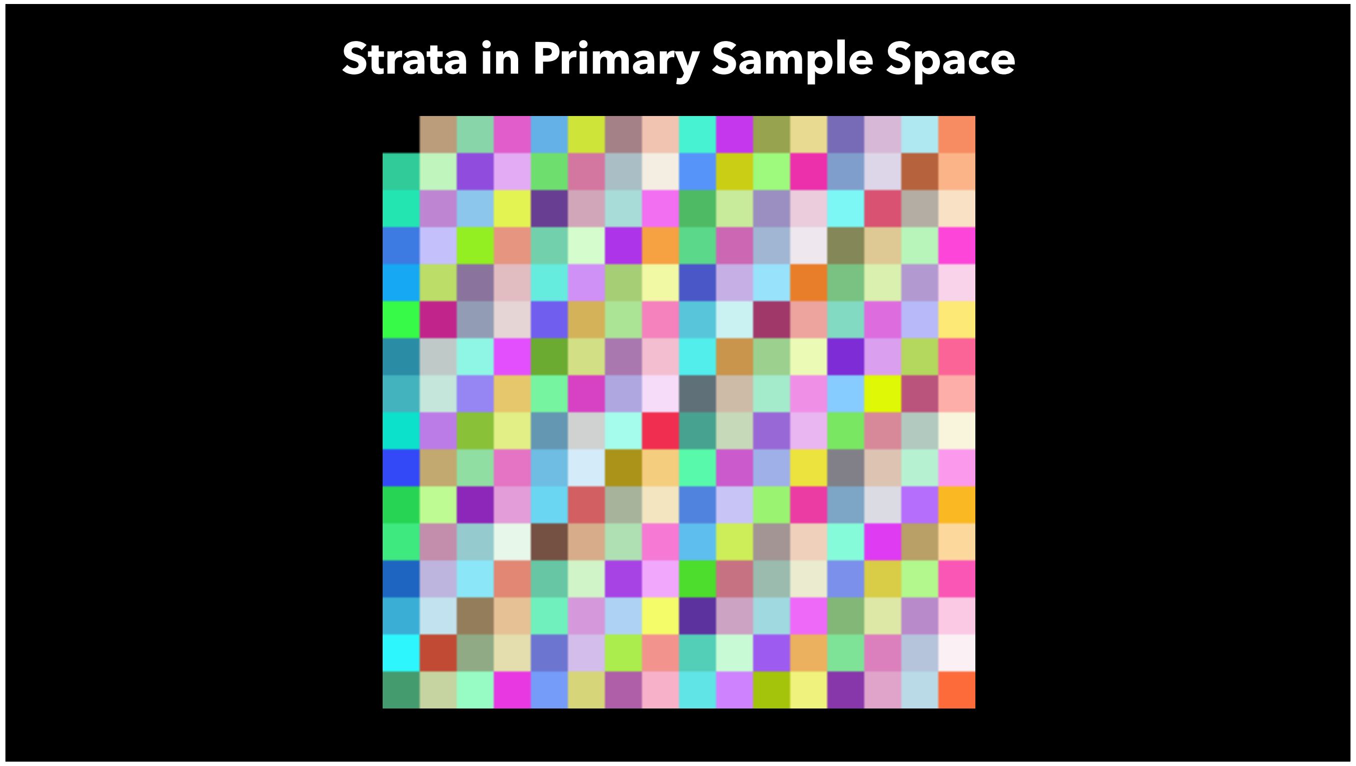
***Probability trees*, McCool and Harwood 1997**
***Wavelet importance sampling: efficiently evaluating products of complex functions*,**
Clarberg et al. 2005

The other option is a hierarchical sample warping algorithm. First, you build a MIP map of the image. Then, you progressively warp the [0,1)^2 sample domain such that the area of the primary sample space associated with each region of the MIP map is proportional to the relative brightness of that region of the MIP map.

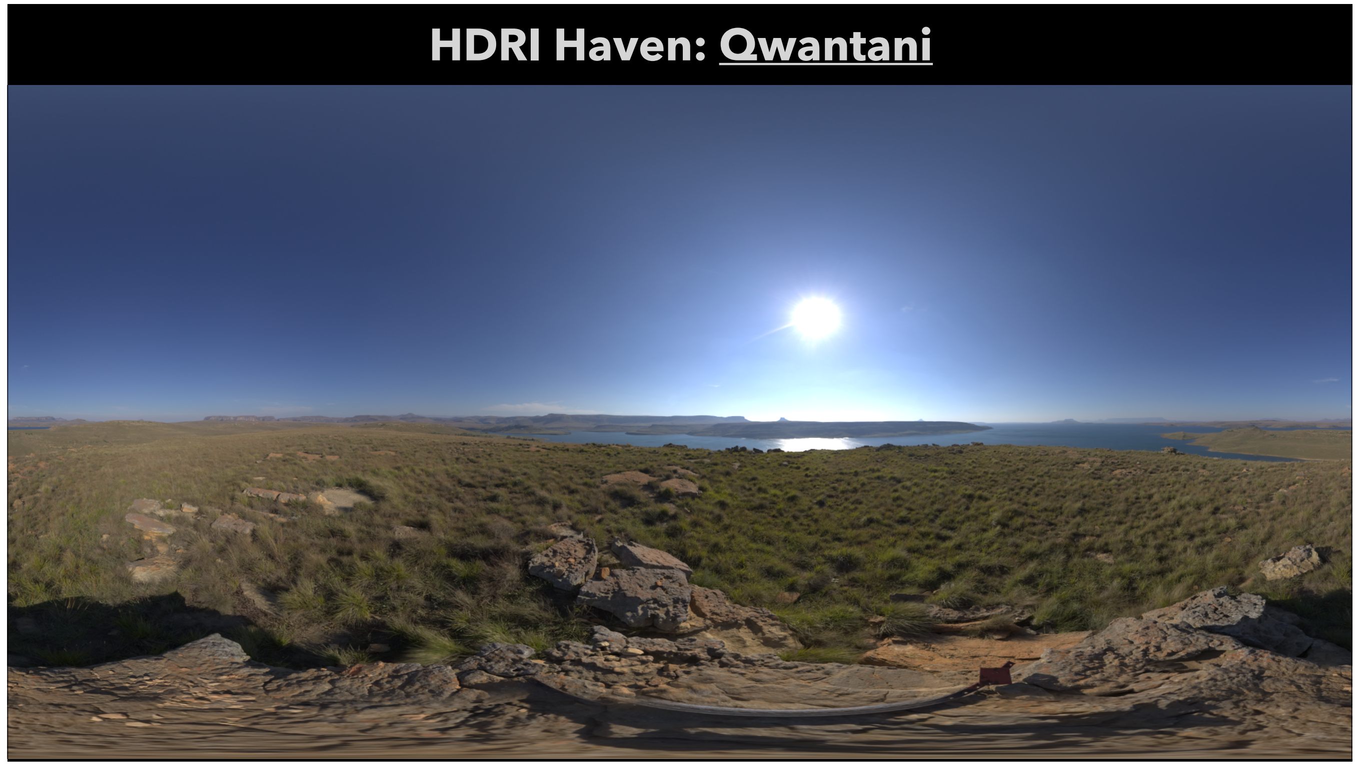
Here we can see an illustration the idea, where a collection of sample points are all warped. In practice, one generally has a single sample point that is successively linearly warped in each dimension at each MIP map level.



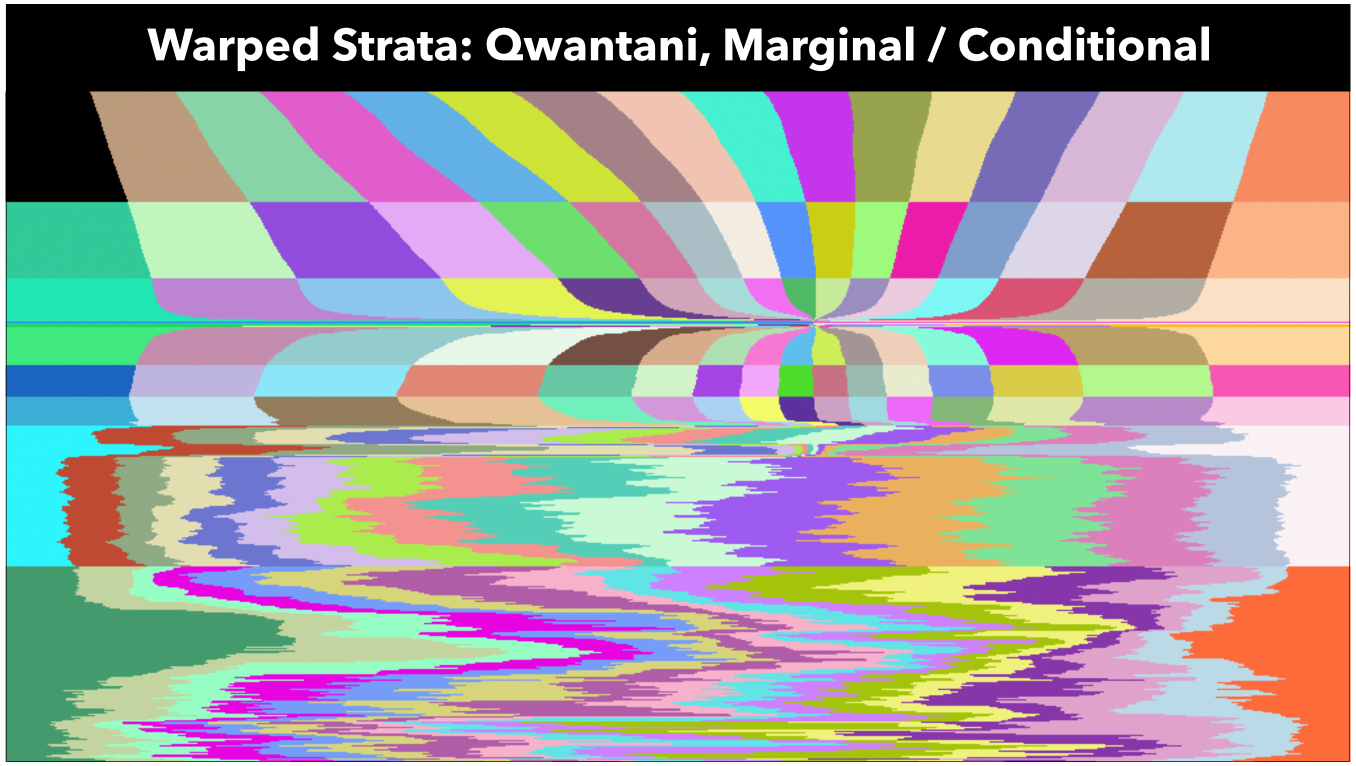
Let’s look at what these warping methods do to sample points. Again, for reference, here’s that outdoor environment map.



And here’s a visualization of the [0,1)^2 sampling domain, with 16x16 strata, each given its own color. We’ll look at how this is stretched by the two warps.



Again, the first environment map.

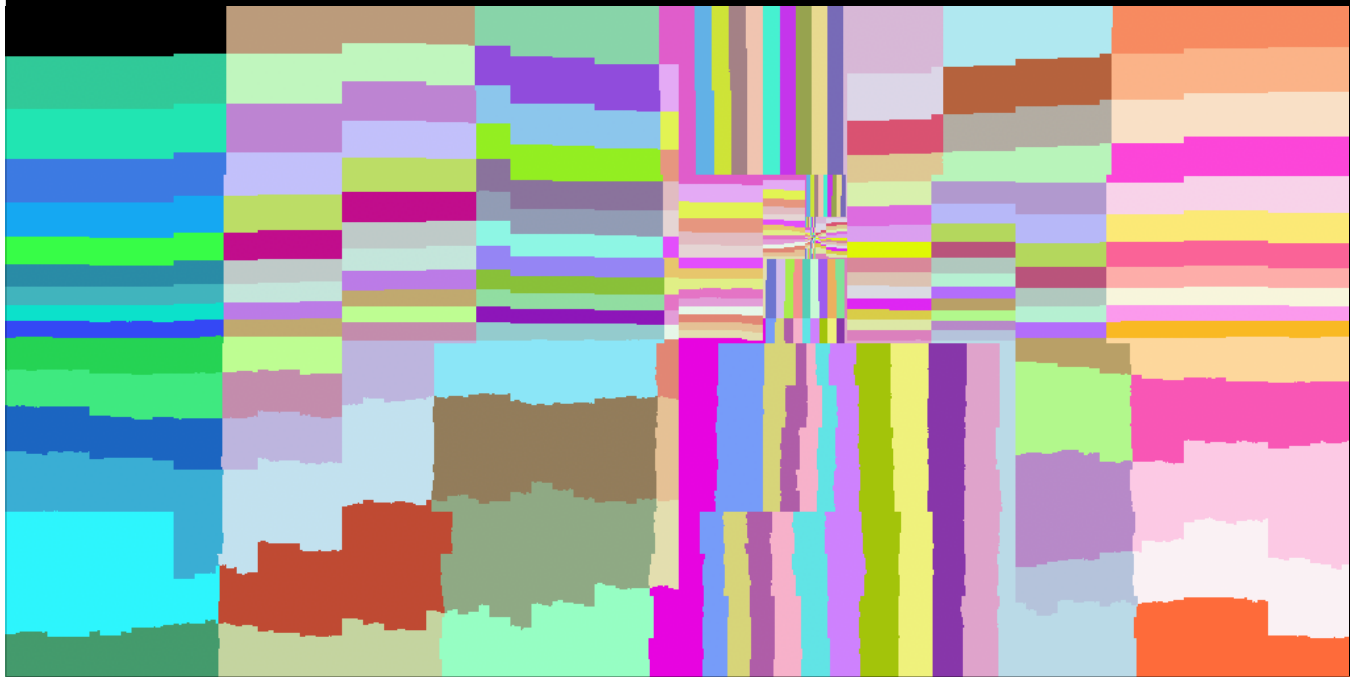


And here are the warped strata with the marginal conditional warping.

One thing to notice is what happens with the scanlines around the sun—all around it, the strata are distorted, pulling towards it.

It's also interesting to see what happens at the bottom, where the ground is—even there, the strata are super wavy. What's happening is that each scanline has its own independent 1D PDF and over the extent of a scanline, it may end up distorting samples quite differently than it's neighbors—there's nothing to cause the strata to hold together.

Warped Strata: Qwantani, Hierarchical



It's quite different with the hierarchical warping. The strata seem better behaved for the most part, though we can see some that are quite elongated. You can also see a few that have been distorted into strange shapes—consider the grey one at the bottom on the right—it has three rectangular sections.

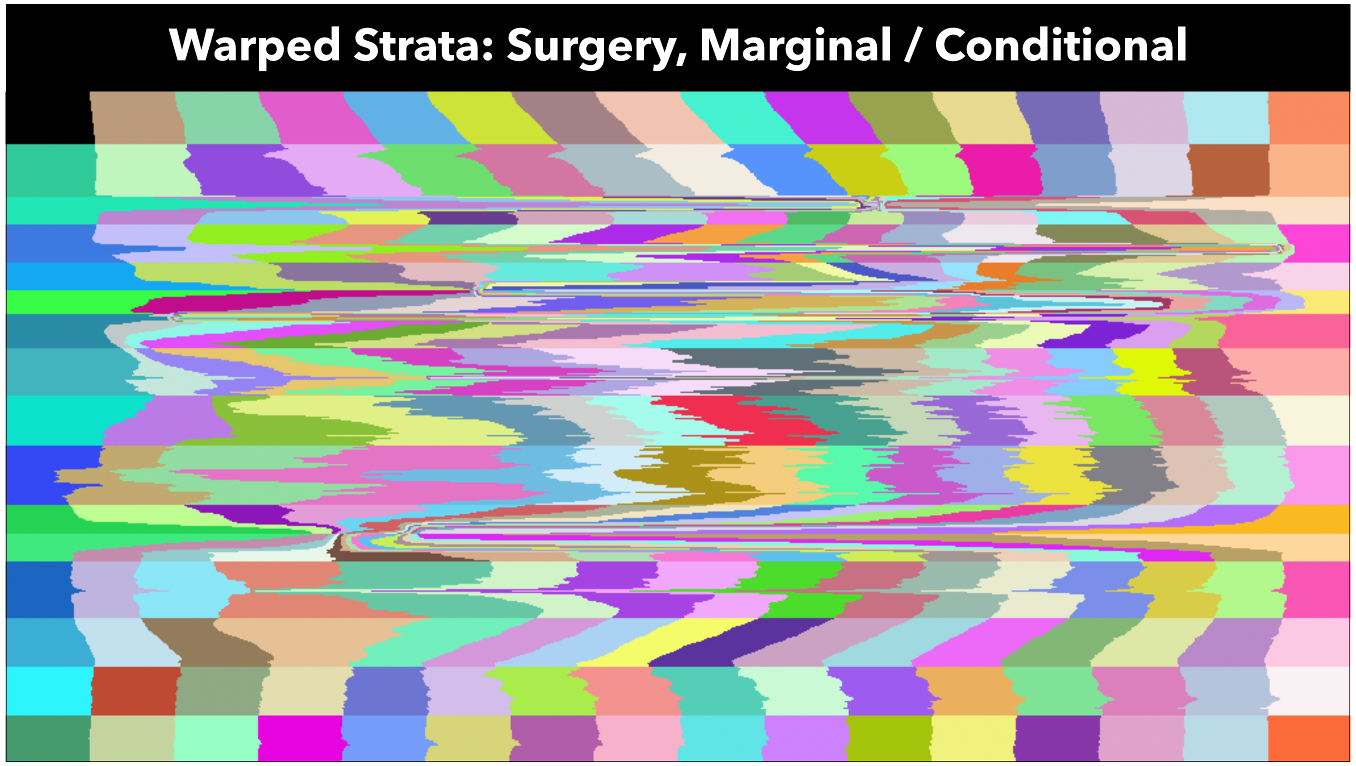
Results: Qwantani
16spp, MSE x 1000

	Marg/Cond	Hierarchical
Stratified	1.22	1.16
pmj02bn	0.88	0.80

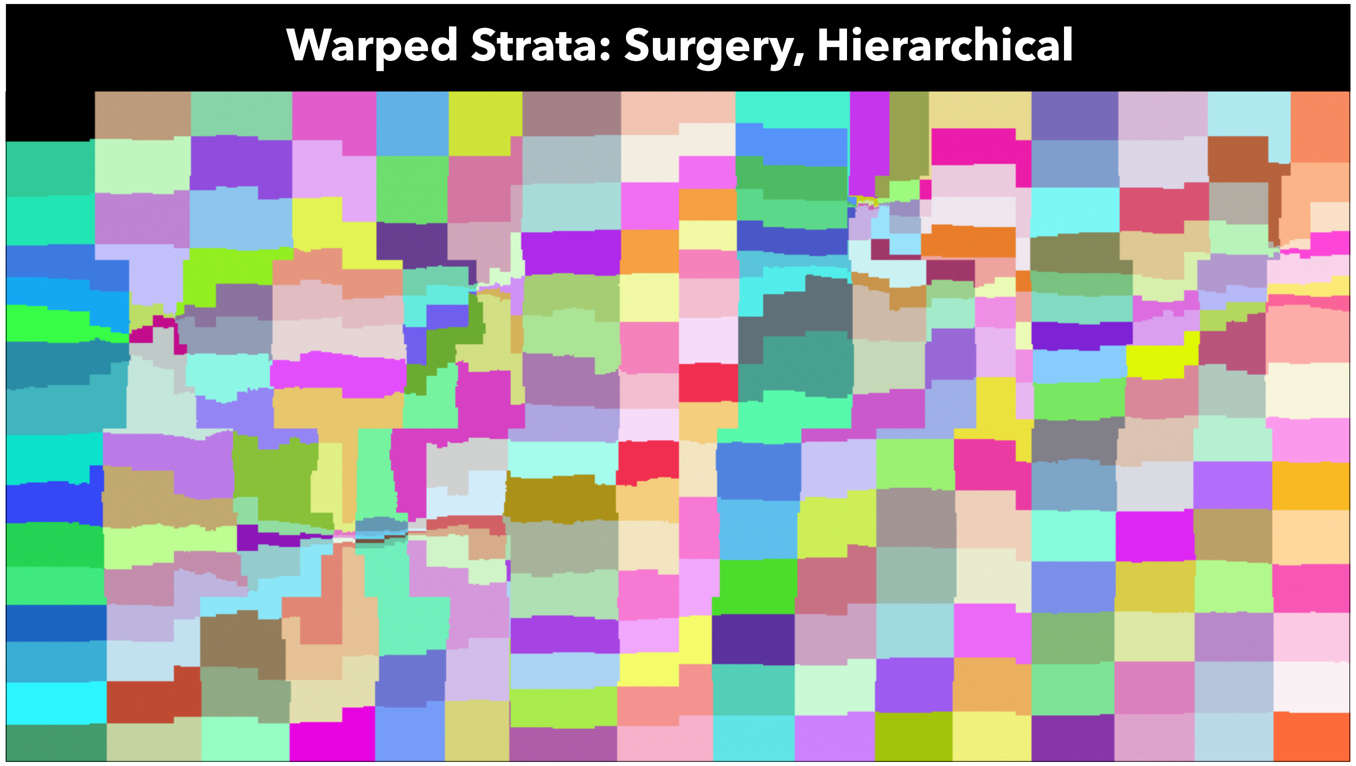
Given what we saw in those those visualizations, I would have expected much better results from the hierarchical warping than the marginal-conditional warping. In practice, it's better, but only about 10% better. A little bit disappointing…

HDRI Haven: Surgery

Let's see what happens with the surgery environment map.



Here are the marginal-conditional strata. There are multiple bright spots in the image, and again we can see the strata being pulled toward them—lots of distortion and that wavy thing going on in the center scanlines.



Here’s the hierarchical warp with the surgery environment map. Again, most strata look pretty good, but things sometimes get weird by the light sources. If you look long enough, you can see some strata that have been sheared into multiple pieces—for example, the orange one in the middle, four down from the top of the image. That’s definitely not a good thing, either.

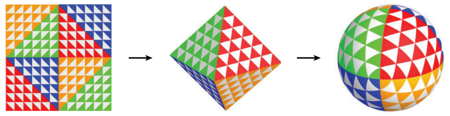
Results: Surgery
16spp, MSE x 1000

	Marg/Cond	Hierarchical
Stratified	2.16	2.03
pmj02bn	1.86	1.63

And again, we see that the hierarchical warping is about 10% better. Worth taking, but not super compelling. The benefit from using well-distributed sample points is much bigger, for example.

Spherical Mappings

- Equi-recangular:
  - Flattens sphere using spherical coordinates
  - Preserves neither area nor angles
- Equi-area:
  - Preserves fractional area
  - e.g. *Fast Equal-Area Mapping of the (Hemi)Sphere using SIMD*, Clarberg 2008



Thus far, we've been looking at environment maps represented using an equi-rectangular mapping from the sphere to a 2D image. (Sometimes known as a "lat-long" map.) It's just computing the spherical coordinates of the direction and scaling those back to [0,1)^2. You'll note that it exhibits quite a bit of distortion up by the poles.

There are lots of other mappings from the sphere to the unit square. One nice one was introduced by Clarberg, combining an octahedral mapping of the sphere to the square and then an equi-area mapping within each quadrant.

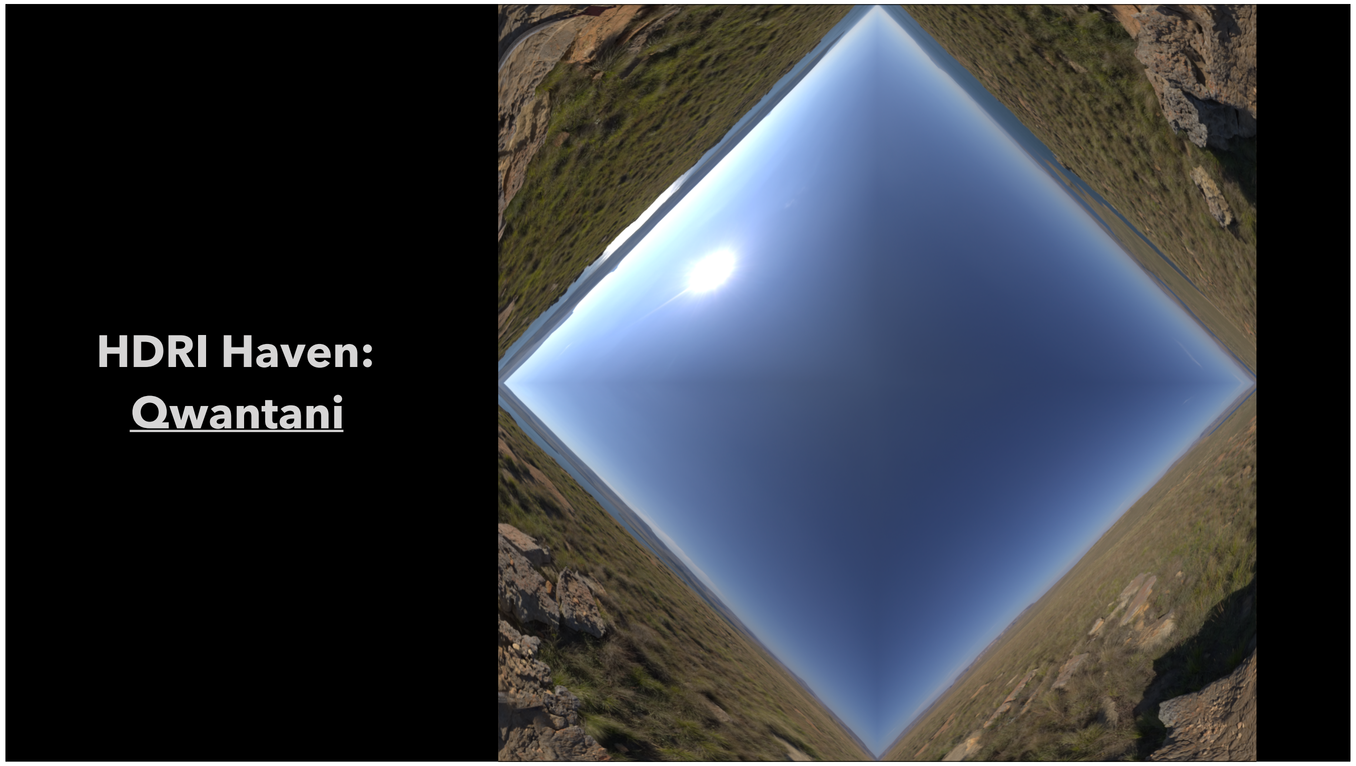
The mapping is shown here—the upper hemisphere corresponds to that rotated square in the middle and the lower hemisphere corresponds to the four triangles around it. You can think of those all folding down to meet underneath.

HDRI Haven: Surgery

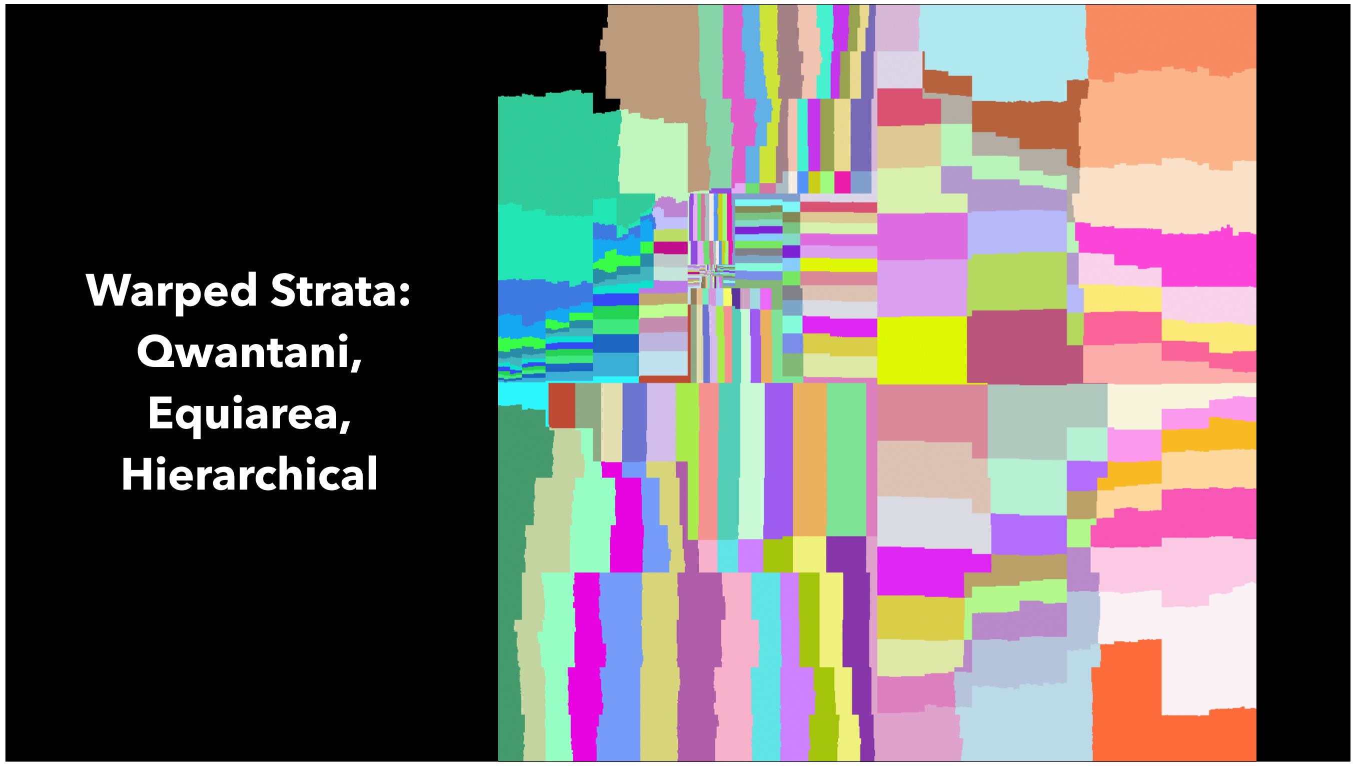
To get a sense of how that mapping works, we'll look at the surgery environment map represented using it. Here it is with the equi-rect mapping.

And here it is with Clarberg's equi-area mapping. So again, you can see that "up" corresponds to that rotated square in the middle.

And here are the strata with the hierarchical warping and the equi-area mapping. There are still some weird ones and still a few that are broken into pieces, but it's pretty well behaved.



And here's the outdoor scene with the equi-area mapping.



And here are the strata with the hierarchical warping. Again, there are some pretty distorted ones and a few broken into pieces, but this is a really tough environment map—with so much dynamic range in a small region of it, one way or another there's got to be a lot of warping to get the samples distributed correctly.

Results: Qwantani, Hierarchial Warp 16spp, MSE x 1000

	Equi-rect	Equi-area
Stratified	1.16	0.90
pmj02bn	0.80	0.55

Let’s look at error, here just considering the hierarchical warp, since it was the winner earlier.

This turns out to be pretty interesting! The parameterization of the environment map turns out to matter much more than the choice of warping algorithm did—error is reduced by 20 to 30 percent by using a parameterization that has less distortion.

Results: Surgery, Hierarchial Warp
16spp, MSE x 1000

	Equi-rect	Equi-area
Stratified	2.03	1.98
pmj02bn	1.63	1.12

We see similar results with pmj02bn points with the surgery environment map, but not much of a win with stratified. It's interesting that there's a difference in that respect compared to the outdoor environment map; I haven't chased down what's going on there.

Recap

- General stratification from low-discrepancy point sets/ sequences provides robustness for non-uniform warpings
- Warpings that pull in different directions are best avoided if possible
- Both how you warp and what domain you choose to warp to can make a big difference

Hopefully you've seen that there are all sorts of interesting things that happen when you take well-chosen points in the canonical domain and use them for Monte Carlo integration.

We saw that sample points that fulfill the general stratification properties of (0,2)-sequences are more robust to non-uniform stretching than regular old stratified points are; that's one of the reasons they work so well.

In the case of the disk mappings, we saw that mappings that pull samples in different directions seem to be a bad idea—intuitively, that seems like the sort of thing that's going to destroy good stratification properties. We saw that adjusting the warping function to account for this, as in the polar4 mapping, can give a massive reduction in error—over two times in that case.

And then finally, with illumination from environment maps, we saw a case where the warping algorithm mattered less than the domain we were warping to: if the target domain itself is distorted, then that in turn can increase integration error.

One way to think about all this is in terms of integrating over the [0,1)^n domain but then distorting the integrand according to the inverse of the warping function. Considering the effect of the warping on the properties of the integrand would likely be a good avenue to more rigorously express some of what we've seen empirically today.

Thanks!