Sampling and Intro to Monte Carlo Estimation

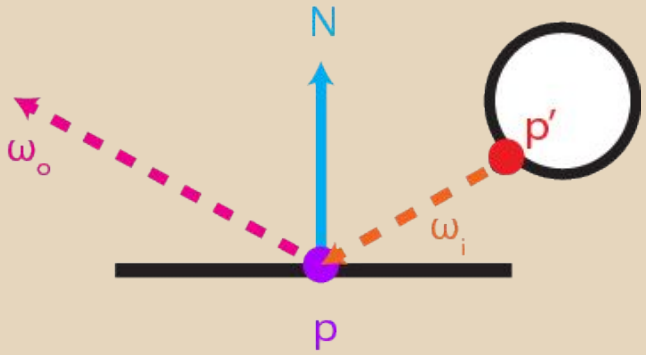
Adam Mally
CIS 561 Spring 2017
University of Pennsylvania

Monte Carlo Path Tracer vs Ray Tracer

- MCPT attempts to solve the lighting equation for all visible points in the scene
- Cast many, many rays per pixel in order to gather enough lighting information
- Effects like caustics, soft shadows, anti-aliasing, and depth of field are essentially free
- Without parallelization and other optimizations, converges to a useful image incredibly slowly

- RT casts one ray per pixel (excluding anti-aliasing)
- Makes physically incorrect assumptions about light transportation in order to ensure a fast run time
- Must add special computations to support/fake effects like caustics and depth of field
- Produces a usable image in a relatively short time, even if the image is not physically accurate

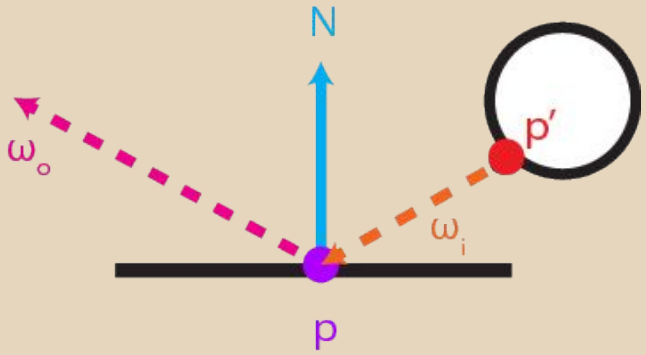
Light Transport Equation - Integral



$$L_o(p,\omega_o) = L_E(p,\omega_o) + \int_S f(p,\omega_o,\omega_i) L_i(p,\omega_i)\, V(p',p)\, \text{absdot}(\omega_i, N)\, d\omega_i$$

- L_o: radiance from an intersection p and direction ω_o.
- L_E: emitted radiance from a point (nonzero only if p is a light source; all other light is *reflected*)
- f: BRDF of surface at p (we'll discuss BRDFs shortly)
- L: radiance at some point x in direction ω_i.
- V: visibility between p' and p (1 when unobstructed, 0 otherwise)
- We incorporate the dot product of ω_i and N due to Lambert's law
- S is the sphere of all possible ω_i that can reach p

Light Transport Equation - Sum

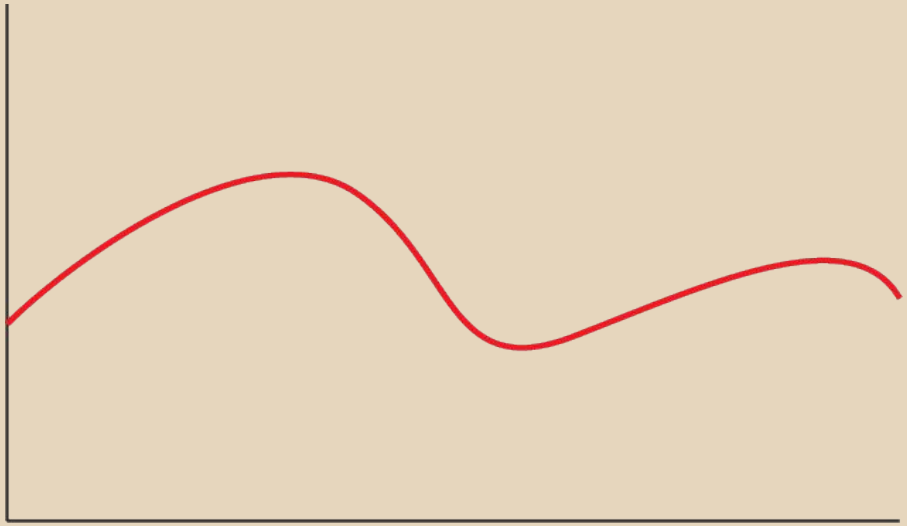


$$L_o(p,\omega_o) = L_E(p,\omega_o) + \Sigma_\omega (f(p,\omega_o,\omega_i)L_i(p,\omega_i)\ V(p',p)\ \text{absdot}(\omega_i, N)\ d\omega_i)$$

- L_o: radiance from a intersection p and direction ω_o.
- L_E: emitted radiance from a point
- f: BRDF of surface at p (we'll discuss BRDFs shortly)
- L: radiance at some point x in direction ω_i.
- V: visibility between p' and p (1 when unobstructed, 0 otherwise)
- We incorporate the dot product of ω_i and N due to Lambert's law
- For some discrete subset of S, choose n ω to sample and average their reflected light energies
  - The more samples taken, the closer the average is to being physically

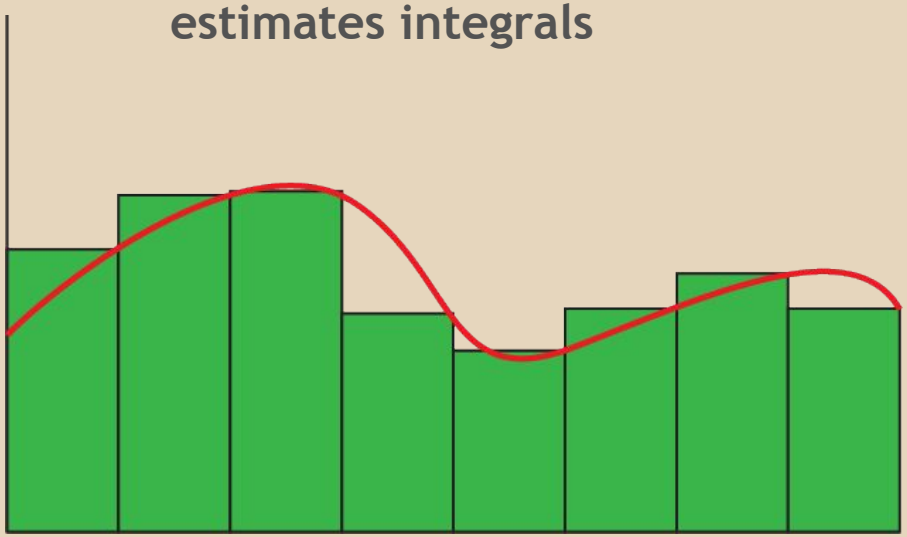
Estimating the value of an integral

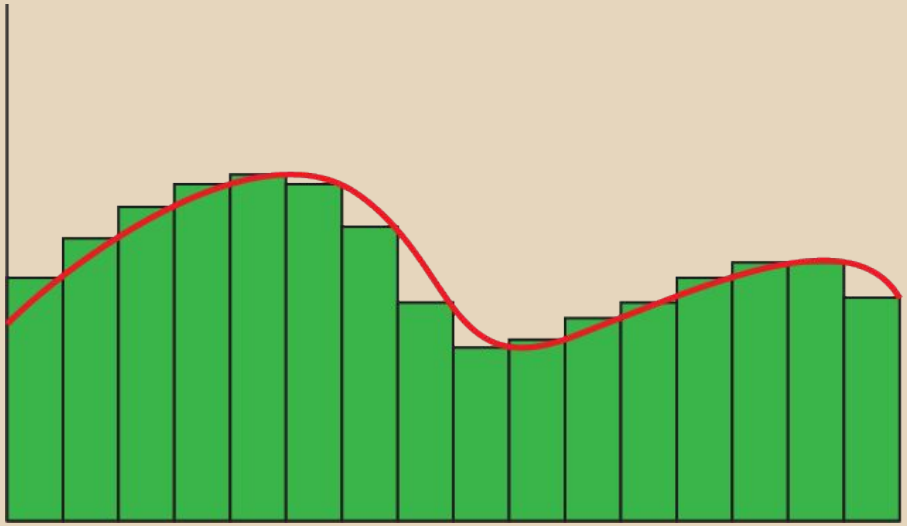
- As we saw in the previous slide, we do not directly compute the value of the light transport integral
  - Too complex to do so
- We sample the function at several points and average the results to estimate the integral's value
  - The example below is on a 2D function for the sake of illustration, but the light transport integral is far more complex



Estimating the value of an integral

- If we sample the function more frequently, our estimation is more accurate
- How do we accurately estimate the integral without having to take millions of samples?
  - Distribute our samples intelligently
  - We'll discuss this more another day
- Remember: these images are meant to help you visualize integral estimation, **but they are not true to how a Monte Carlo path tracer estimates integrals**





Estimating the value of a high-dimensional integral

- Estimation approaches like the midpoint method or trapezoidal integration are fine for low-dimensional functions
  - Their rate of convergence (number of computations needed for accuracy) is far too slow to be used for the light transport equation
- We use **Monte Carlo Estimation** to compute the expected value of the light transport equation integral
  - Monte Carlo takes **random** samples of a function and averages the results together
  - Gets different results depending on the samples taken
  - With enough samples, the average result is the correct integral value

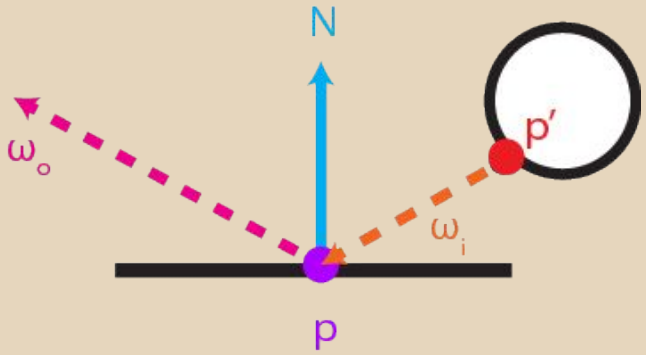
MC Path Tracer Convergence

As you take more samples, the image converges to the correct integral for each pixel

Random variables

- As was previously stated, we choose random samples from the domain of a function in order to estimate the value of its integrals
- We want to use **continuous random variables** for this task
  - CRVs take on values over continuous domains, e.g. all real numbers or all possible directions within a unit sphere
- A very important CRV is the **canonical uniform random variable**
  - Commonly denoted as ξ (Xi, pronounced “ksigh” or “ksaɪ“)
  - Spans the domain [0, 1) with all values being equally probable
  - Relatively simple to sample from arbitrary distributions by transforming this CRV
- How can we use ξ in a Monte Carlo path tracer?

Light Transport Equation

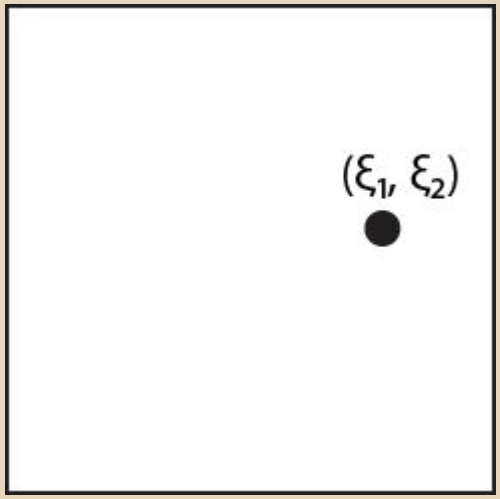


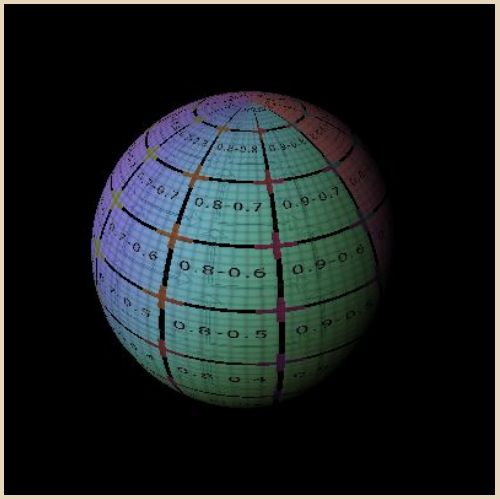
$$L_o(p,\omega_o) = L_E(p,\omega_o) + \int_S f(p,\omega_o,\omega_i) L_i(p,\omega_i)\, V(p',p)\, \text{absdot}(\omega_i, N)\, d\omega_i$$

- S is the sphere of all possible ω_i that can reach p
  - S is the domain of the LTE
  - We want to sample from this domain to estimate the integral
- Generally, we really only care about one half of S, depending on whether or not we are dealing with a reflective or transmissive material
- We want to use ξ to generate random ray directions within a hemisphere $\mathcal{H}$

Sampling Shapes

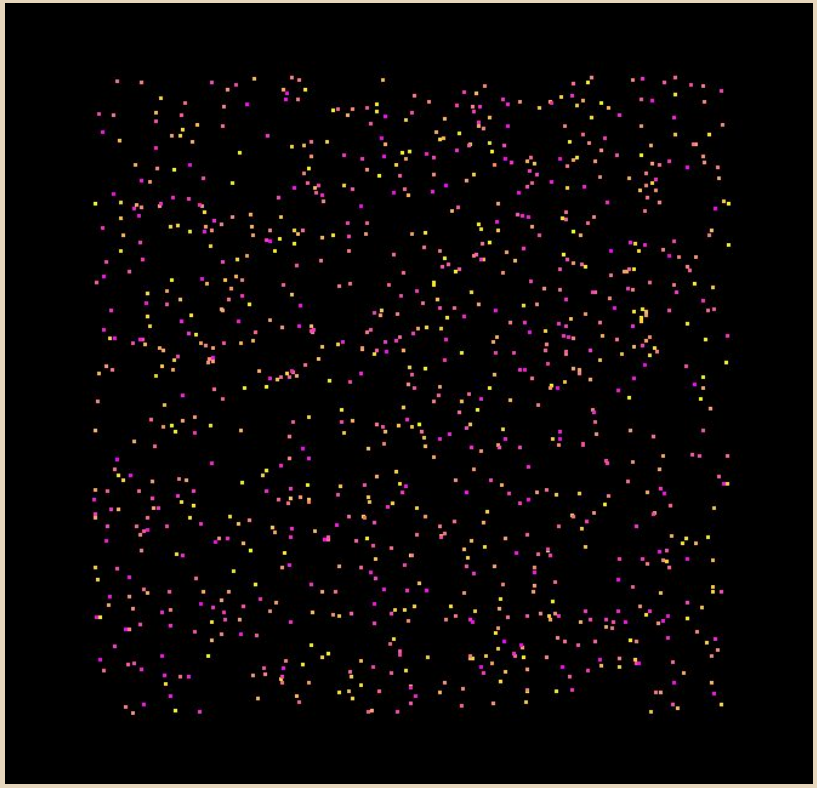
- It's rather simple to generate random samples on a 2D square
  - Just create a pair of uniform random variables ξ_1 and ξ_2 to make up the X and Y (or U and V, if you like) coordinates of a square
- Most 3D surfaces can be mapped to 2D planes
  - e.g. polar coordinates to map a sphere to a square

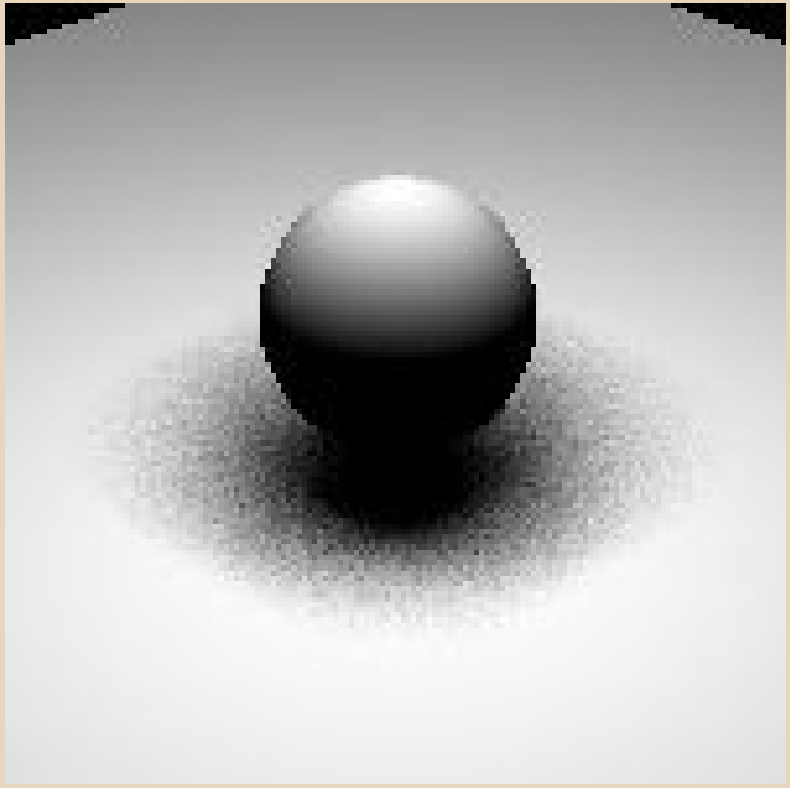




Distribution of Samples: Pure random

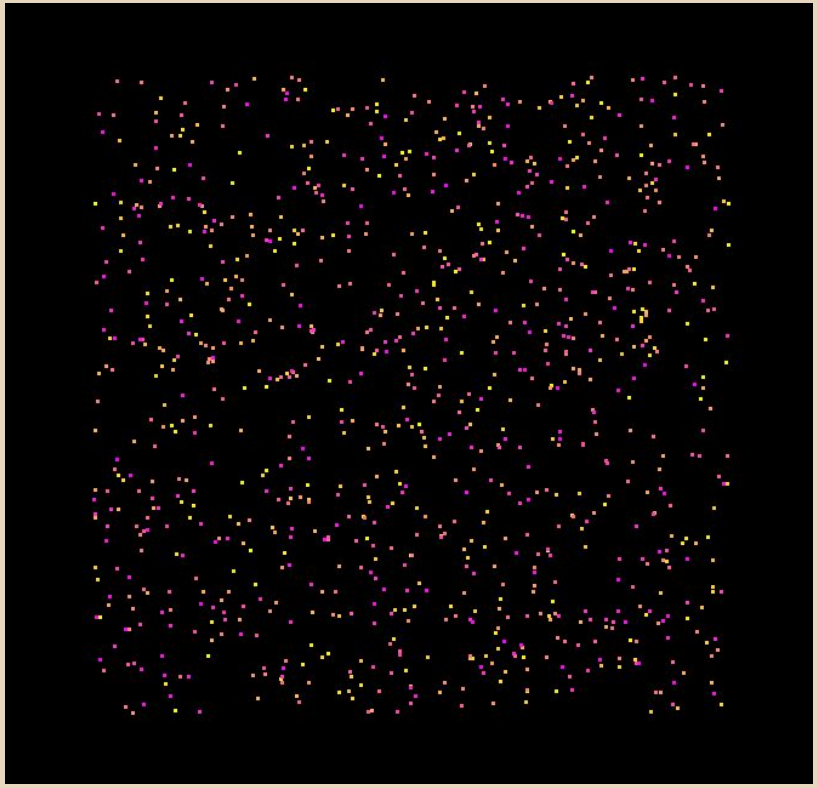
- Entirely random samples are the most “mathematically pure”, but they tend to cause poor convergence rates
- Let’s look at the example of soft shadows from area lights:

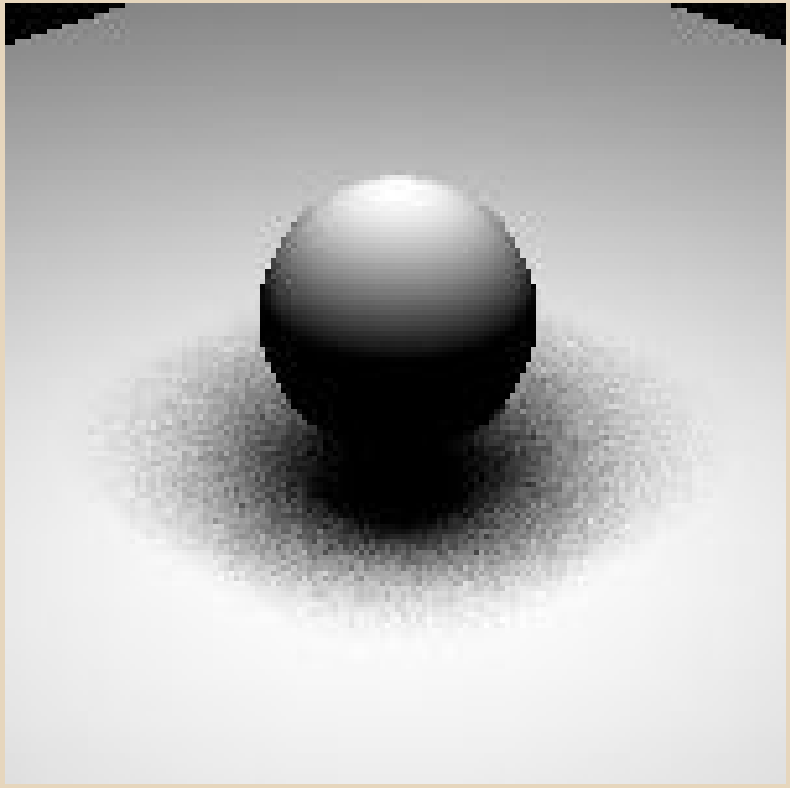




Distribution of Samples: Pure random

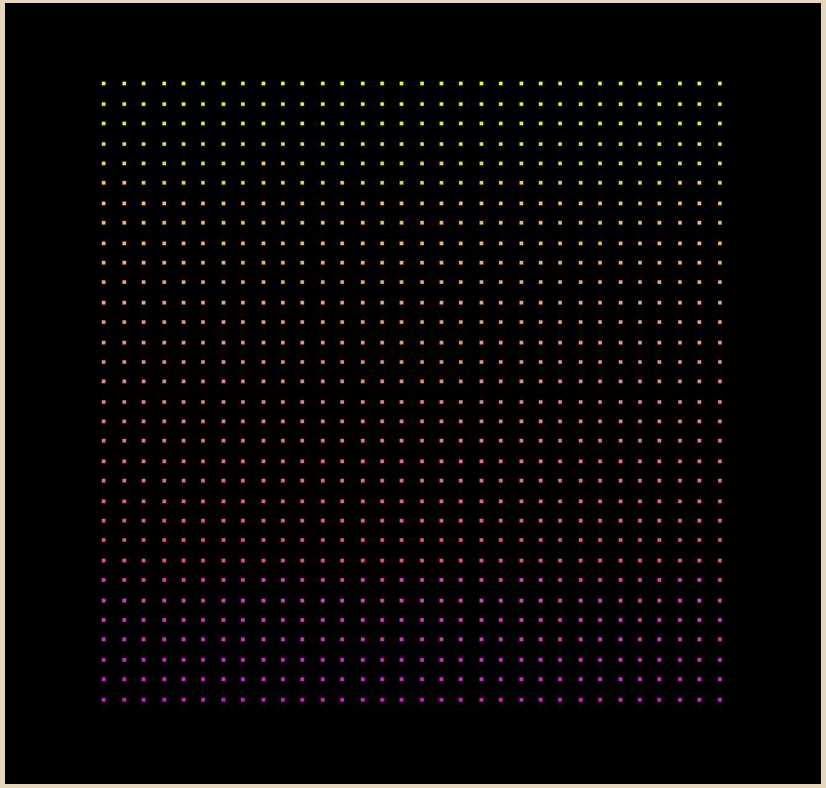
- The shadow is quite grainy and needs many more samples per pixel to converge to a smooth shadow
- No guarantee of uniform distribution with purely random

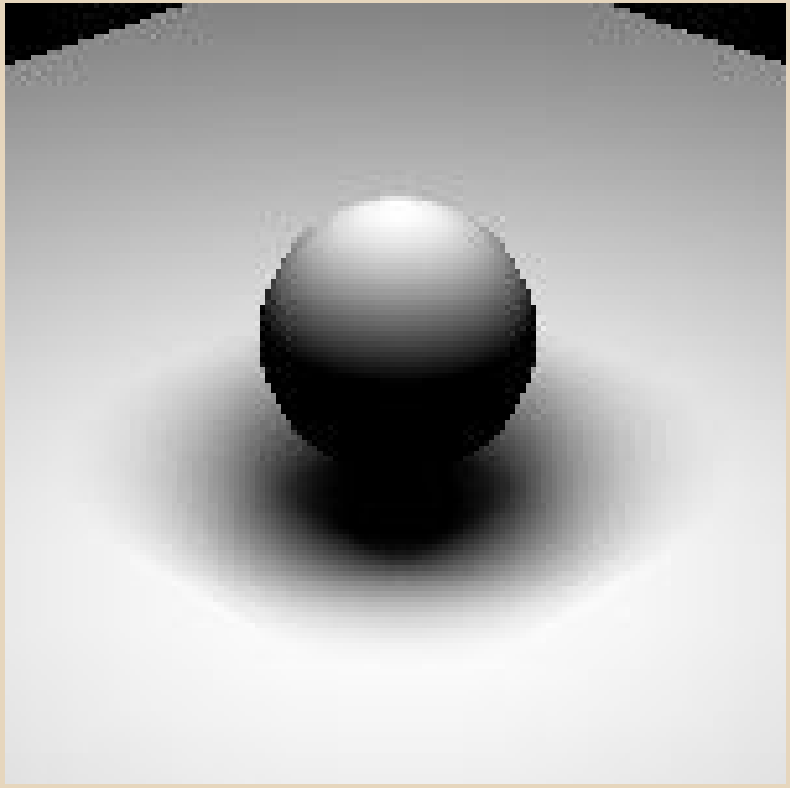




Distribution of Samples: Grid

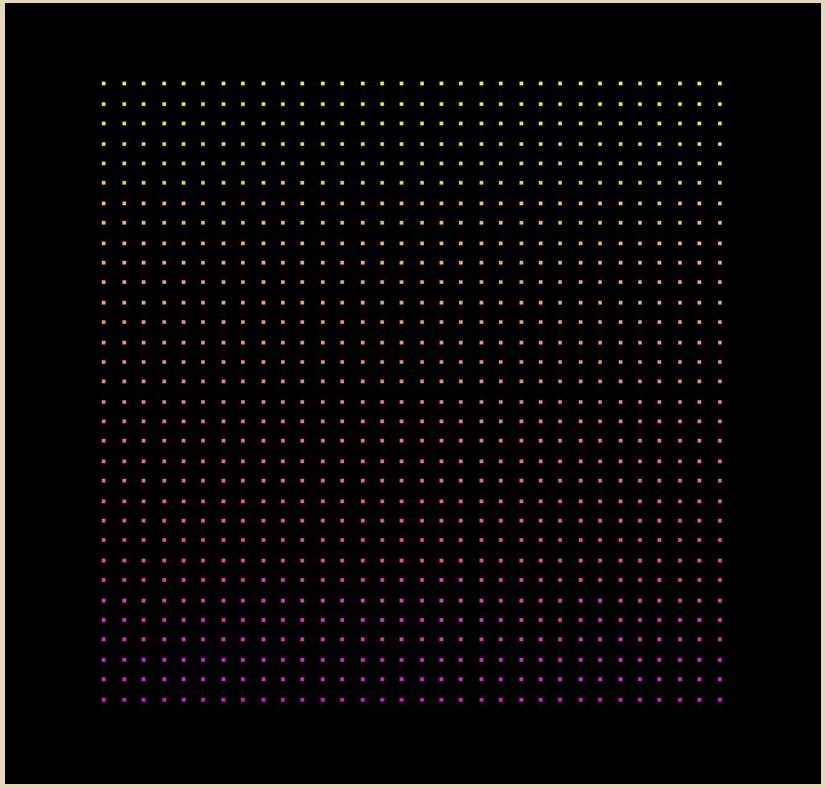
- What if we use a grid-based distribution instead?
  - Place a sample at the center of each grid cell
  - Guaranteed uniform distribution

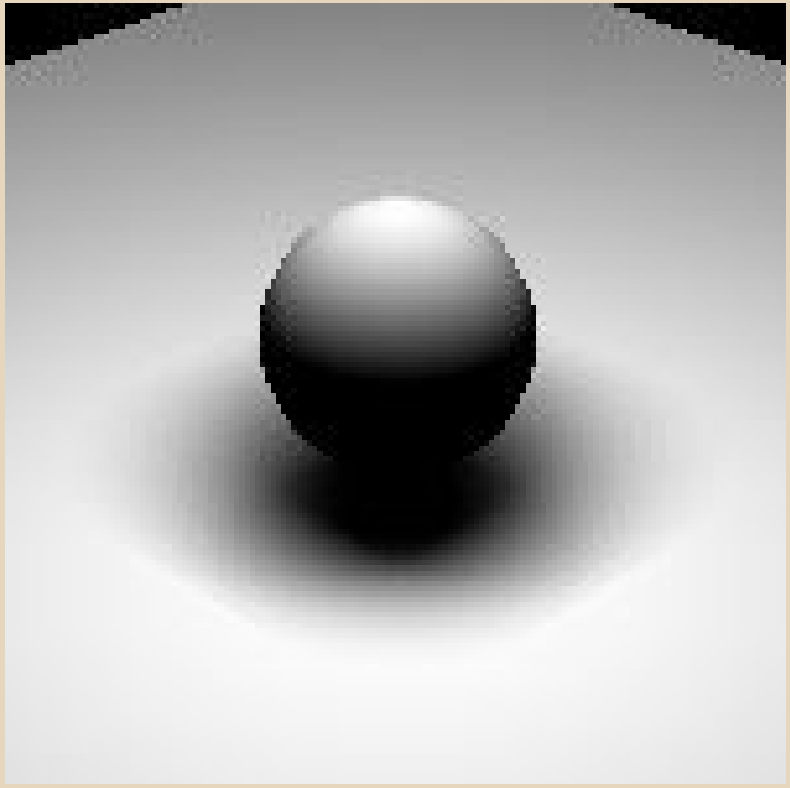




Distribution of Samples: Grid

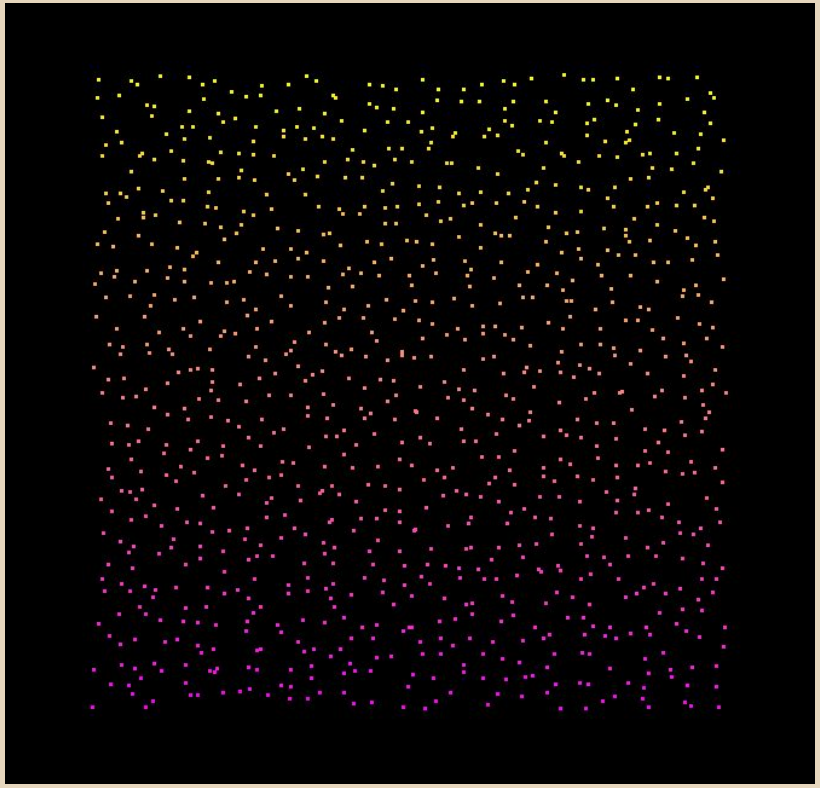
- Better shadow, but now there are noticeable stratifications of shadow color
- Grid-based sampling is *too* uniform to look natural

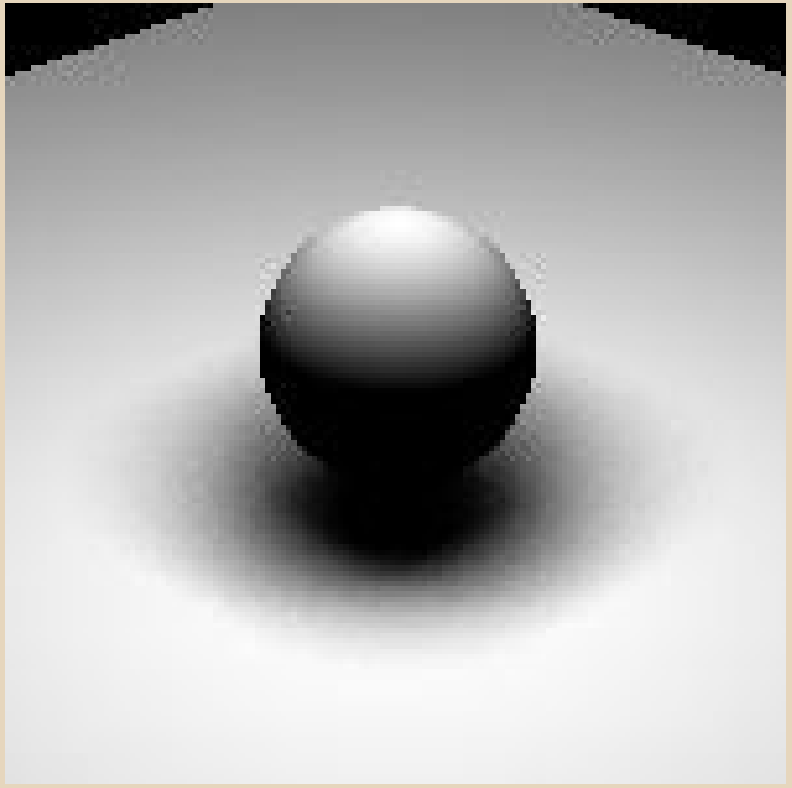




Distribution of Samples: Stratified

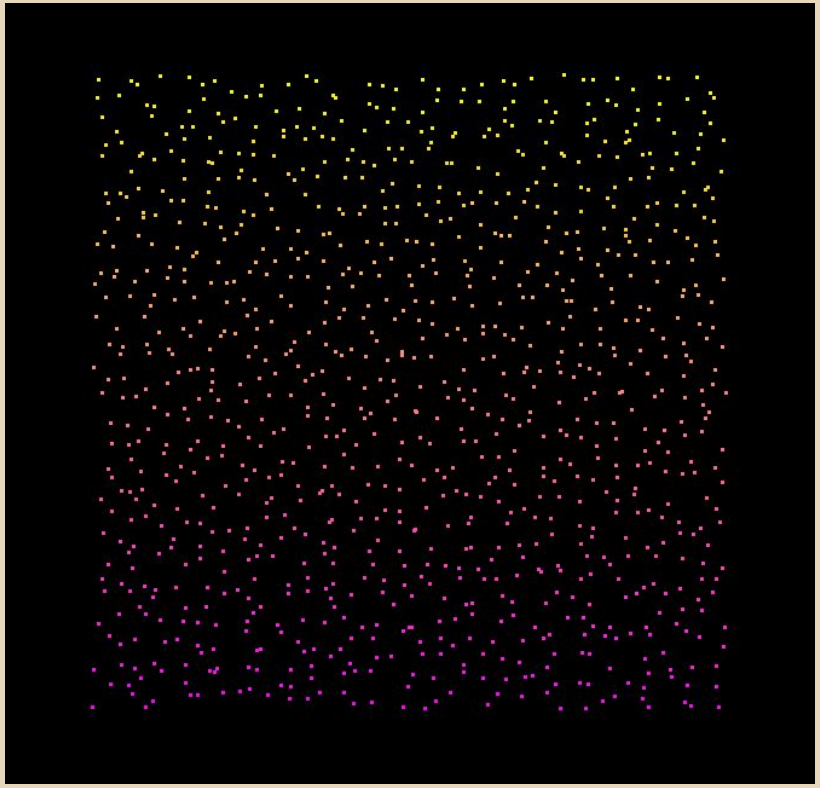
- What if we combine the grid and random approaches?
- Divide the square into cells, and place one sample in a random position within each cell

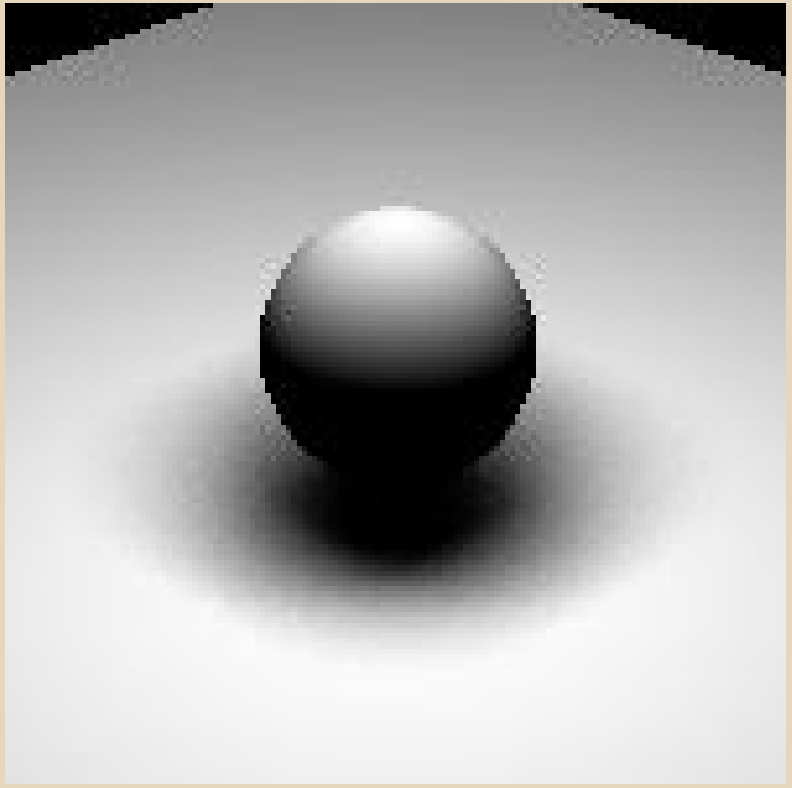




Distribution of Samples: Stratified

- The shadow is much more uniform in appearance than the purely random one, and it exhibits no obvious artifacts





Warping 2D Samples

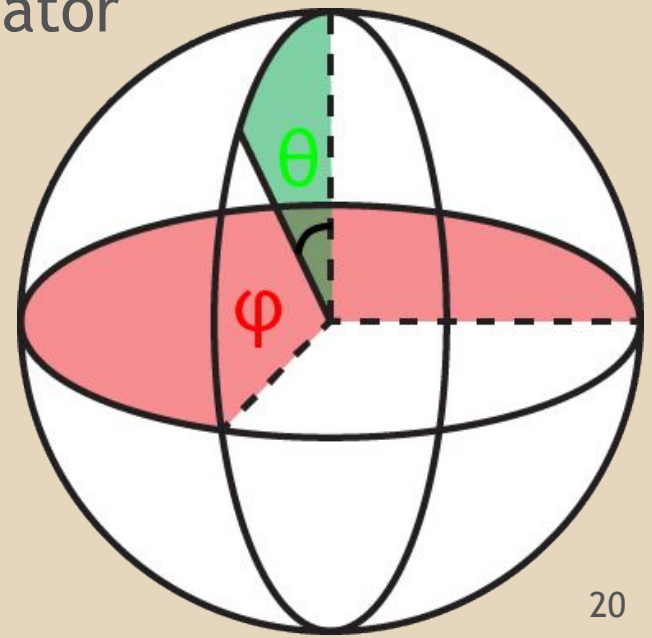
- We can create a good distribution of samples on a square, hooray!
  - So what?

Warping 2D Samples

- We can create a good distribution of samples on a square, hooray!
  - So what?
- We want to generate samples on 3D surfaces by warping our 2D samples
- We want to map these 2D points onto the surfaces of various 3D shapes **while maintaining the relative spacing of the 2D points**
- What can we do?

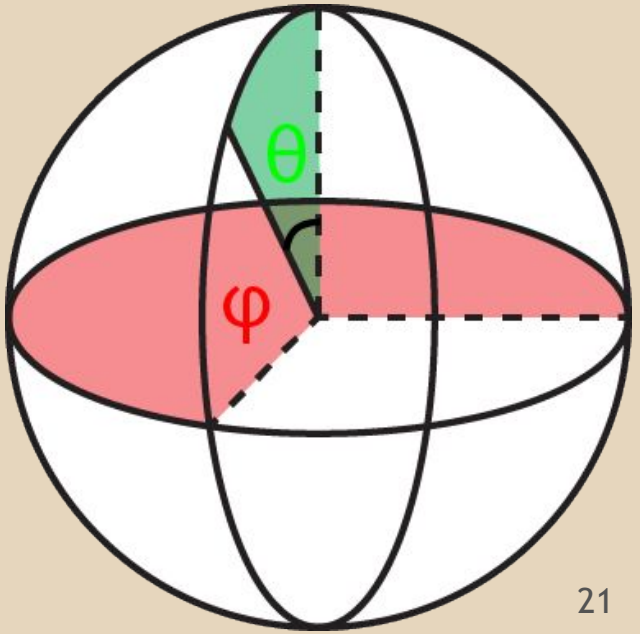
Square to Sphere

- We'll use polar coordinates (Θ, ϕ) to map (ξ_1, ξ_2) to the surface of a sphere
  - $0 \leq \Theta \leq \pi/2, \quad 0 \leq \phi \leq 2\pi$
- First let's figure out how to map the X and Y coordinates of our sphere
  - If we were to just handle the equator of the sphere, we'd get:
    $x = \cos(\phi), \quad y = \sin(\phi)$
  - We have to account for all "layers" of the sphere, so we incorporate Θ as well:
    $x = \sin(\Theta)\cos(\phi), \quad y = \sin(\Theta)\sin(\phi)$



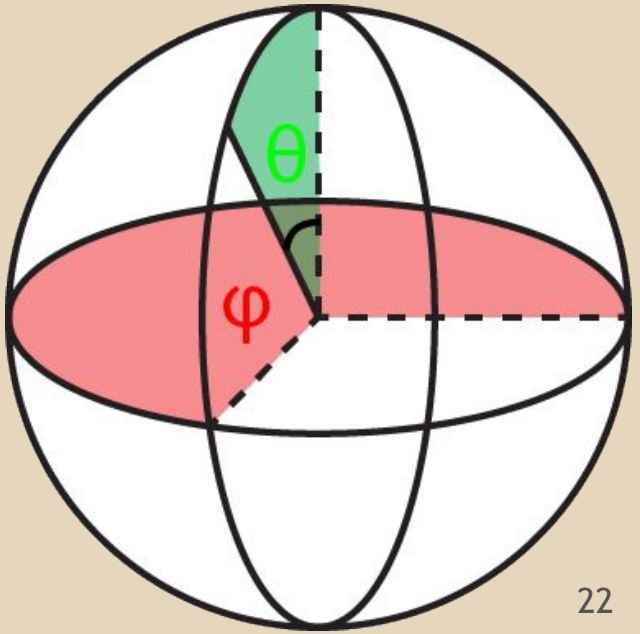
Square to Sphere

- So far, $x = \sin(\theta)\cos(\phi)$, $y = \sin(\theta)\sin(\phi)$
- What if we define z first? (Z is the vertical component of the sphere pictured)
- For a hemisphere, $z = \cos(\theta)$
- We need to account for both halves, so we get
  $z = 1 - 2\cos(\theta)$
- Let's bring in the ξs



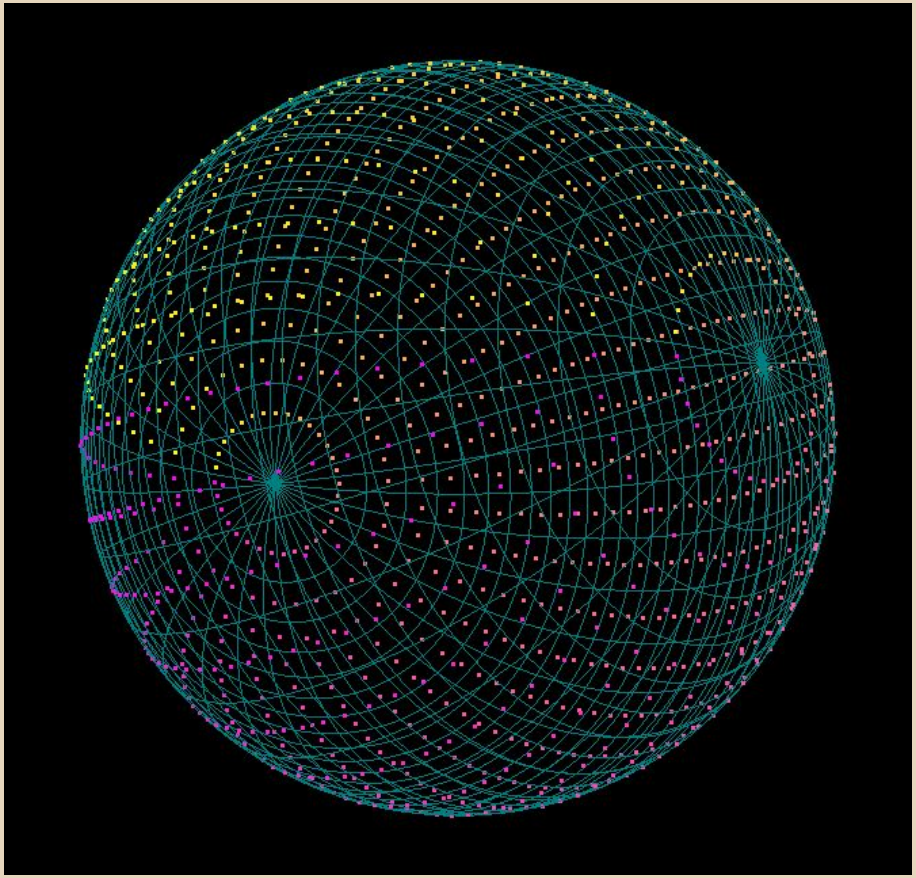
Square to Sphere

- $x = \sin(\theta)\cos(\phi)$
- $y = \sin(\theta)\sin(\phi)$
- $z = 1 - 2\cos(\theta)$
- Let's take $\phi = 2\pi\xi_2$ and $\theta = \cos^{-1}(\xi_1)$
  - There is a full proof of why we choose these values in the textbook
- We can rewrite our functions to be:
- $x = \cos(2\pi\xi_2)\sqrt{(1 - z^2)}$
- $y = \sin(2\pi\xi_2)\sqrt{(1 - z^2)}$
- $z = 1 - 2\xi_1$



Square to Sphere: Result

- Here’s a visualization of 1024 points warped to a sphere:

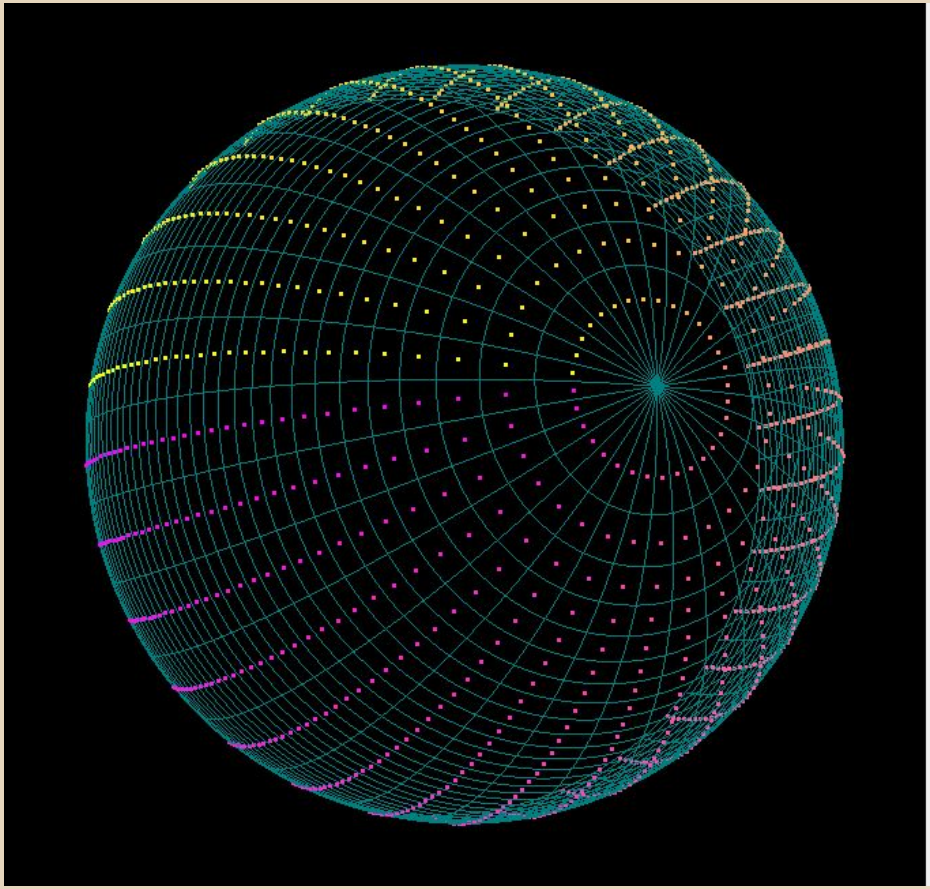


Square to Hemisphere (Uniform)

- Almost the same as mapping a square to a sphere
- $x = \sin(\theta)\cos(\phi)$
- $y = \sin(\theta)\sin(\phi)$
- $z = \cos(\theta)$
- Using $\phi = 2\pi\xi_2$ and $\theta = \cos^{-1}(\xi_1)$, we get
- $x = \cos(2\pi\xi_2)\sqrt{(1 - z^2)}$
- $y = \sin(2\pi\xi_2)\sqrt{(1 - z^2)}$
- $z = \xi_1$

Square to Hemisphere (Uniform)

- Here’s a visualization of 1024 points uniformly warped to a hemisphere:



Square to Hemisphere (Weighted)

- Let's look at the light transport equation again:

$$L_o(p,\omega_o) = L_E(p,\omega_o) + \Sigma_\omega (f(p,\omega_o,\omega_i)L_i(p,\omega_i)\ V(p',p)\ \text{absdot}(\omega_i, N)\ d\omega_i)$$

- Each element in the summation is multiplied by the dot product between the surface normal and incoming ray
  - What is the dot product of two vectors proportional to?

Square to Hemisphere (Weighted)

- Let's look at the light transport equation again:

$$L_o(p,\omega_o) = L_E(p,\omega_o) + \Sigma_\omega (f(p,\omega_o,\omega_i)L_i(p,\omega_i)\ V(p',p)\ \text{absdot}(\omega_i, N)\ d\omega_i)$$

- Each element in the summation is multiplied by the dot product between the surface normal and incoming ray
  - What is the dot product of two vectors proportional to?
  - The cosine of the angle between them!

Square to Hemisphere (Weighted)

- Let's look at the light transport equation again:

$$L_o(p,\omega_o) = L_E(p,\omega_o) + \Sigma_\omega (f(p,\omega_o,\omega_i)L_i(p,\omega_i)\ V(p',p)\ \text{absdot}(\omega_i, N)\ d\omega_i)$$

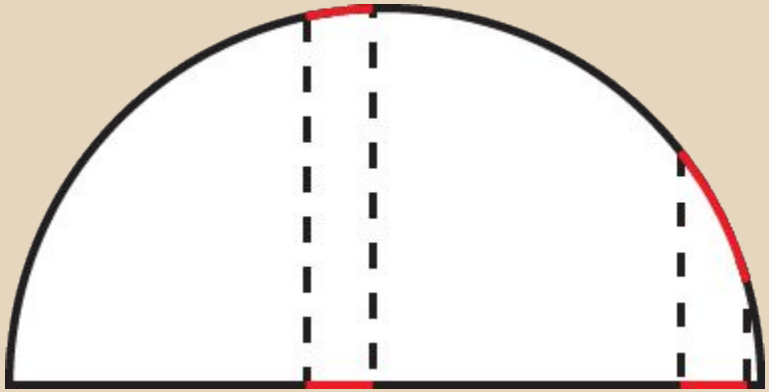
- We want to bias our samples toward the pole of the hemisphere so that each sample has a more meaningful contribution to the overall sum
- This is known as cosine-weighted sampling
  - The greater the cosine of the angle between sample ray and surface normal, the more likely that sample is to be chosen

Malley's Method

- Cosine-weighted hemisphere sampling is pretty simple to implement
- Just project a collection of samples uniformly distributed on a disk to the surface of a hemisphere
  - This is known as Malley's Method
  - It's not named after me
- If the disk samples are uniformly distributed, why do they become biased toward the hemisphere pole when projected?

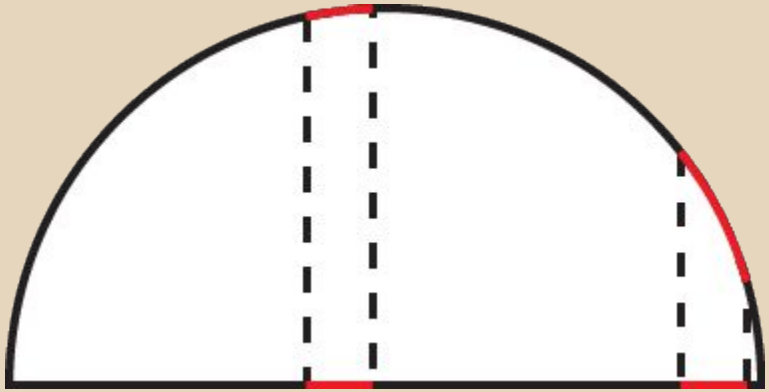
Malley's Method

- If we pick two areas of equal size on the disc and project them to the hemisphere, the area nearer to the edge becomes much larger
- Points that were close together on the disc will be spread significantly when projected to the edge of the hemisphere, while points will maintain approximate spacing near the pole
- This has the effect of clustering samples near the hemisphere pole
- How do we implement this?



Malley's Method

- If we assume the radius of the hemisphere to be 1.0, then we can extrapolate a Z coordinate given an X and a Y:
- $z = \sqrt{(1 - x^2 - y^2)}$
- This ensures all points have a distance of 1 from the center of the hemisphere's base

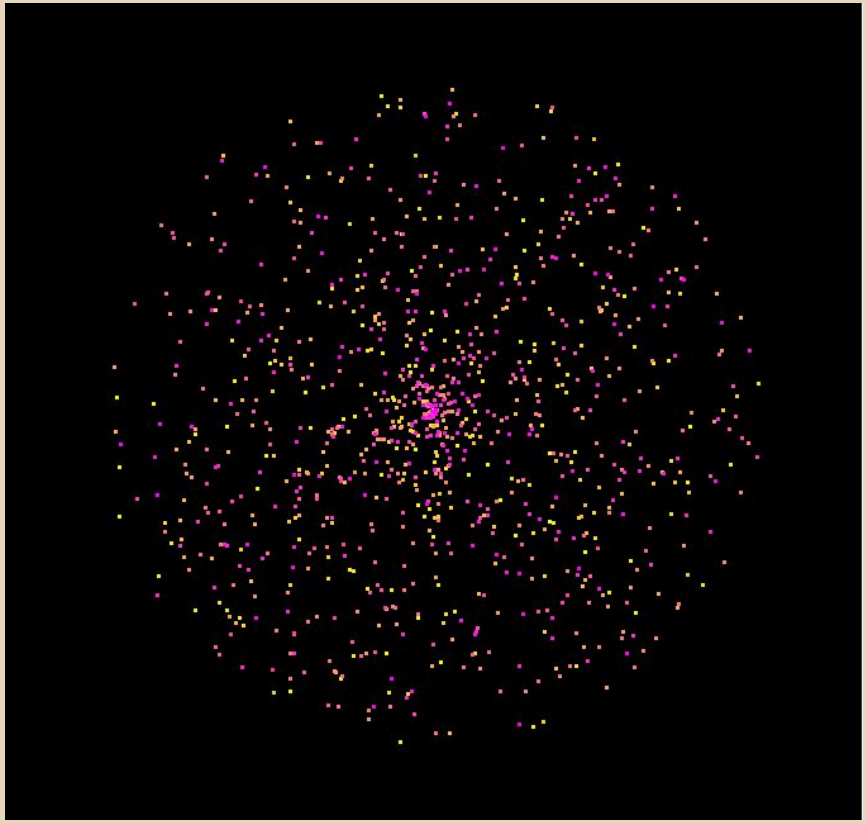


Square to Disc

- We need to uniformly sample a disc to perform Malley's Method
- Why not say radius $r = \xi_1$, $\theta = 2\pi\xi_2$?

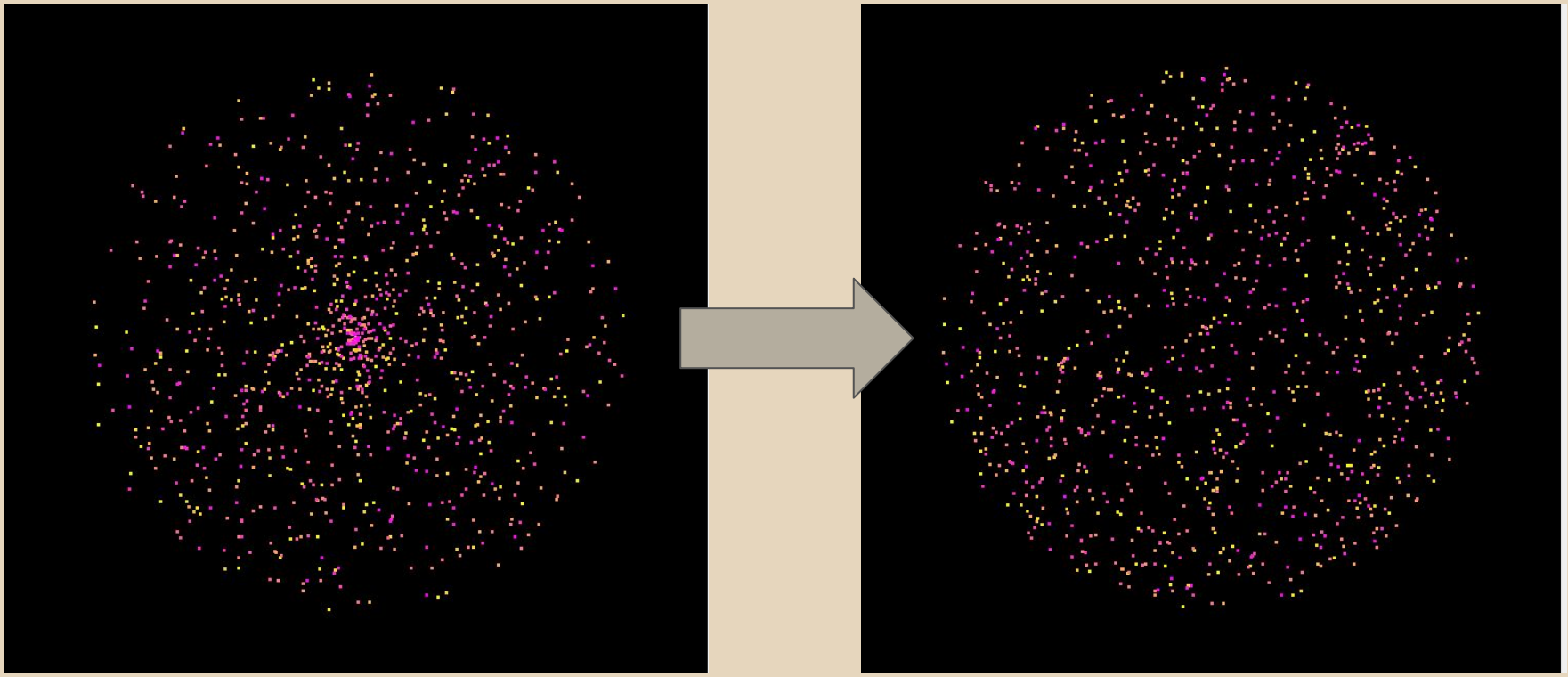
Square to Disc

- We need to uniformly sample a disc to perform Malley's Method
- Why not say radius $r = \xi_1$, $\theta = 2\pi\xi_2$?
- This ends up clustering the samples near the center of the disc
  - Arc lengths are smaller between angles at lower radii than at higher radii



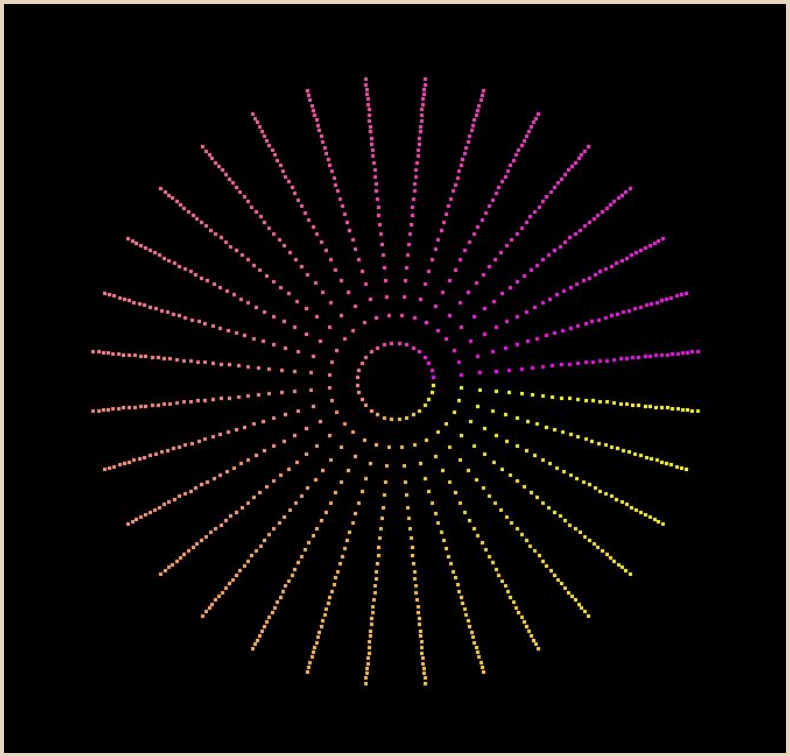
Square to Disc

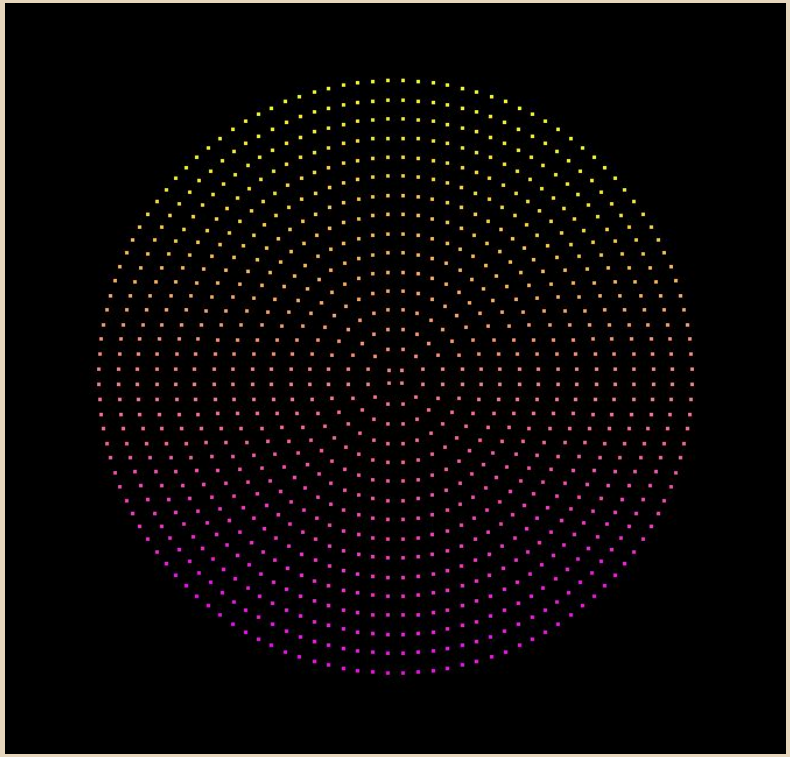
- Instead, radius $r = \sqrt{(\xi_1)}$, $\theta = 2\pi\xi_2$
- Now we bias the samples' radii toward the outer edges of the disc, counteracting the clustering



Square to Disc

- One slight issue with this method is that it doesn't quite preserve relative spacing of samples
- PBRT discusses a second method of warping a square to a disc that is better about this





What is a BRDF?

- A function that evaluates the energy emitted along ray ω_o given a point of intersection **p** and the direction from which the incoming light emits, ω_i
- Entirely dependent on the attributes of the material sampled at point p
- Examples:
  - A Lambertian BRDF simply takes the material's base color and divides it by π
    - Since we integrate over the entire *hemisphere* of visible directions from **p**, we need to divide by π so that the integral evaluates to the material's base color
  - A BRDF describing perfect specular reflection would be a discontinuous function that returns 0 (black) for all pairs of ω_o and ω_i that are not reflections of one another around the normal at **p**