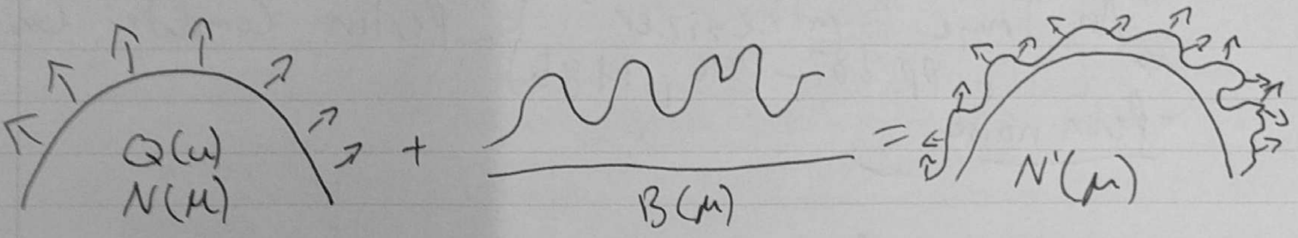
Bump Mapping

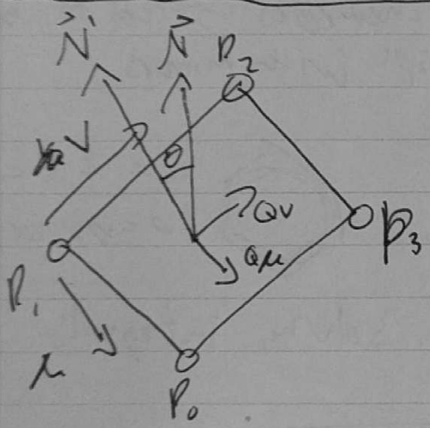
References

- J.F. Blinn "Simulation of Wrinkled Surfaces", Computer Graphics, vol 12, N13, 1978, pp 286-292
- Advanced Rendering Technologies by Watt & Watt 1990



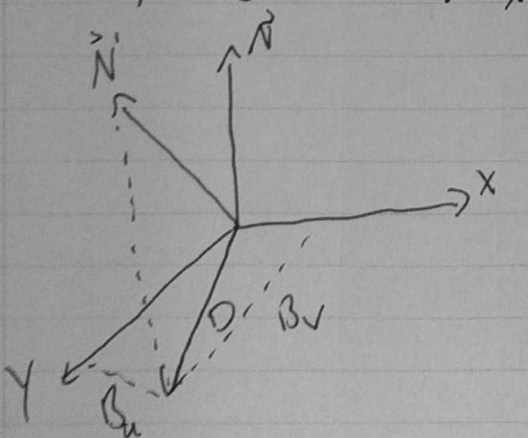
$N' = N + D$ - Perturb the normals

Geometric Representation of Bump Mapping

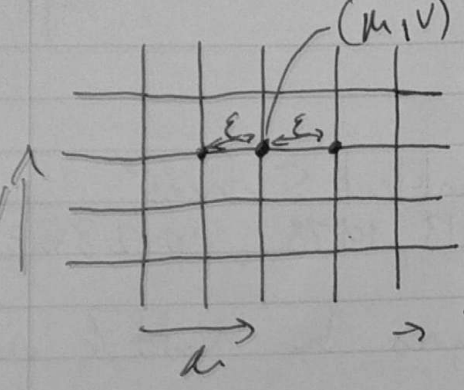


$u, v \in [0, 1]$
$N = Q_v \times Q_u$
Q_v and Q_u partial derivatives of the surface $Q(u,v)$ at the point (u, v)

$X = N \times Q_v$ $\quad Y = N \times Q_u$



D is given by $D = B_u X - B_v Y$ where B_u and B_v are partial derivatives of the Bump Map $B(u,v)$



$B(u,v)$ $\varepsilon = \frac{1}{64}$

$B_u = (B(u+\varepsilon, v) - B(u-\varepsilon, v)) / (2 * \varepsilon)$

$B_v = (B(u, v+\varepsilon) - B(u, v-\varepsilon)) / (2 * \varepsilon)$

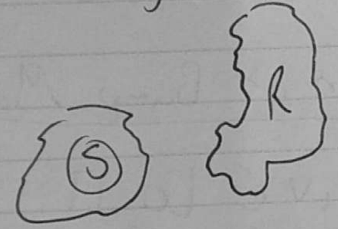
→ Estimation of the partial derivitive.

"An Image Synthesizer" K. Perlin Computer Graphics 19 (3), pp 287-296, 1985)

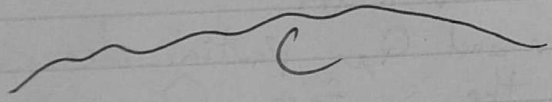
Perlin noise

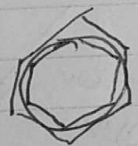
Digression: Painting with graphics phenomeron.

214 BC

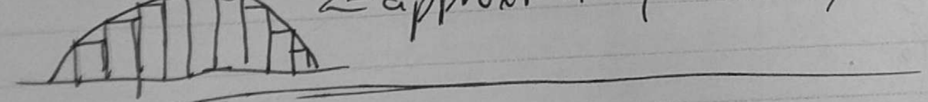


Romans were invading Archemedes tried to burn Ships with mirrors





use 2 octogons to approximate π



← approximation, solving integrals.

1985 J. Kajiya came up with the Rendering Eq.

Rendering Equation

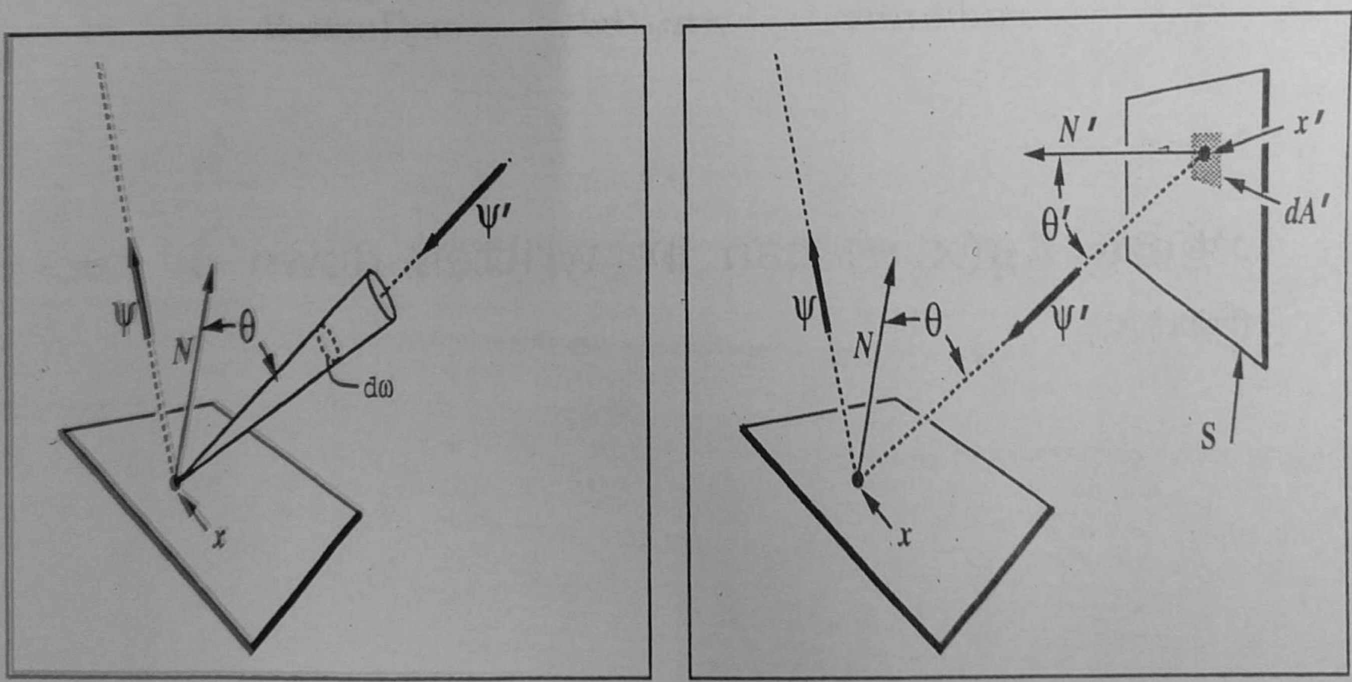
$$\underbrace{L(\mathsf{x},\psi)}_{radiance} = \underbrace{L_E(\mathsf{x},\psi)}_{emitted} + \underbrace{L_R(\mathsf{x},\psi)}_{reflected}$$

where $L_R(\mathsf{x},\psi)$ can be written down in two forms.

$$L_R(\mathsf{x},\psi) = \int_{\text{incoming } \psi'} f(\mathsf{x},\psi,\psi') L(\mathsf{x},\psi') \cos\theta d\omega'$$

or:

$$L_R(\mathsf{x},\psi) = \int_{\text{all } \mathsf{x}'} g(\mathsf{x},\mathsf{x}') f(\mathsf{x},\psi,\psi') L(\mathsf{x}',\psi') \cos\theta \frac{dA' \cos\theta'}{||\mathsf{x}'-\mathsf{x}||^2}$$



Distributed Ray Tracing

Cook-Porter-Carpenter (1984)

Apply distribution-based sampling to many parts of the ray-tracing algorithm
Rays can also be stochastically distributed in *object space* to simulate

Gloss/Translucency

- Perturb directions reflection/transmission, with distribution based on angle from ideal ray

Depth of field

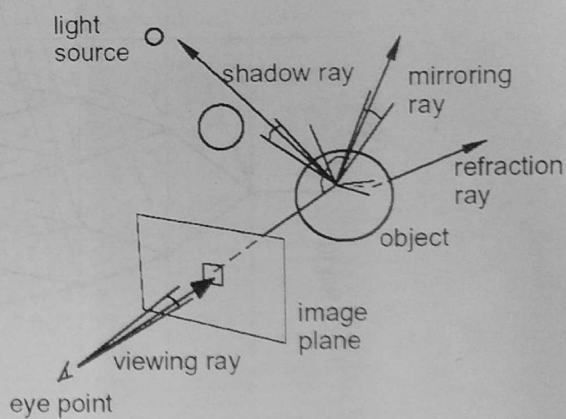
- Perturb eye position on lens

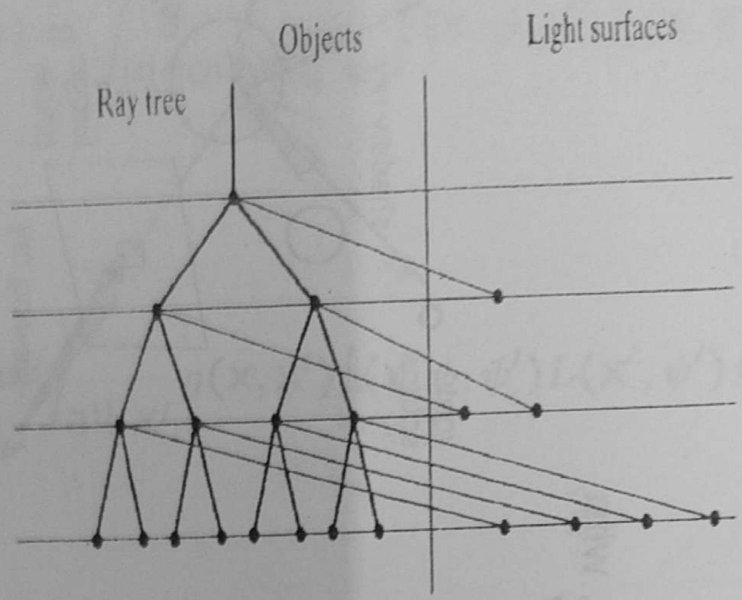
Soft shadow

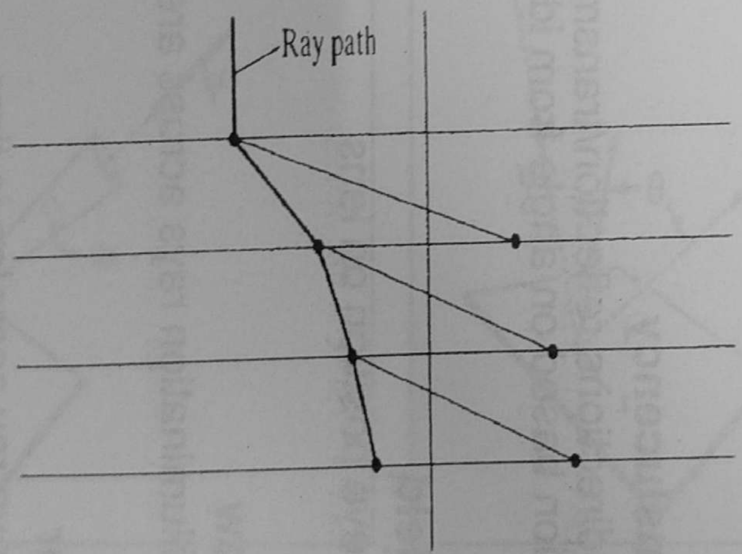
- Perturb illumination rays across area light

Motion blur

- Perturb eye ray samples in time

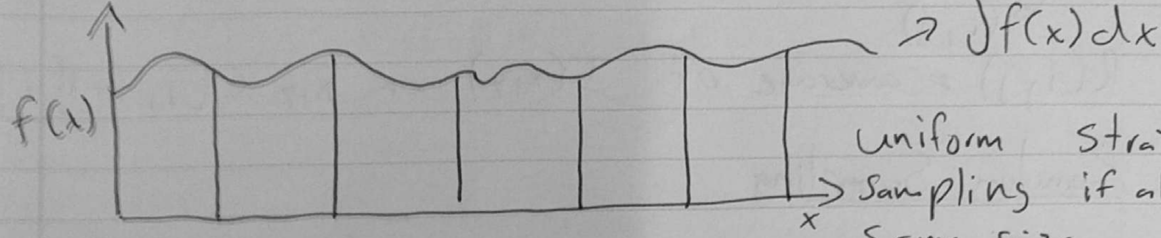



Objects
Light surfaces
Ray tree

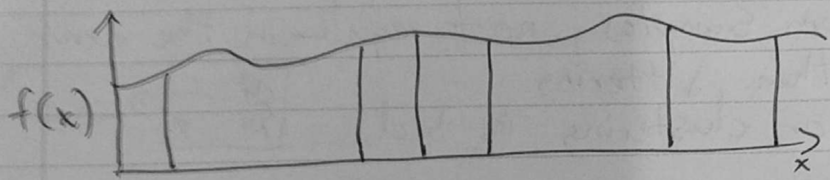

Ray path

Stratified Sampling

- break up the domain into disjoint regions (strata) and then place a sample into each region.



uniform stratified samplings if all the same size



non-uniform stratified samplings

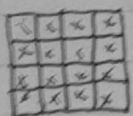
Importance Sampling

- take more samples where the "signal" is more important (larger fluctuans or larger wloes??)

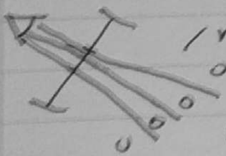
"Distributed Ray Tracing" Cook et all 1984

- 16 strata on the pixel (can use more, but more costly)
- does not deal with diffuse interactions

Regular Sampling



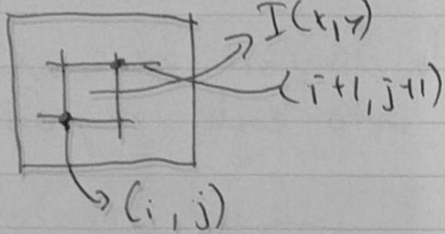
- easy to generate and fast - error is regular (aliasing)



/ regular error bad, would rather noise to average

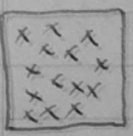
→

Regular Sampling

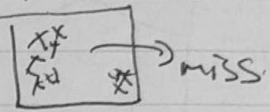


$R(i,j)$ = average of $I(x,y)$ on $x,y \in [i, i+1] \times [j, j+1]$

Random Sampling



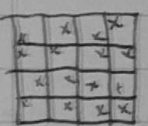
- 16 random samples, not regular in the error which is good
- faster than jittering
- clumping or clustering is bad



$$R(i,j) = \frac{1}{N} \sum_{0}^{N-1} I(i + z_1, j + z_2)$$

$z_{1,2}$ = random number uniformly distributed in $[0,1]$

Jittering



- random within the strata
- error is not random
- less clumping, better picture
- more expensive to select sampling points.
- overhead of implementation.

$$R(n,) = \frac{1}{n^2} \sum_{l=0}^{n-1} \sum_{k=0}^{n-1} I\left(i + \frac{k+z_1}{n}, j + \frac{l+z_2}{n}\right)$$