

pixel: $[i, i+1] \times [j, j+1]$

original: for each pixel (i, j) do
 $c_{ij} = \text{raycolor}(i + 0.5, j + 0.5)$

regular supersampling:

for each pixel (i, j) do

$c = 0$

for $p = 0$ to $n-1$ do

for $q = 0$ to $n-1$ do

$c = c + \text{raycolor}(i + (p+0.5)/n, j + (q+0.5)/n)$

~~c_{ij}~~

$c_{ij} = c/n^2$

equivalent to rendering

$n \times n \times r \times r$ image + averaging blocks of
 $n \times n \times r \times r$ to get an $r \times r$ image

Slide 2

Random samples

for each pixel (i, j) do

$c = 0$

for $q = 0$ to n^2 do

$c = c + \text{ray-color}(i + \varepsilon, j + \varepsilon)$

$c_{ij} = c / n^2$

ε is a uniform random number in the range $[0, 1)$

Noise will likely be noticeable if individual pixel samples are jittered

Random perturbation of a regular grid!

for each pixel (i, j) do

$c = 0$

for $r = 0$ to $n-1$ do

for $q = 0$ to $n-1$ do

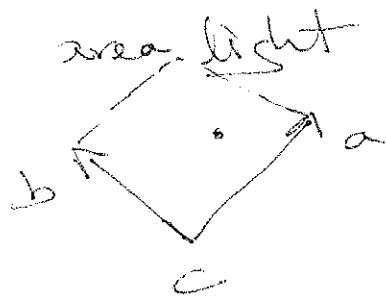
$c = c + \text{ray-color}(i + (r + \varepsilon)/n, j + (q + \varepsilon)/n)$

$c_{ij} = c / r^2$

Soft shadows

slide 28

N light points each with the intensity $1/N$ of the light
 equivalent to regular sampling in anti-aliasing
 $\Rightarrow$ many mini-light required to get smooth results



$$r = c + \xi_1 a + \xi_2 b$$

random pt r on surface of light

jitter both pixel samples + light samples:

for each pixel (i, j) do

$c = 0$

generate $N = n^2$ jittered 2D points r $r[]$
 " " " " $S[]$

shuffle r to $S \leftarrow$ ensure no correlations between r + S

for $p = 0$ to $N-1$ do

$$c = c + \text{ray-color}(\underbrace{i + r[p].x(), j + r[p].y()}_{\text{jittered } x, y}, \underbrace{s(p)}_{\text{static}})$$

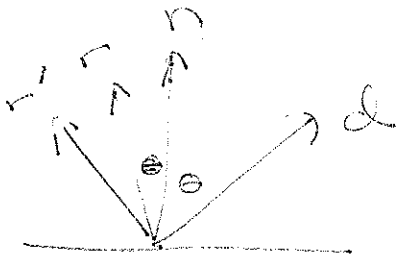
$$C_{ij} = c / N$$

shuffle

for $i = N-1$ down to 1 do

cross = rand integer between 0 and i inclusive

swap array elements i and $cross$



to r & r' , sample a random $\square$ of width a
 a constant degree of blur

r & bias with $\vec{w} = \vec{r}^2$ (+ $\vec{u}, \vec{v}$)

$$u = -\frac{a}{2} + \epsilon a$$

$$v = -\frac{a}{2} + \epsilon' a$$

$$r' = r + u\vec{u} + v\vec{v}$$

r' might be below the surface $\rightarrow$
 set color contribution to zero