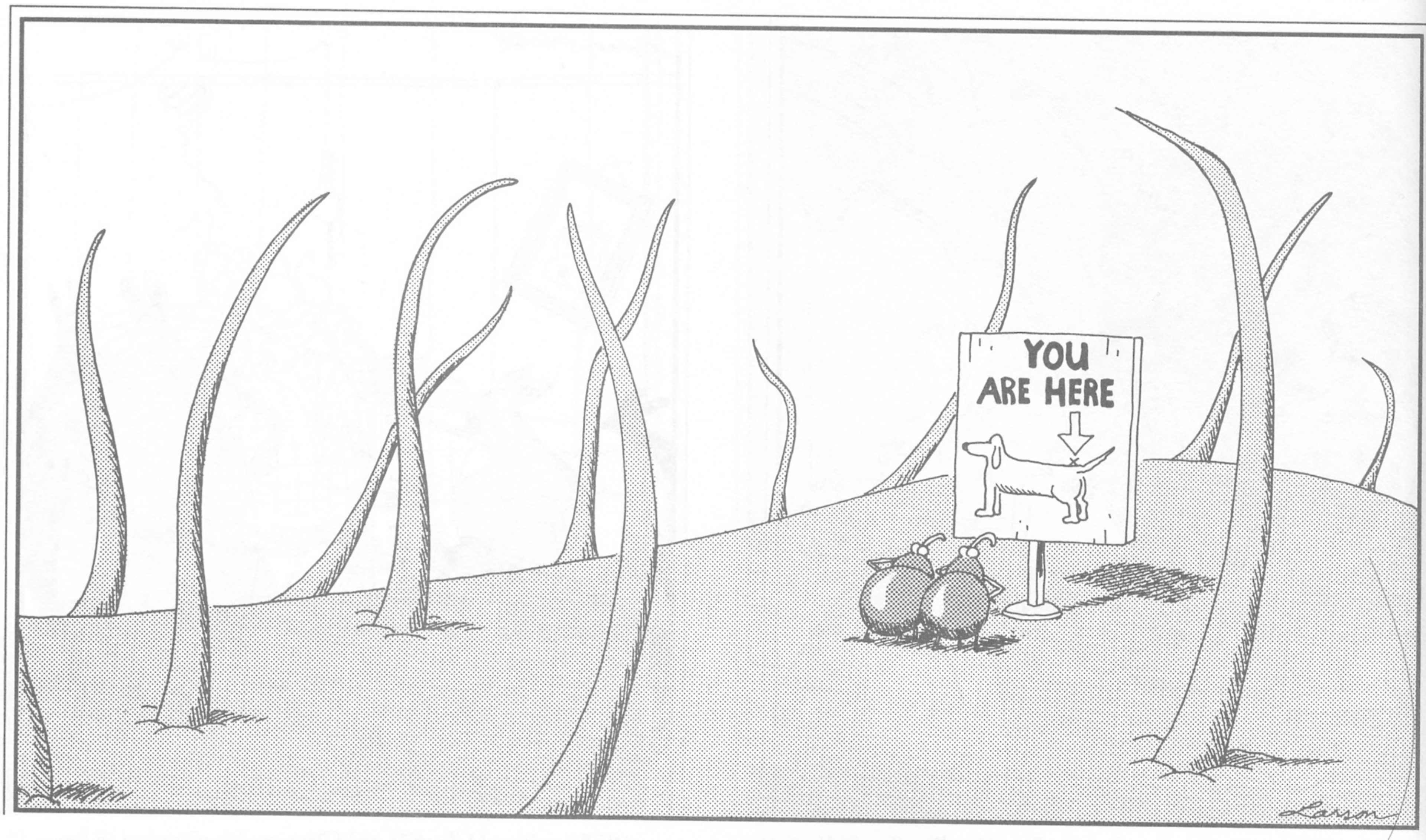
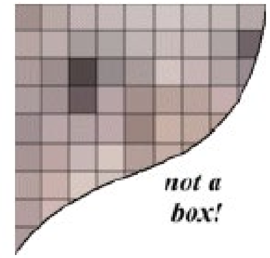
Sampling and Monte-Carlo Integration

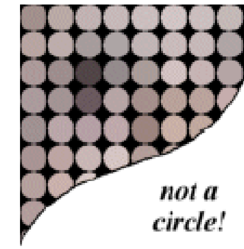
Sampling and Monte-Carlo Integration

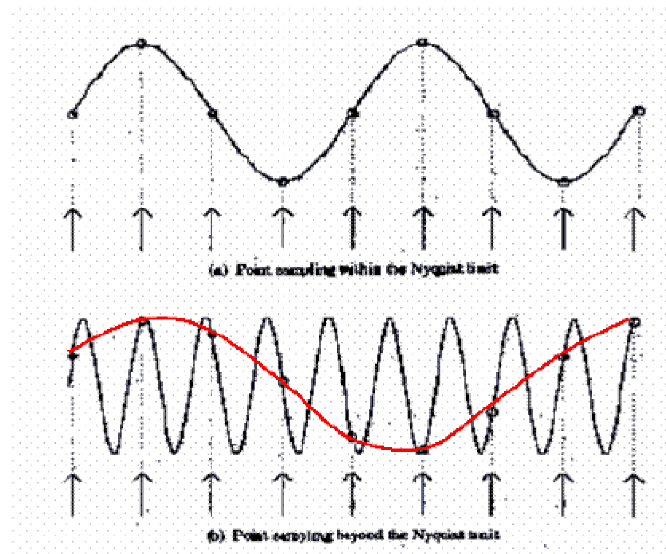


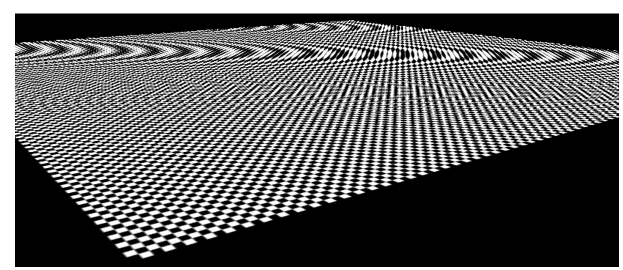
Last Time

- Pixels are samples
- Sampling theorem
- Convolution & multiplication
- Aliasing: spectrum replication
- Ideal filter
  - And its problems
- Reconstruction
- Texture prefiltering, mipmaps









Quiz solution: Homogeneous sum

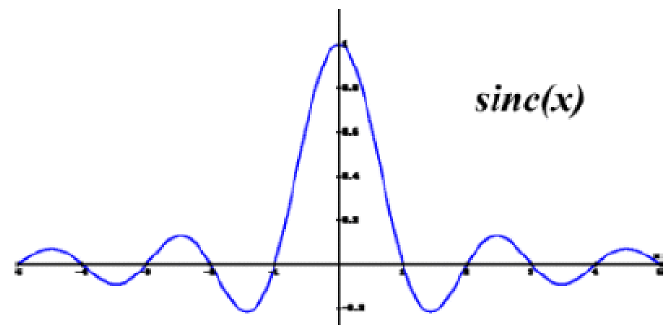
- $(x_1, y_1, z_1, 1) + (x_2, y_2, z_2, 1)$
  $= (x_1+x_2, y_1+y_2, z_1+z_2, 2)$
  ¼ $((x_1+x_2)/2, (y_1+y_2)/2, (z_1+z_2)/2)$
- This is the average of the two points
- General case: consider the homogeneous version of (x_1, y_1, z_1) and (x_2, y_2, z_2) with w coordinates w_1 and w_2
- $(x_1 w_1, y_1 w_1, z_1 w_1, w_1) + (x_2 w_2, y_2 w_2, z_2 w_2, w_2)$
  $= (x_1 w_1+x_2w_2, y_1 w_1+y_2w_2, z_1 w_1+z_2w_2, w_1+w_2)$
  ¼ $((x_1 w_1+x_2w_2)/(w_1+w_2), (y_1 w_1+y_2w_2)/(w_1+w_2),$
  $(z_1 w_1+z_2w_2)/(w_1+w_2))$
- This is the weighted average of the two geometric points

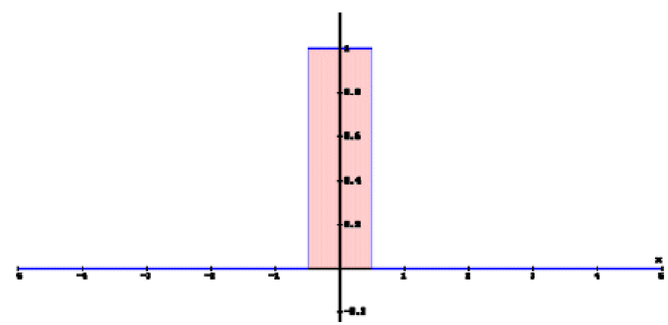
Today’s lecture

- Antialiasing in graphics
- Sampling patterns
- Monte-Carlo Integration
- Probabilities and variance
- Analysis of Monte-Carlo Integration

Ideal sampling/reconstruction

- Pre-filter with a perfect low-pass filter
  - Box in frequency
  - Sinc in time
- Sample at Nyquist limit
  - Twice the frequency cutoff
- Reconstruct with perfect filter
  - Box in frequency, sinc in time
- And everything is great!



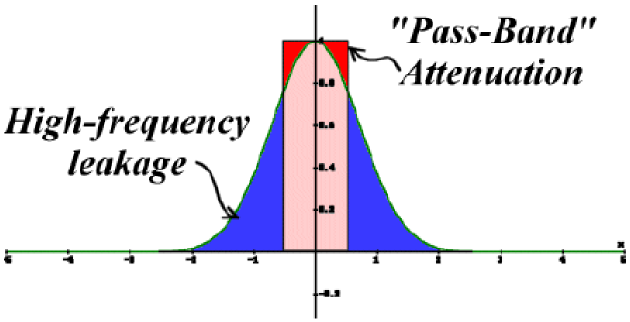


Difficulties with perfect sampling

- Hard to prefilter
- Perfect filter has infinite support
  - Fourier analysis assumes infinite signal and complete knowledge
  - Not enough focus on local effects
- And negative lobes
  - Emphasizes the two problems above
  - Negative light is bad
  - Ringing artifacts if prefiltering or supports are not perfect

At the end of the day

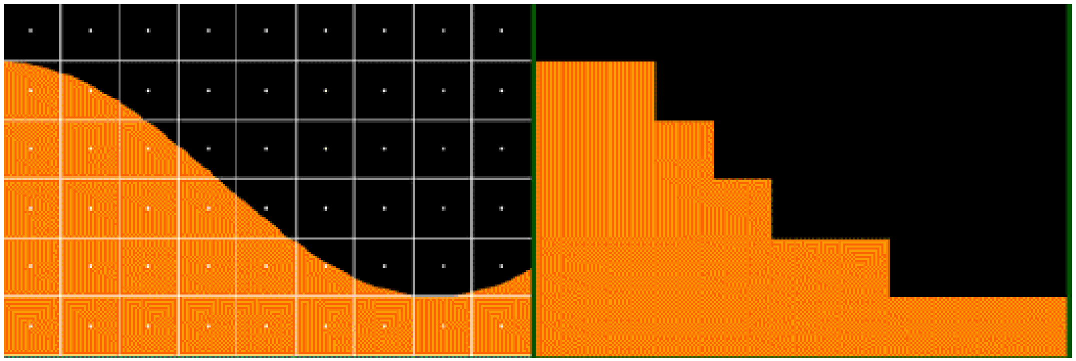
- Fourier analysis is great to understand aliasing
- But practical problems kick in
- As a result there is no perfect solution
- Compromises between
  - Finite support
  - Avoid negative lobes
  - Avoid high-frequency leakage
  - Avoid low-frequency attenuation
- Everyone has their favorite cookbook recipe
  - Gaussian, tent, Mitchell bicubic



The special case of edges

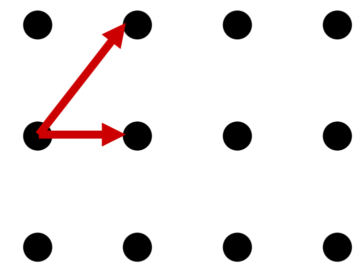
- An edge is poorly captured by Fourier analysis
  - It is a local feature
  - It combines all frequencies (sinc)
- Practical issues with edge aliasing lie more in the jaggies (tilted lines) than in actual spectrum replication

Jagged boundaries

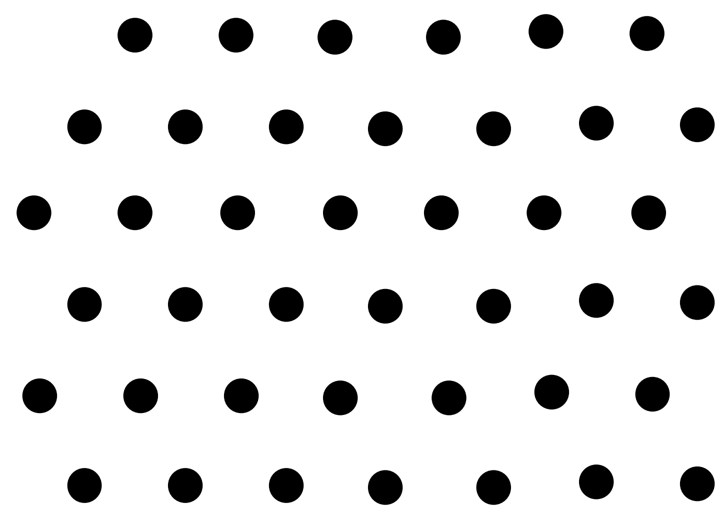


Anisotropy of the sampling grid

- More vertical and horizontal bandwidth
  - E.g. $\sqrt{2}$ less bandwidth in diagonal

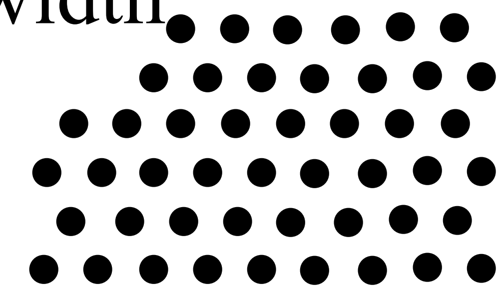


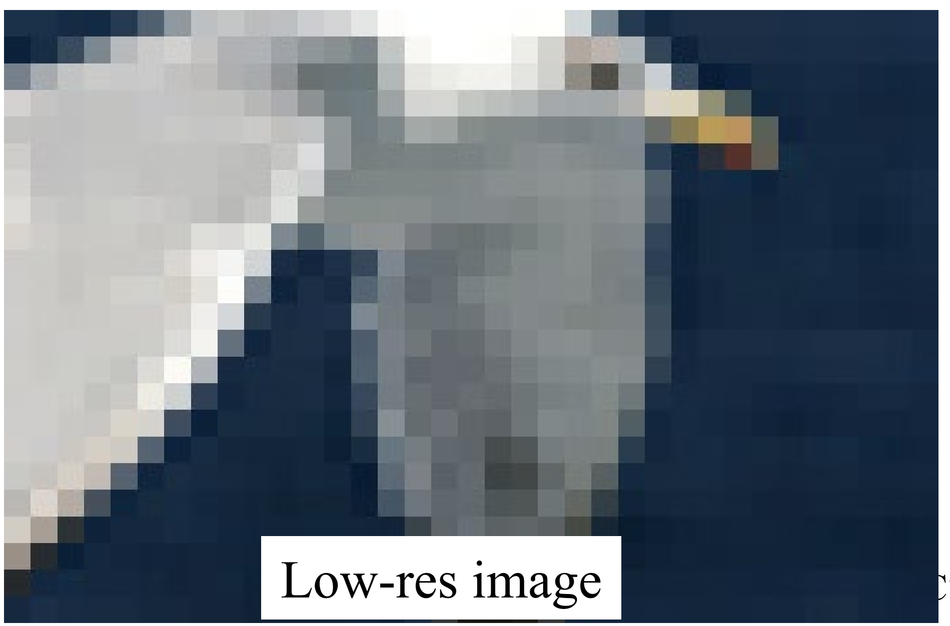
- A hexagonal grid would be better
  - Max anisotropy
    $cos 30° = \sqrt{3/4} = 0.86$
  - But less practical



Anisotropy of the sampling grid

- More vertical and horizontal bandwidth
- A hexagonal grid would be better
  - But less practical
- Practical effect: vertical and horizontal direction show when doing bicubic upsampling





Low-res image

Bicubic upsampling

Philosophy about mathematics

- Mathematics are great tools to model (i.e. describe) your problems
- They afford incredible power, formalism, generalization
- However it is equally important to understand the practical problem and how much the mathematical model fits

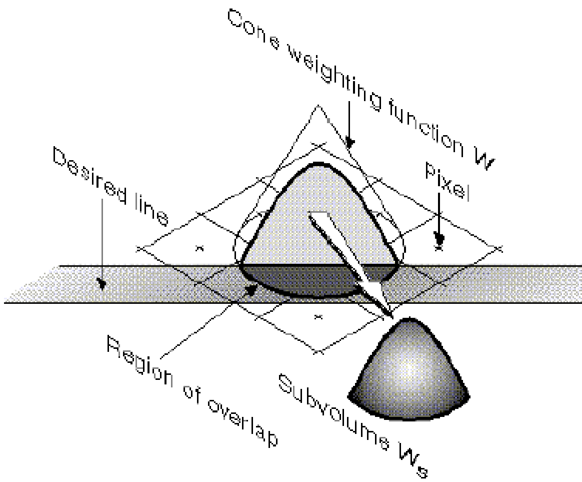
Questions?

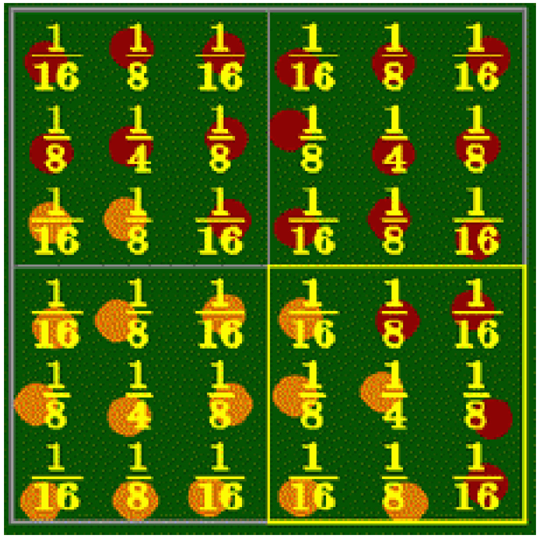
Today's lecture

- Antialiasing in graphics
- Sampling patterns
- Monte-Carlo Integration
- Probabilities and variance
- Analysis of Monte-Carlo Integration

Supersampling in graphics

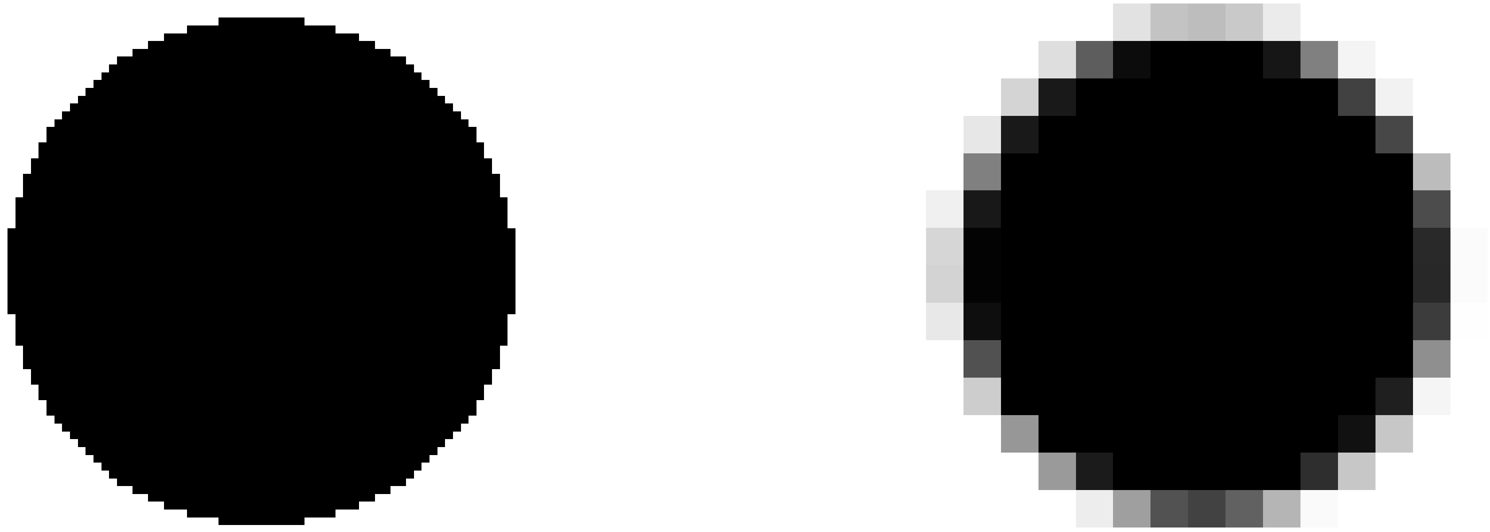
- Pre-filtering is hard
  - Requires analytical visibility
  - Then difficult to integrate analytically with filter
- Possible for lines, or if visibility is ignored
- usually, fall back to supersampling





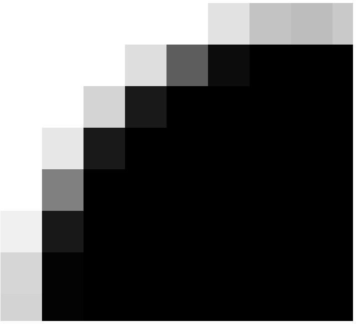
Uniform supersampling

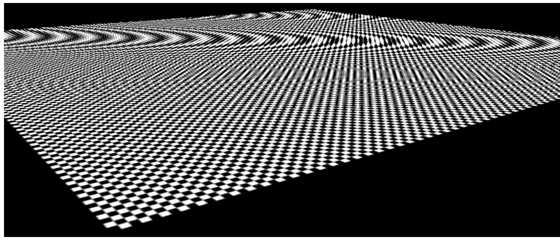
- Compute image at resolution k*width, k*height
- Downsample using low-pass filter
  (e.g. Gaussian, sinc, bicubic)



Uniform supersampling

- Advantage:
  - The first (super)sampling captures more high frequencies that are not aliased
  - Downsampling can use a good filter
- Issues
  - Frequencies above the (super)sampling limit are still aliased
- Works well for edges, since spectrum replication is less an issue
- Not as well for repetitive textures
  - But mipmapping can help





Multisampling vs. supersampling

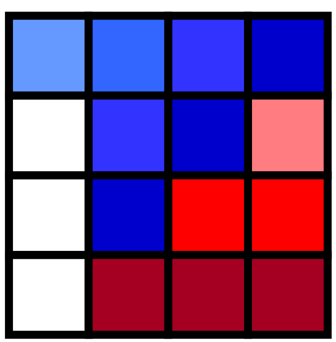
- Observation:
  - Edge aliasing mostly comes from visibility/rasterization issues
  - Texture aliasing can be prevented using prefiltering
- **Multi**sampling idea:
  - Sample rasterization/visibility at a higher rate than shading/texture
- In practice, same as **super**sampling, except that all the subpixel get the same color if visible

Multisampling vs. supersampling

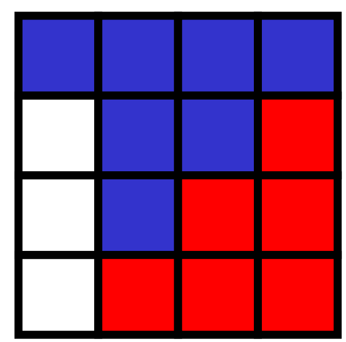
```
For each triangle
    For each pixel
        Compute pixelcolor //only once for all subpixels
        For each subpixel
            If (all edge equations positive &&
               zbuffer [subpixel] > currentz )
                Then Framebuffer[subpixel]=pixelcolor
```

- The subpixels of a pixel get different colors only at edges of triangles or at occlusion boundaries

Example:
2 Gouraud-shaded triangles



Subpixels in supersampling



Subpixels in multisampling

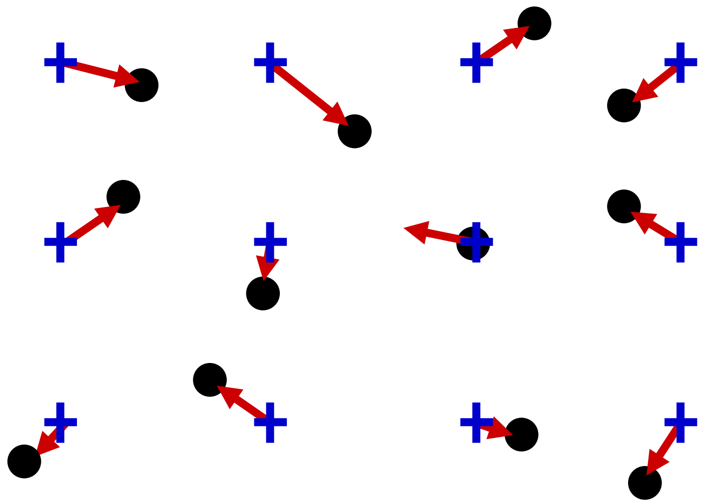
Questions?

Uniform supersampling

- Problem: supersampling only pushes the problem further: The signal is still not bandlimited
- Aliasing happens

Jittering

- Uniform sample + random perturbation
- Sampling is now non-uniform
- Signal processing gets more complex
- In practice, adds noise to image
- But noise is better than aliasing Moiré patterns



Jittered supersampling

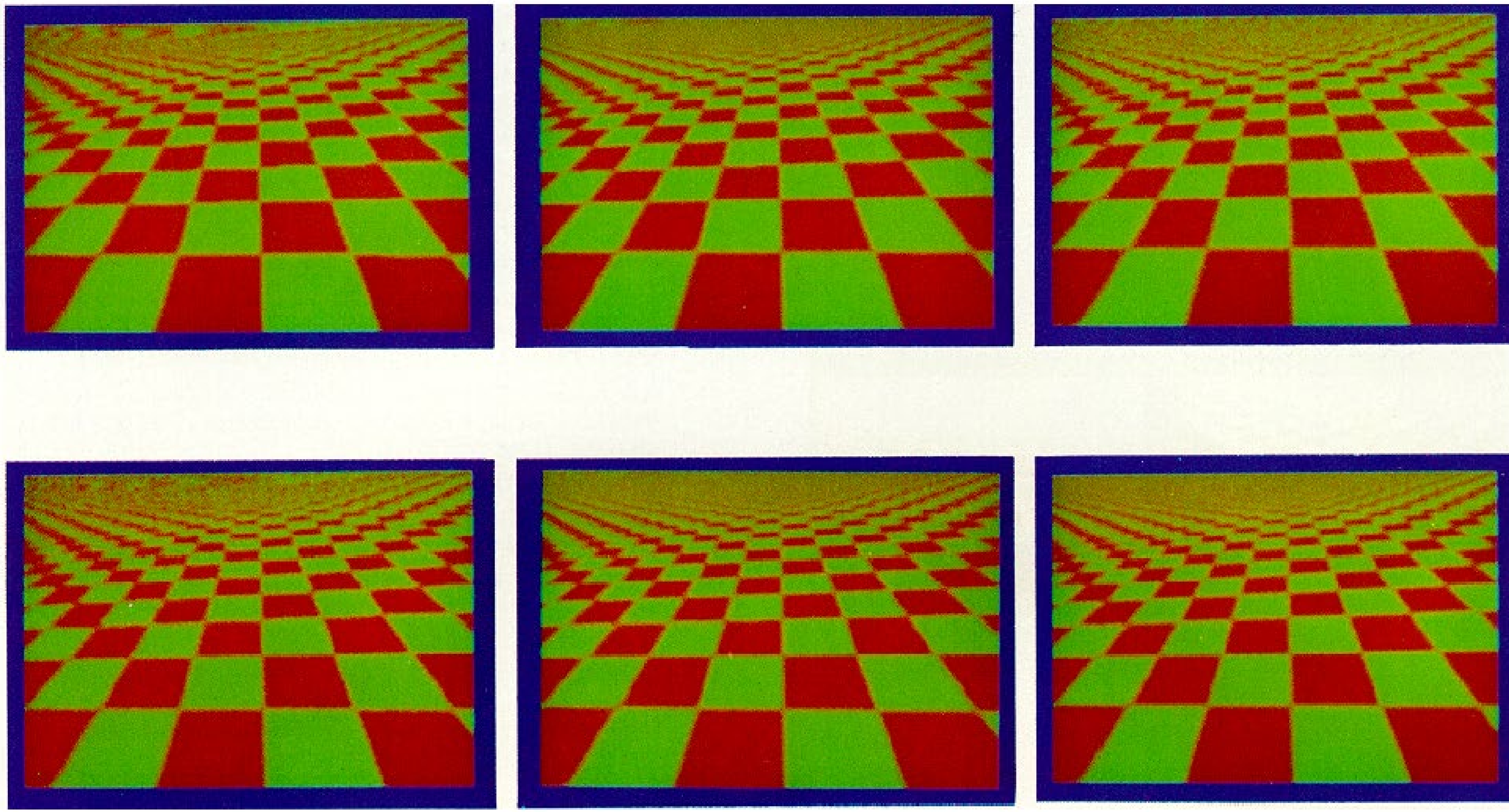


Figure 11
Jittered sampling of a slowly moving texture with jitter of 0, .5, and 1 from left to right and oversampling rates of 1 and 2 from top to bottom.

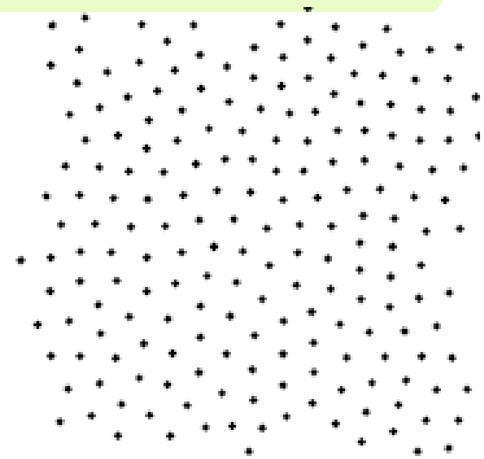
Jittering

- Displaced by a vector a fraction of the size of the subpixel distance
- Low-frequency Moire (aliasing) pattern replaced by noise
- Extremely effective
- Patented by Pixar!
- When jittering amount is 1, equivalent to stratified sampling (cf. later)

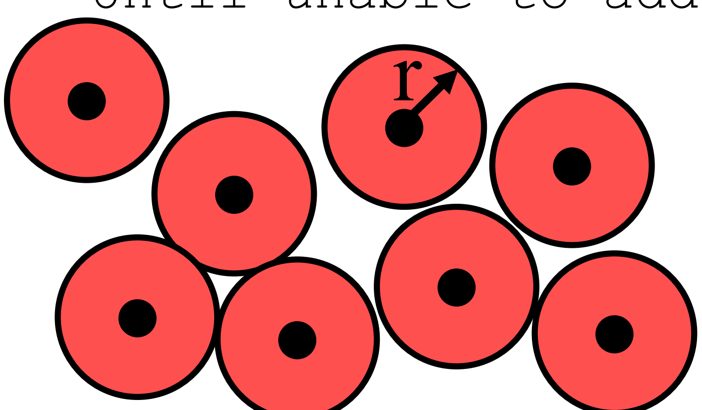
Poisson disk sampling and blue noise

- Essentially random points that are not allowed to be closer than some radius r
- Dart-throwing algorithm:

```
Initialize sampling pattern as empty
Do
   Get random point P
   If P is farther than r from all samples
      Add P to sampling pattern
Until unable to add samples for a long time
```

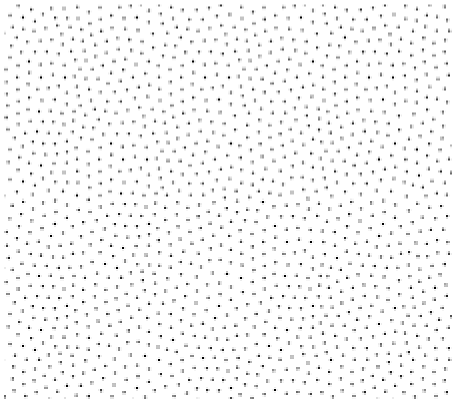


From Hiller et al.

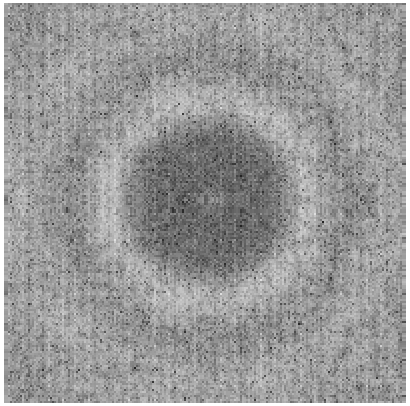


Poisson disk sampling and blue noise

- Essentially random points that are not allowed to be closer than some radius r
- The spectrum of the Poisson disk pattern is called blue noise:
- No low frequency
- Other criterion: Isotropy (frequency content must be the same for all direction)

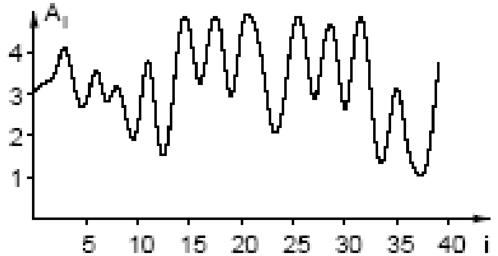


Poisson disk pattern



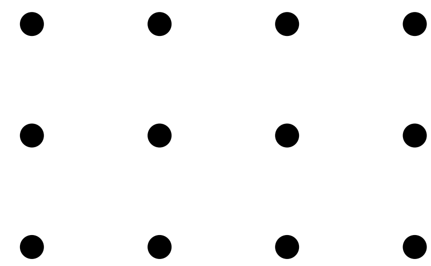
Fourier transform

Anisotropy (power spectrum per direction)

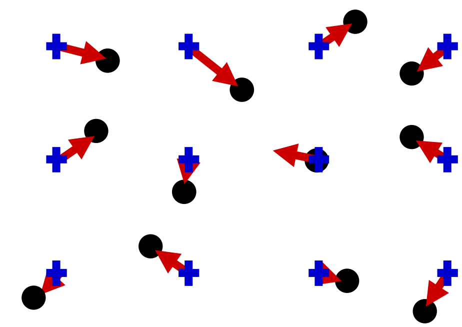


Recap

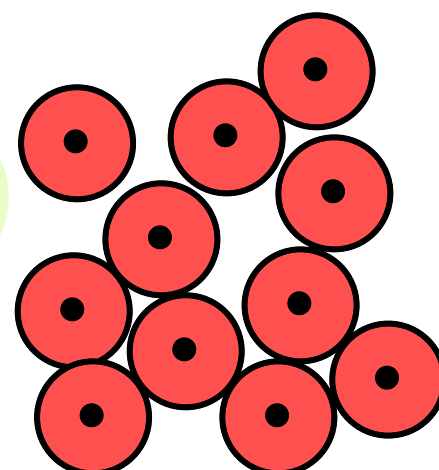
- Uniform supersampling
  - Not so great



- Jittering
  - Great, replaces aliasing by noise

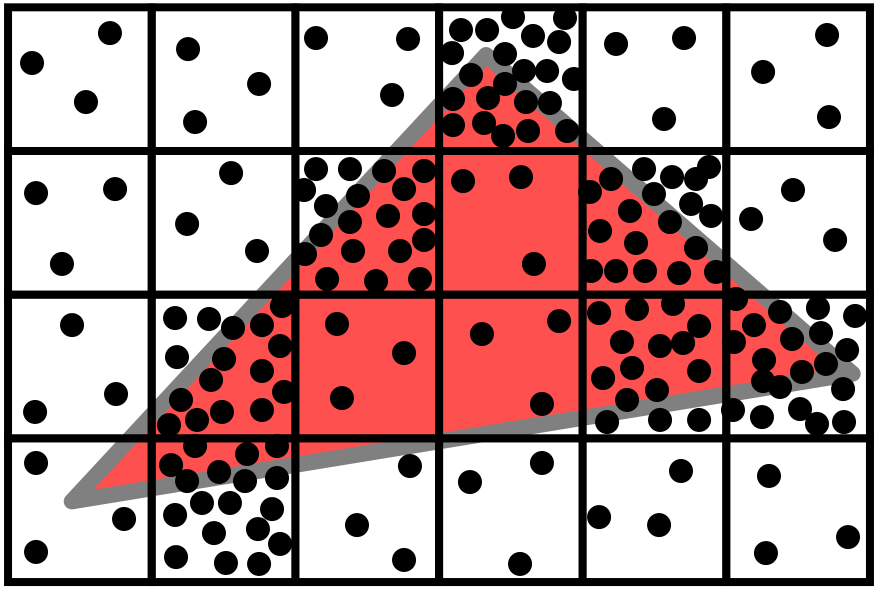


- Poisson disk sampling
  - Equally good, but harder to generate
  - Blue noise and good (lack of) anisotropy



Adaptive supersampling

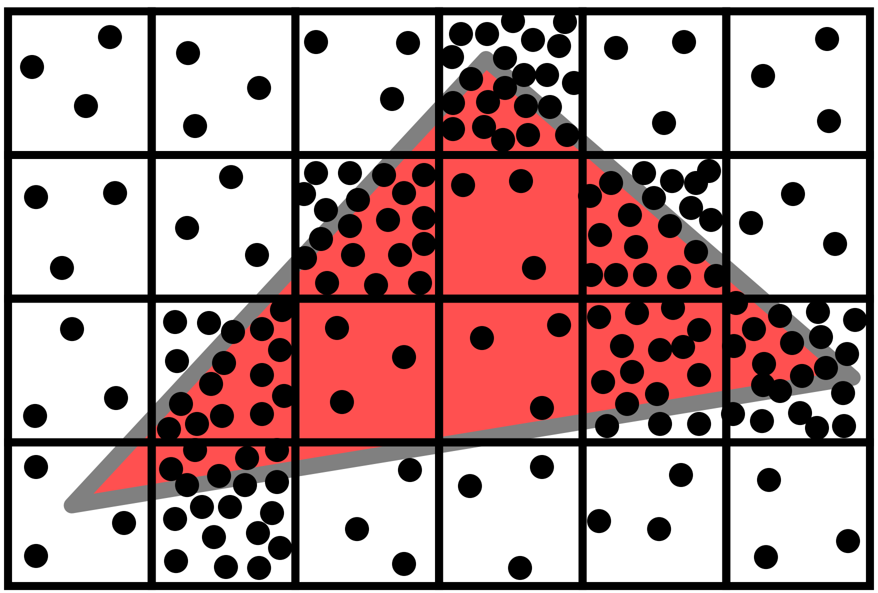
- Use more sub-pixel samples around edges



Adaptive supersampling

- Use more sub-pixel samples around edges

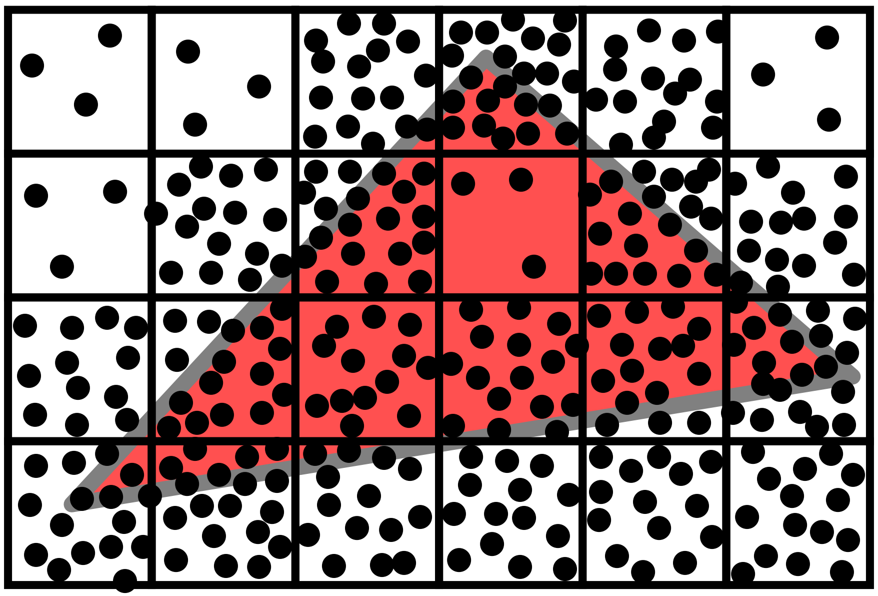
```
Compute color at small number of sample
If their variance is high
   Compute larger number of samples
```



Adaptive supersampling

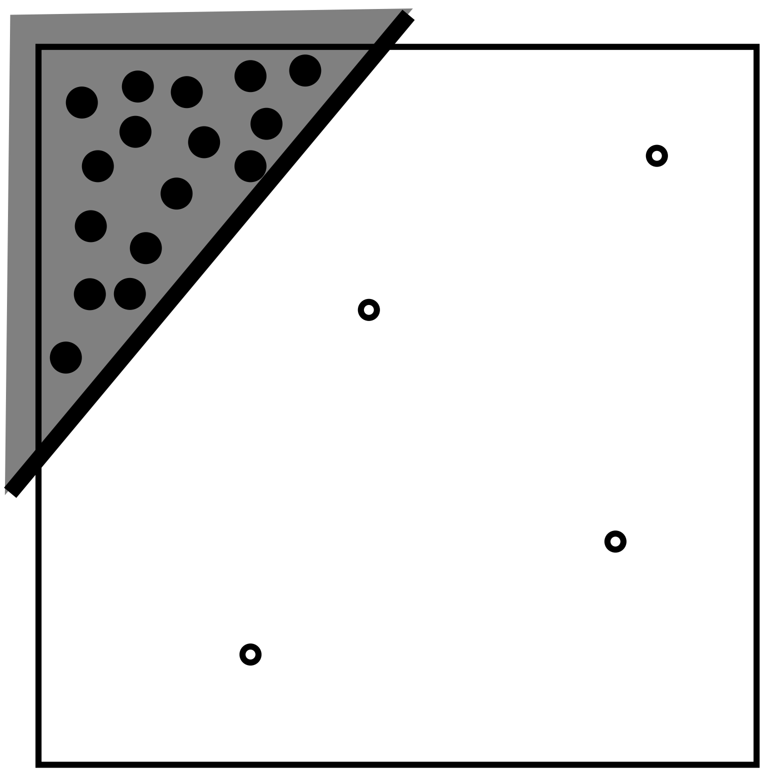
- Use more sub-pixel samples around edges

```
Compute color at small number of sample
If variance with neighbor pixels is high
   Compute larger number of samples
```



Problem with non-uniform distribution

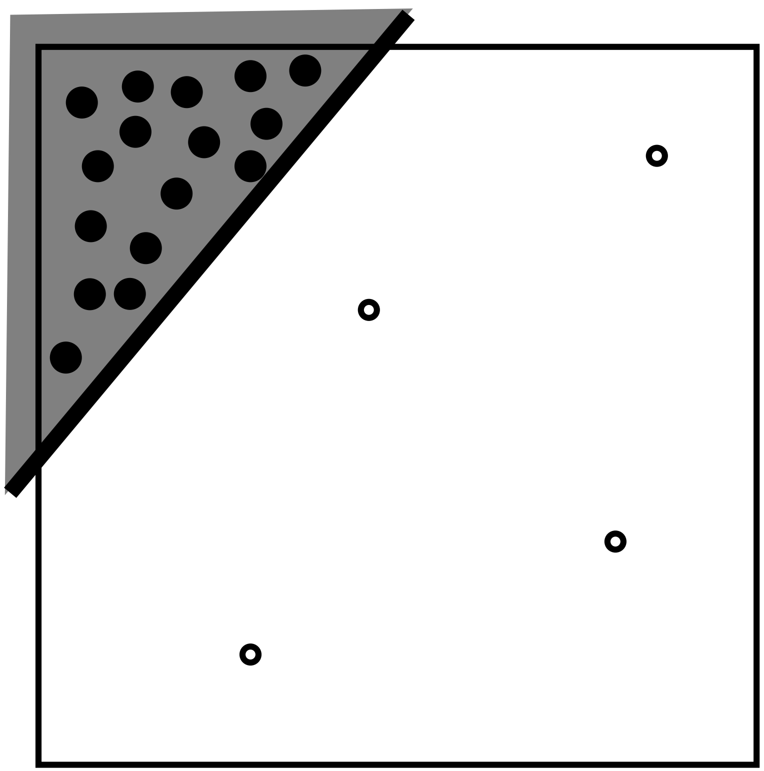
- Reconstruction can be complicated

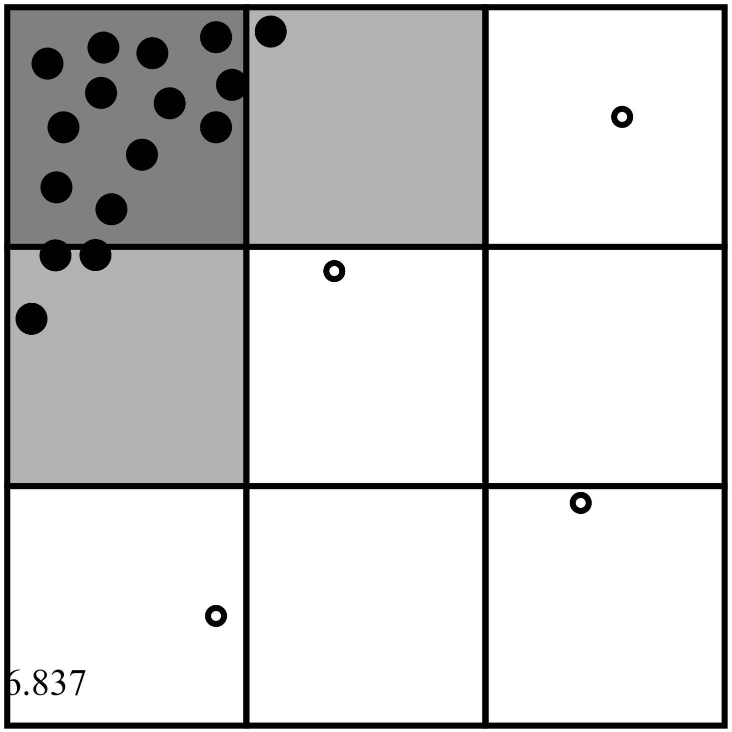


80% of the samples are black
Yet the pixel should be light grey

Problem with non-uniform distribution

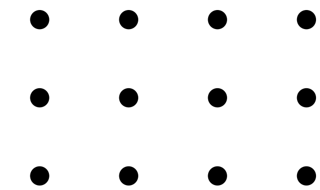
- Reconstruction can be complicated
- Solution: do a multi-level reconstruction
  - Reconstruct uniform sub-pixels
  - Filter those uniform sub-pixels



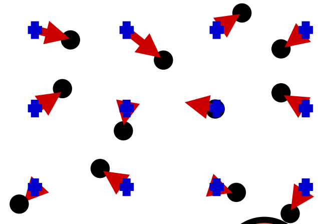


Recap

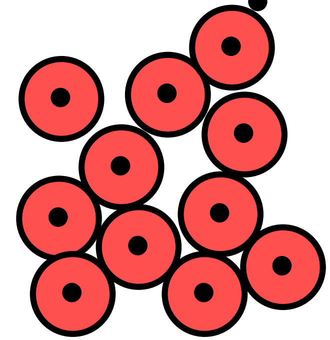
- Uniform supersampling
  - Not so great



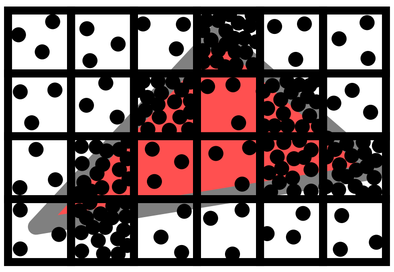
- Jittering
  - Great, replaces aliasing by noise



- Poisson disk sampling
  - Equally good, but harder to generate
  - Blue noise and good (lack of) anisotropy



- Adaptive sampling
  - Focus computation where needed
  - Beware of false negative
  - Complex reconstruction



Questions?

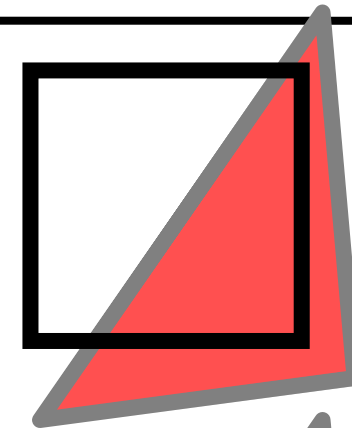
Today's lecture

- Antialiasing in graphics
- Sampling patterns
- Monte-Carlo Integration
- Probabilities and variance
- Analysis of Monte-Carlo Integration

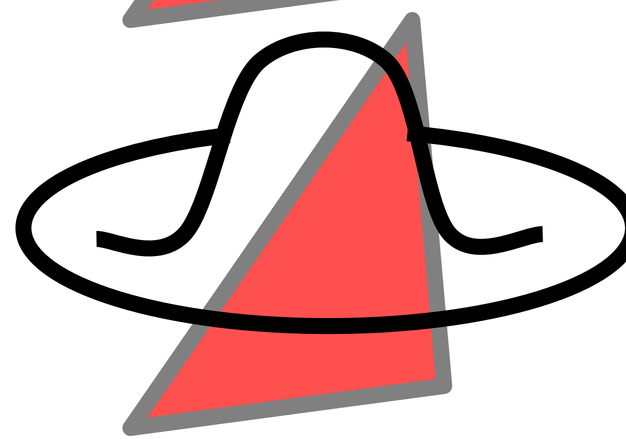
Shift of perspective

- So far, Antialiasing as signal processing
- Now, Antialiasing as integration
- Complementary yet not always the same

Why integration?



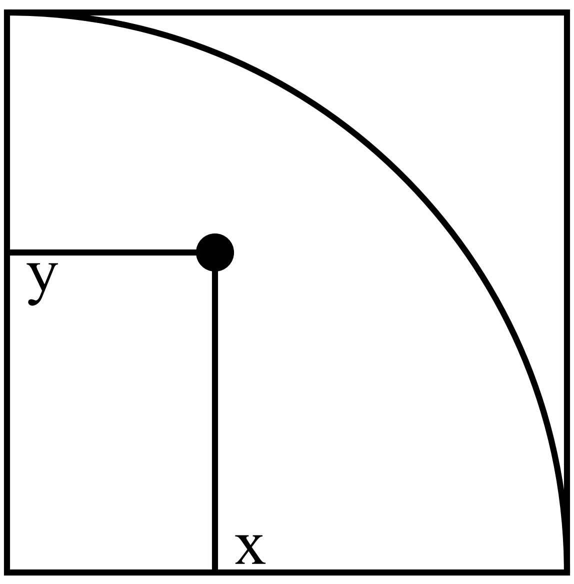
- Simple version:
  compute pixel coverage



- More advanced:
  Filtering (convolution)
  is an integral
  pixel = s filter * color
- And integration is useful in tons of places in graphics

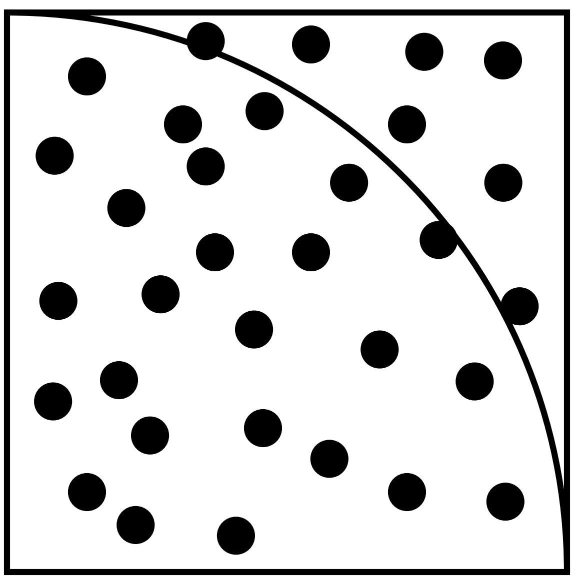
Monte-Carlo computation of π

- Take a square
- Take a random point (x,y) in the square
- Test if it is inside the ¼ disc ($x^2+y^2 < 1$)
- The probability is $\pi/4$



Monte-Carlo computation of π

- The probability is π /4
- Count the inside ratio n = # inside / total # trials
- $\pi \approx n * 4$
- The error depends on the number or trials



Why not use Simpson integration?

- Yeah, to compute π,
  Monte Carlo is not very efficient
- But convergence is independent of dimension
- Better to integrate high-dimensional functions
- For d dimensions, Simpson requires N^d domains

Dumbest Monte-Carlo integration

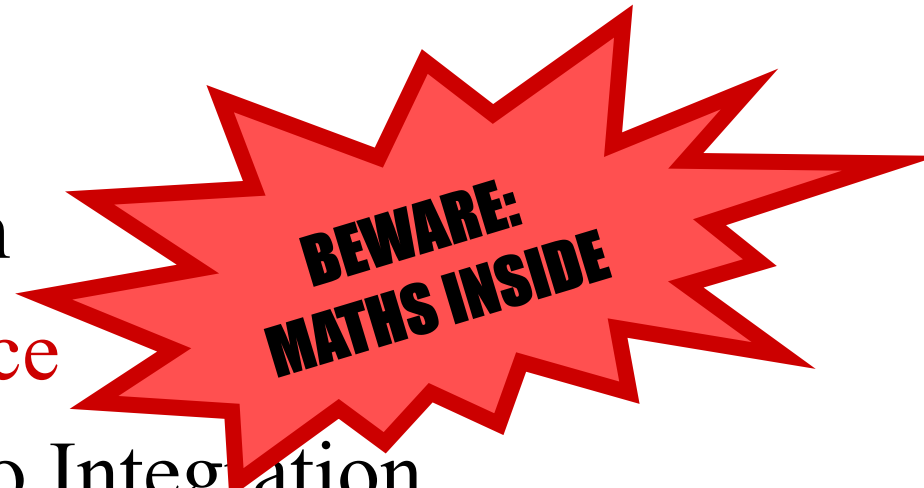
- Compute 0.5 by flipping a coin
- 1 flip: 0 or 1=> average error =0.5
- 2 flips: 0, 0.5, 0.5 or 1 =>average error=0. 25
- 4 flips: 0 (*1),0.25 (*4), 0.5 (*6), 0.75(*4), 1(*1) => average error =0.1875

- Does not converge very fast
- Doubling the number of samples does not double accuracy

Questions?

Today's lecture

- Antialiasing in graphics
- Sampling patterns
- Monte-Carlo Integration
- Probabilities and variance
- Analysis of Monte-Carlo Integration



Review of probability (discrete)

- Random variable can take discrete values x_i
- Probability p_i for each x_i

  $0 \cdot p_i \cdot 1$

  If the events are mutually exclusive, $\Sigma\, p_i = 1$
- Expected value

$$E(x) = \sum_{i=1}^{n} p_i x_i$$

- Expected value of function of random variable
  – $f(x_i)$ is also a random variable

$$E[f(x)] = \sum_{i=1}^{n} p_i f(x_i)$$

Ex: fair dice

$$
\begin{aligned}
E(x_{dice}) &= \sum_{i=1}^{6} p_i x_i \\
&= \sum_{i=1}^{6} \frac{1}{6} x_i \\
&= \frac{1}{6}(1+2+3+4+5+6) \\
&= 3.5
\end{aligned}
$$

Variance & standard deviation

- Variance σ^2:
  Measure of deviation from expected value
- Expected value of square difference (MSE)

$$\sigma^2 = E[(x - E[x])^2] = \sum_i (x_i - E[x])^2 p_i$$

- Also

$$\sigma^2 = E[x^2] - (E[x])^2$$

- Standard deviation σ:
  square root of variance (notion of error, RMS)

Questions?

Continuous random variables

- Real-valued random variable x
- Probability density function (PDF) p(x)
  - Probability of a value between x and x+dx is p(x) dx
- Cumulative Density Function (CDF) P(y):
  - Probability to get a value lower than y

$$P(y) = Pr(x \leq y) = \int_{-\infty}^{y} p(x) dx$$

Properties

- p(x) ¸ 0   but can be greater than 1   !!!!

$$\int_{-\infty}^{\infty} p(x)dx =$$

$$p(x) =$$

$$Pr(a \leq x \leq b) =$$

- P is positive and non-decreasing

Properties

- p(x) ¸ 0 but can be greater than 1 !!!!

$$\int_{-\infty}^{\infty} p(x)dx = 1$$

$$p(x) = \frac{dP(x)}{dx}$$

$$Pr(a \le x \le b) = P(b) - P(a) = \int_a^b p(z)dz$$

- P is positive and non-decreasing

Example

- Uniform distribution between a and b

- Dirac distribution

Expected value

$$E[x] = \int_{-\infty}^{\infty} x p(x) dx$$

$$E[f(x)] = \int_{-\infty}^{\infty} f(x) p(x) dx$$

- Expected value is linear

$$E[f_1(x) + a\, f_2(x)] \;=\; E[f_1(x)] + a\, E[f_2(x)]$$

Variance

$$\sigma^2 = E[(x - E[x])^2] = \int_{-\infty}^{\infty} (x - E[x])^2 p(x) dx$$

- Variance is not linear !!!!
- $\sigma^2[x+y] = \sigma^2[x] + \sigma^2[y] + 2\,\mathrm{Cov}[x,y]$
- Where Cov is the covariance
  - $\mathrm{Cov}[x,y] = E[xy] - E[x]\,E[y]$
  - Tells how much they are big at the same time
  - Null if variables are independent
- But $\sigma^2[ax] = a^2\,\sigma^2[x]$