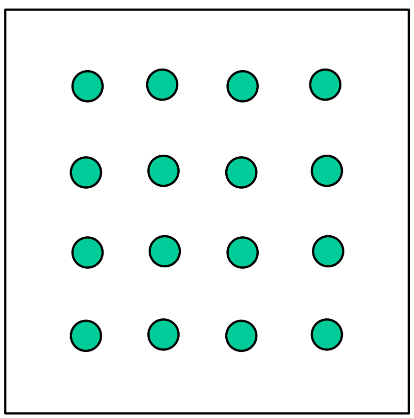
CS 563 Advanced Topics in Computer Graphics
Sampling and Reconstruction III

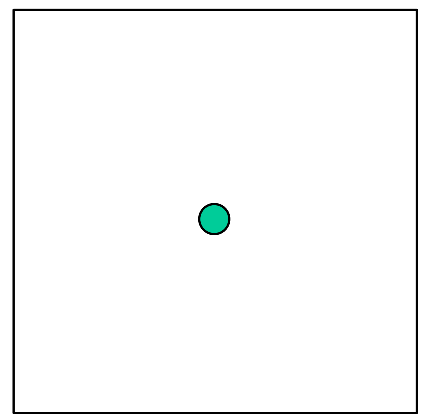
by Emmanuel Agu

Uniform Supersampling

- Increasing the sampling rate:
  - Moves each spectra copy further apart
  - Potentially reducing the overlap and thus aliasing
- Resulting samples must be resampled (filtered) to image sampling rate



$$Pixel = \sum_k w_k \times Sample_k$$

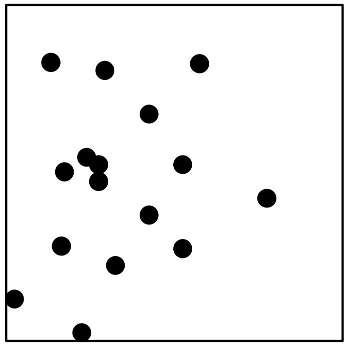


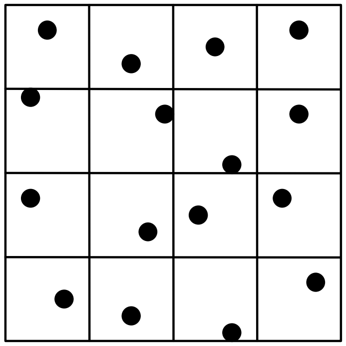
Non-Uniform Sampling - Intuition

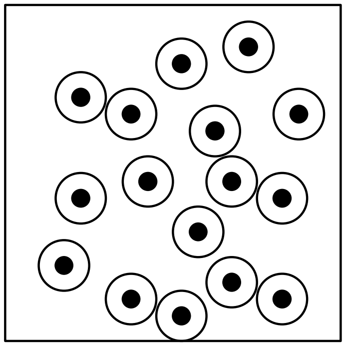
- Non-uniform sampling
  - Essentially, non-uniform sampling converts aliases into broadband noise
  - Less noticeable by eye
  - Noise is incoherent, and much less objectionable
  - Based on Yellot theory (1983)

Non-Uniform Sampling - Patterns

- Poisson
  - Pick ***n*** random points in sample space
- Uniform Jitter
  - Subdivide sample space into n regions
- Poisson Disk
  - Pick n random points, but not too close



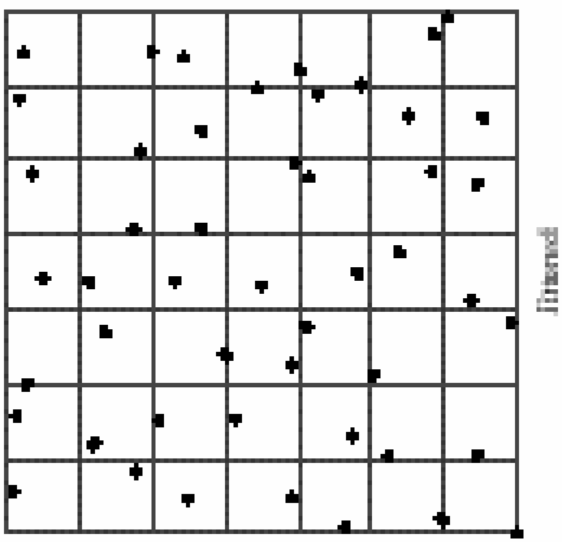




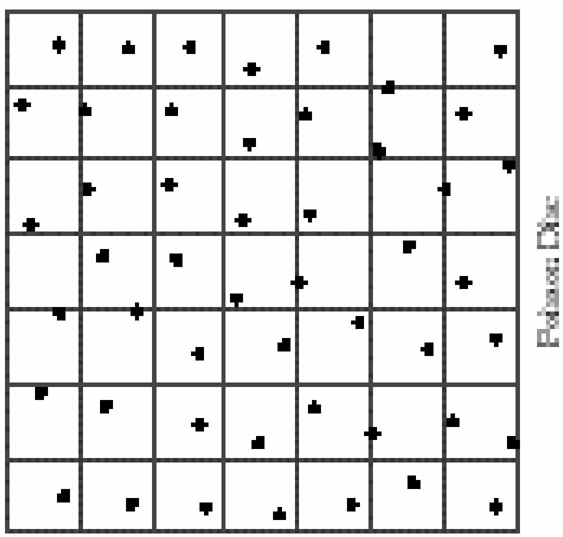
Best-Candidate Sampling

- Jittered stratification
  - Randomness (inefficient)
  - Clustering problems
  - Undersampling ("holes")
- Stratified, Low Discrepancy Sequences
  - Still (visibly) aliased
- "Ideal": Poisson disk distribution
  - too computationally expensive
- Best candidate sampling - approximation to Poisson disk

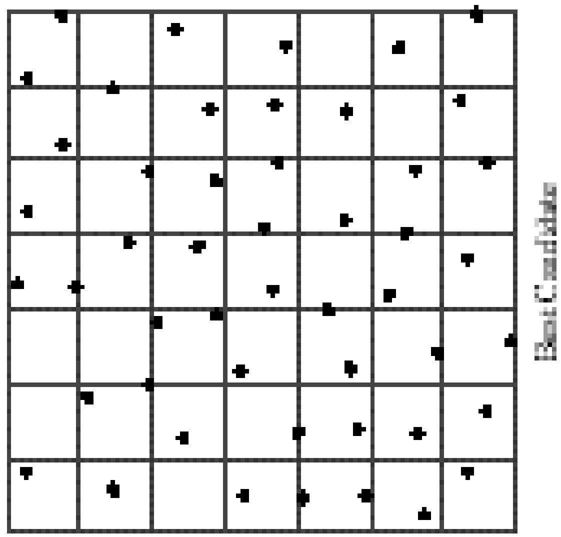
Best-Candidate Sampling



Jittered



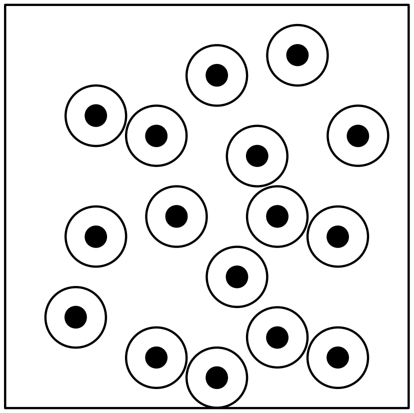
Poisson Disk



Best Candidate

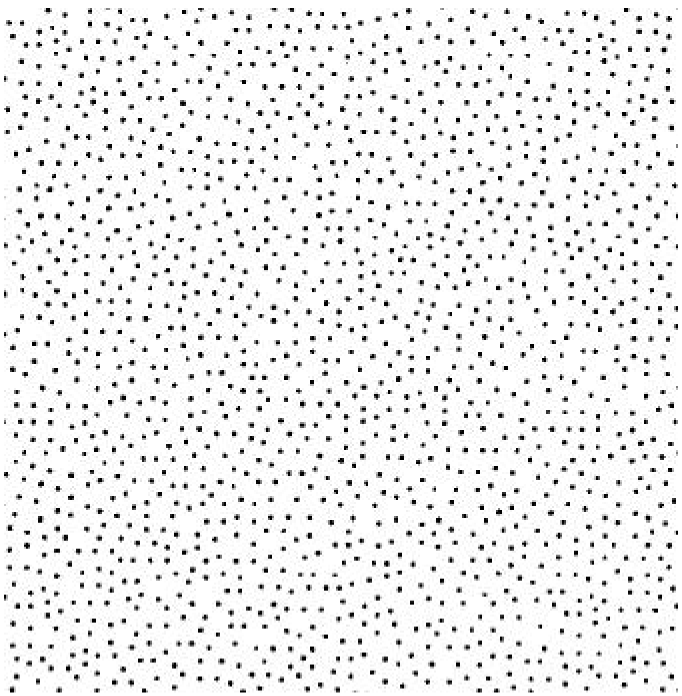
Poisson Disk

- Comes from eye structure of – rods and cones
- Dart Throwing
- No two points are closer than a threshold
- Very expensive – time consuming
- Compromise – Best Candidate Sampling
  - Don Mitchell
  - Generates many ***potential*** candidates randomly, only insert ***farthest one*** to all previous samples.
  - Compute “tilable pattern” offline that is reused by tiling the image plane (translating and scaling).
  - Toroidal topology – paste on toroid
  - Affects distance between points on top to bottom

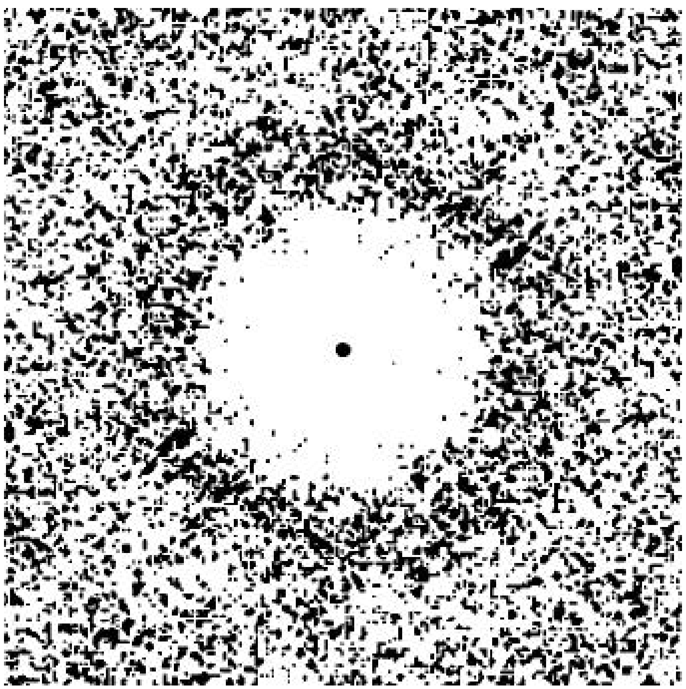


Poisson Disk Sampling

- Spectral characteristics:
  - **Poisson:** completely uniform (white noise). High and low frequencies equally present
  - **Poisson disc:** Pulse at origin (DC component of image), surrounded by empty ring (no low frequencies), surrounded by white noise



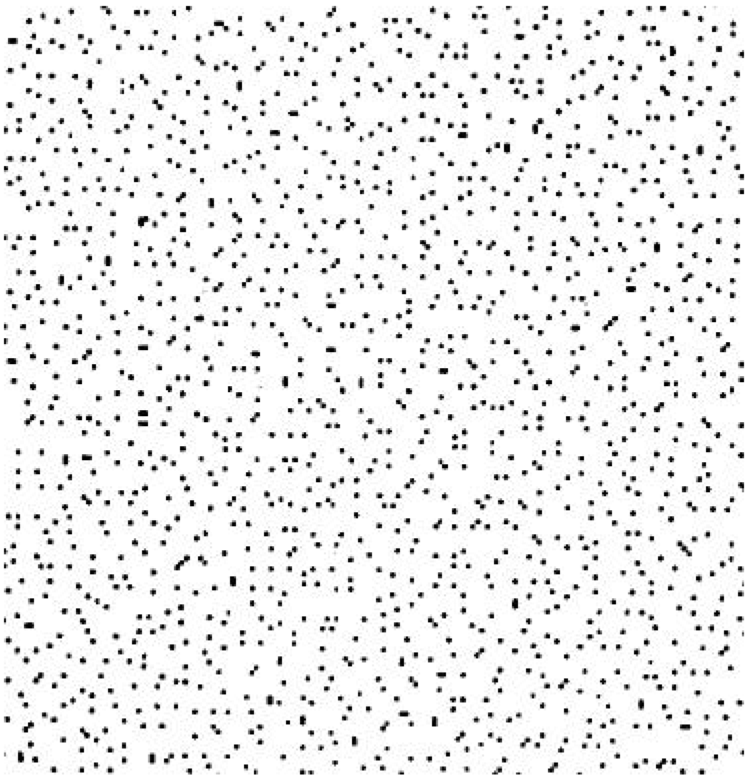
Spatial Domain



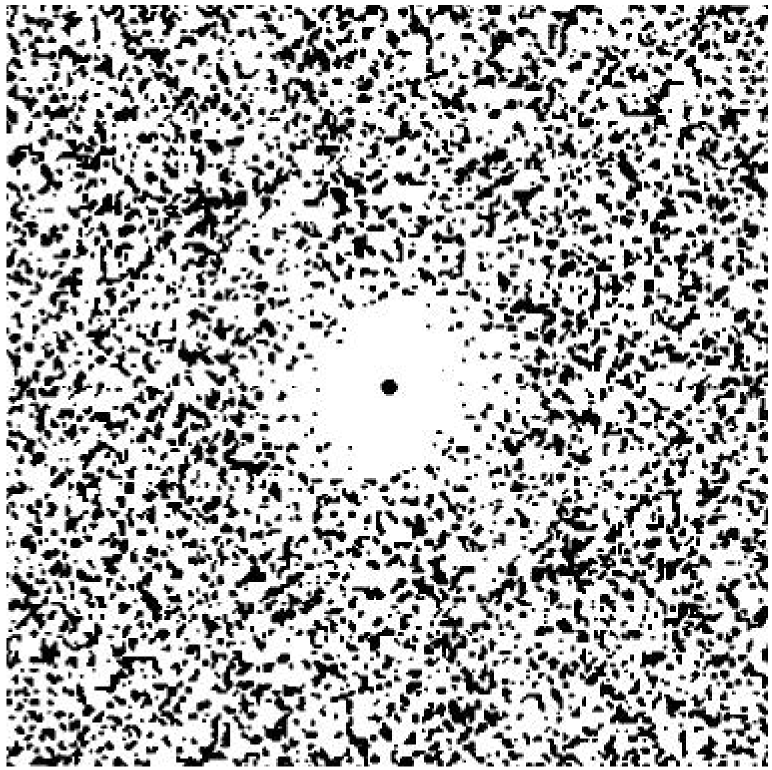
Fourier Domain

Uniform Jittered Sampling

- Spectral characteristics:
  - Jitter: Approximates Poisson disc spectrum,
  - **But** with a smaller empty disc.



Spatial Domain



Fourier Domain

Poisson Disk algorithm

$i \leftarrow 0$
while $i < N$
  $x_i \leftarrow$ unit()
  $y_i \leftarrow$ unit()
  reject $\leftarrow$ false
	Throw a dart.

for $k \leftarrow 0$ to $i-1$ $d \leftarrow (x_i - x_k)^2 + (y_i - y_k)^2$	*Check the distance to all other samples.*

if $d < (2r_p)^2$ then
      reject $\leftarrow$ true
      break
      endif
	This one is too close—forget it.

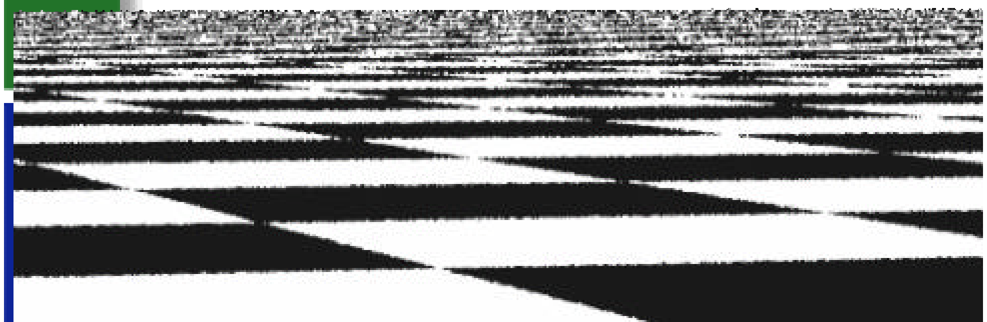
endfor	

if not *reject* then
    $i \leftarrow i + 1$
    endif
	Append this one to the pattern.

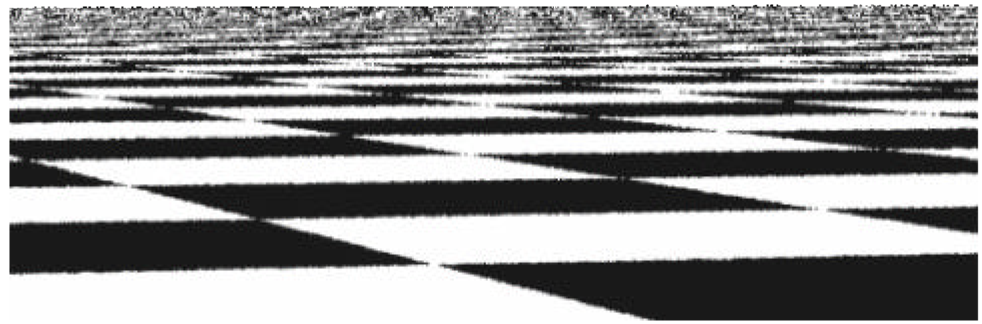
endwhile	

Texture

Jitter with 1 sample/pixel



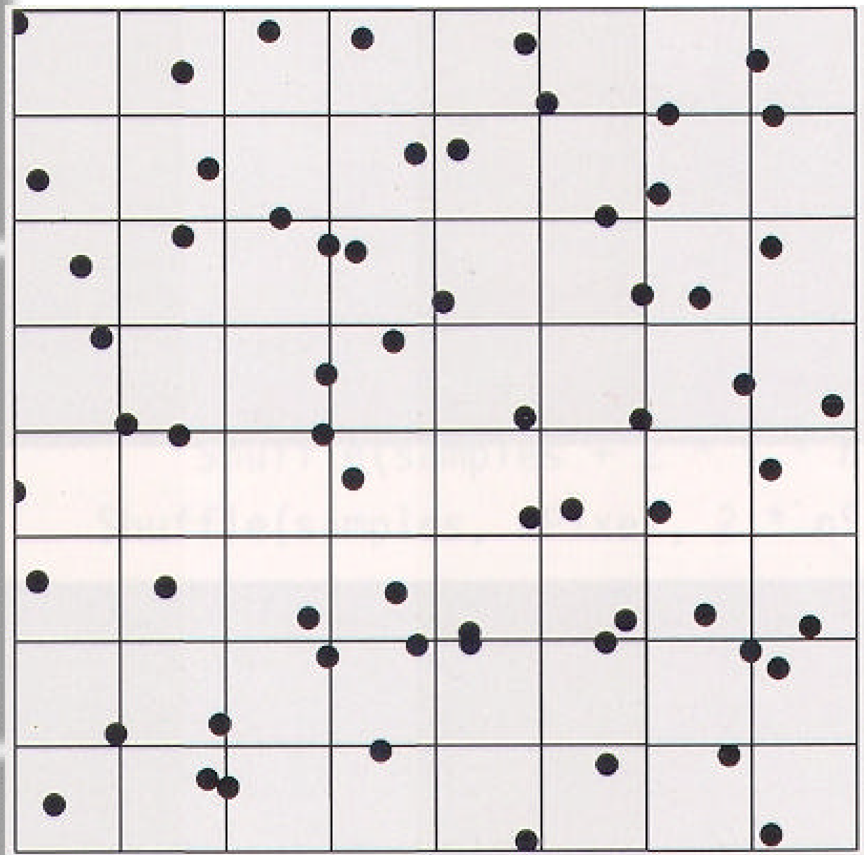
Best Candidate with 1 sample/pixel



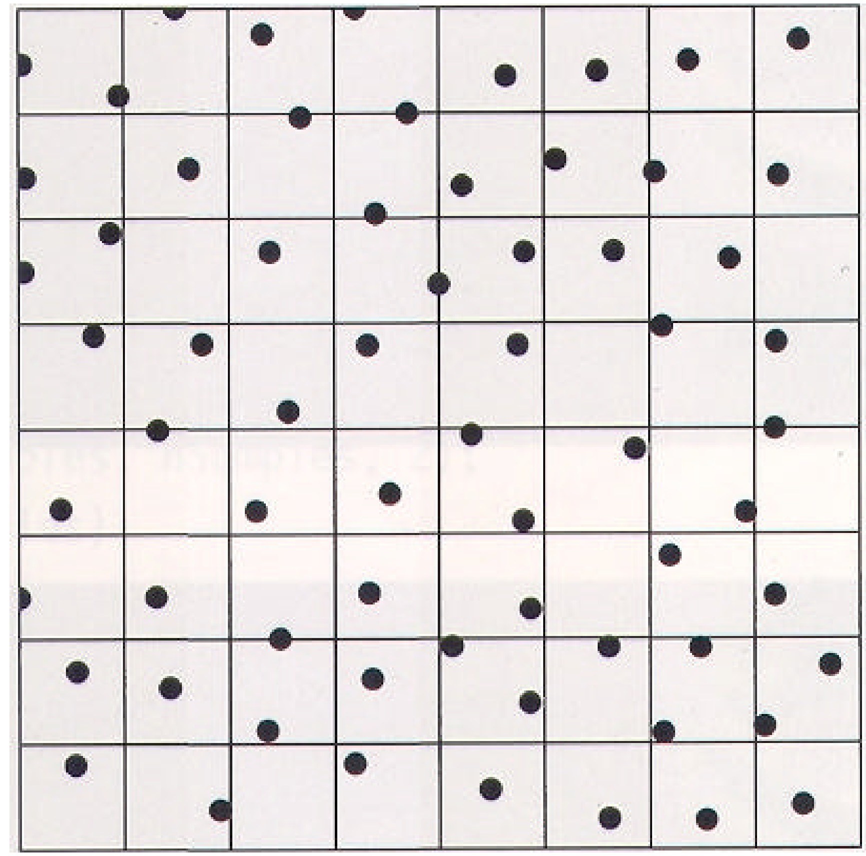
Jitter with 4 sample/pixel

Best Candidate with 4 sample/pixel

Best candidate sampling



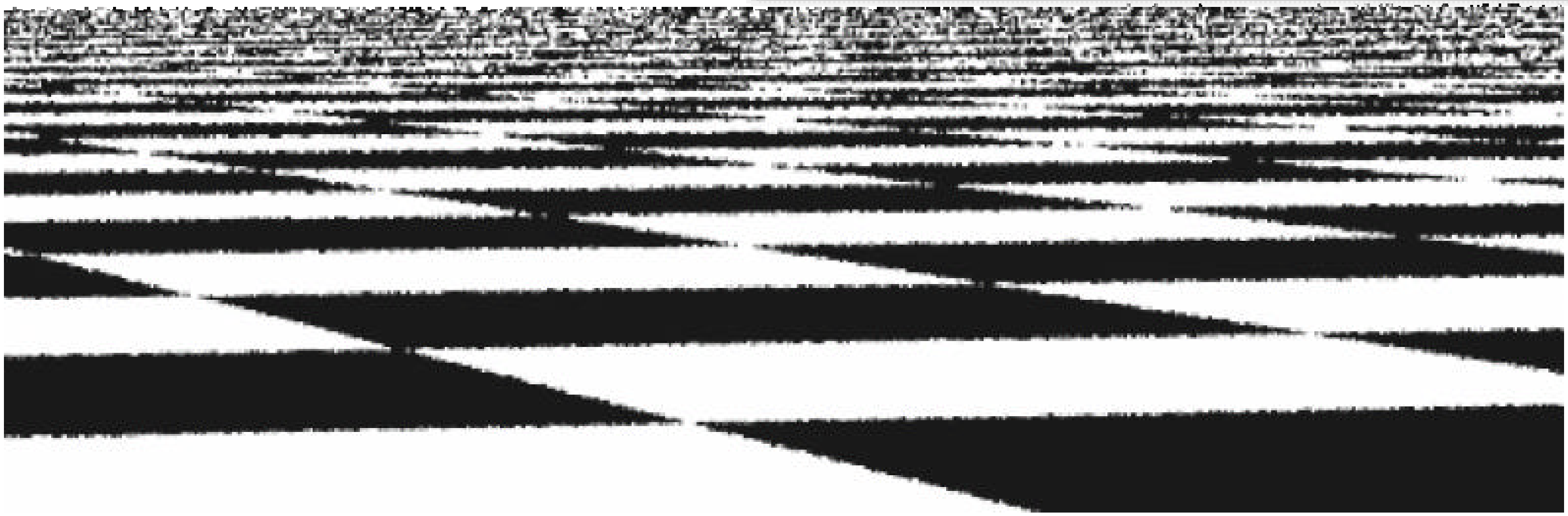
stratified jittered



best candidate

It avoids holes and clusters.

Best candidate sampling



stratified jittered, 1 sample/pixel

best candidate, 1 sample/pixel

Best candidate sampling

stratified jittered, 4 sample/pixel

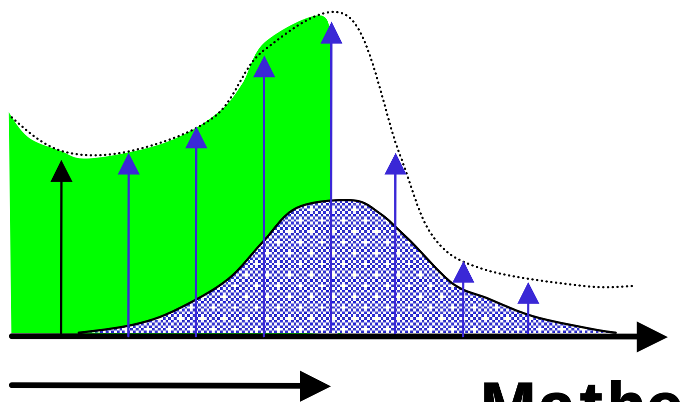
best candidate, 4 sample/pixel

Ideal Reconstruction

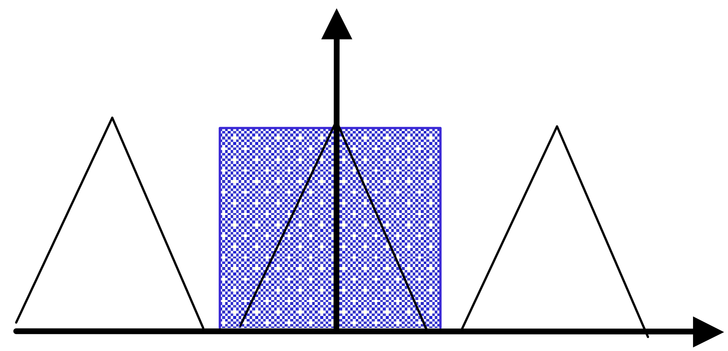
- Ideally, use perfect low-pass filter - the sinc function - to bandlimit the sampled signal
- Thus remove all copies of spectra introduced by sampling
- Unfortunately,
  - The sinc has infinite extent and we must use simpler filters with finite extents. Physical processes in particular do not reconstruct with sincs
  - The sinc may introduce ringing which are perceptually objectionable

How? - Reconstruction

Spatial Domain:



Frequency Domain:



Mathematically:

$f(x)*h(x)$

- Convolution:

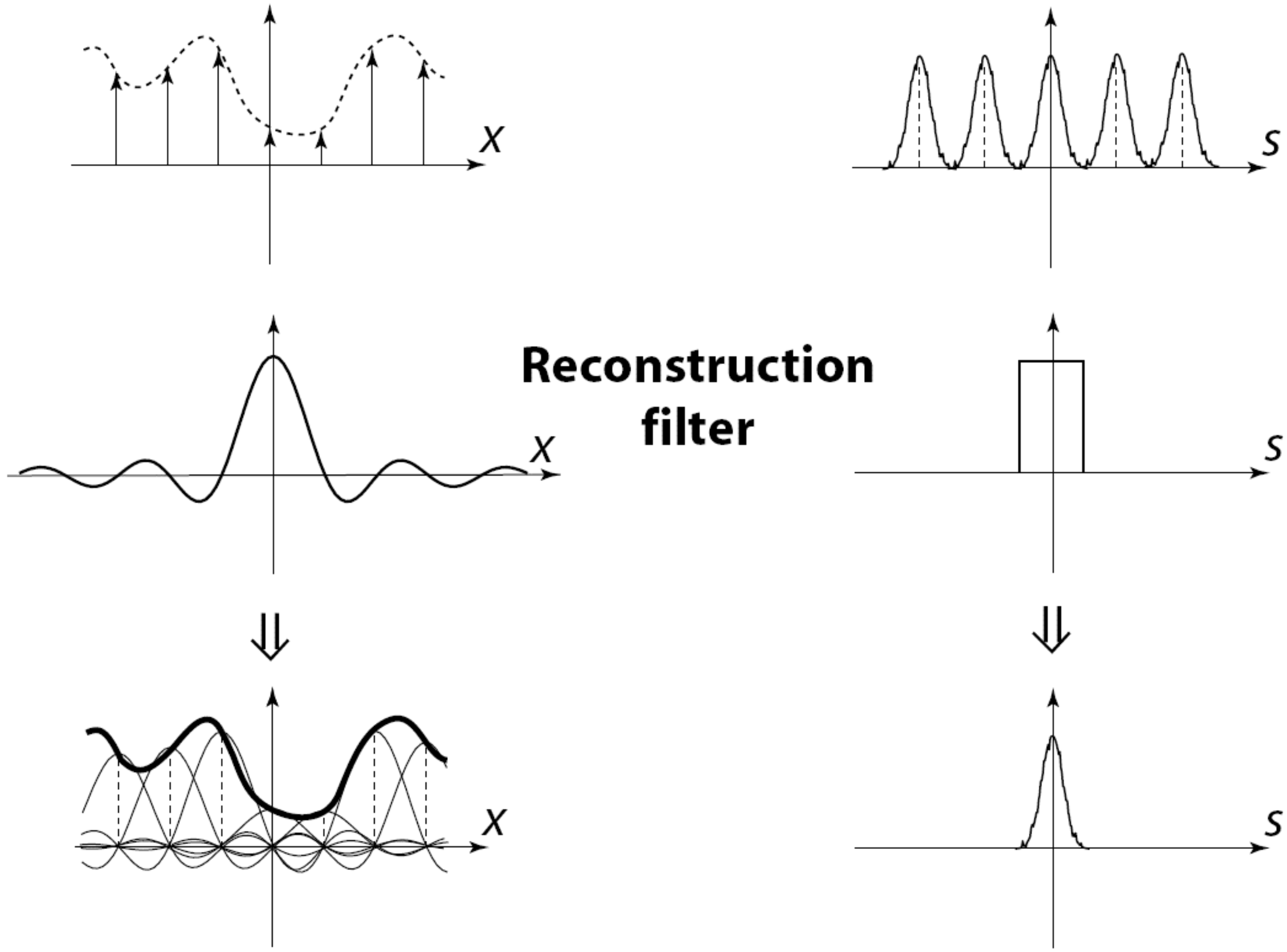
$$\int_{-\infty}^{\infty} f(t) \times h(x-t)\,dt$$

Evaluated at discrete points (sum)

- Multiplication:

$$F(\omega) \times H(\omega)$$

Reconstruction

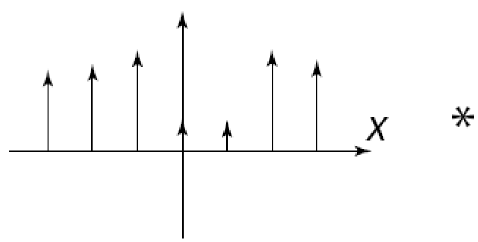


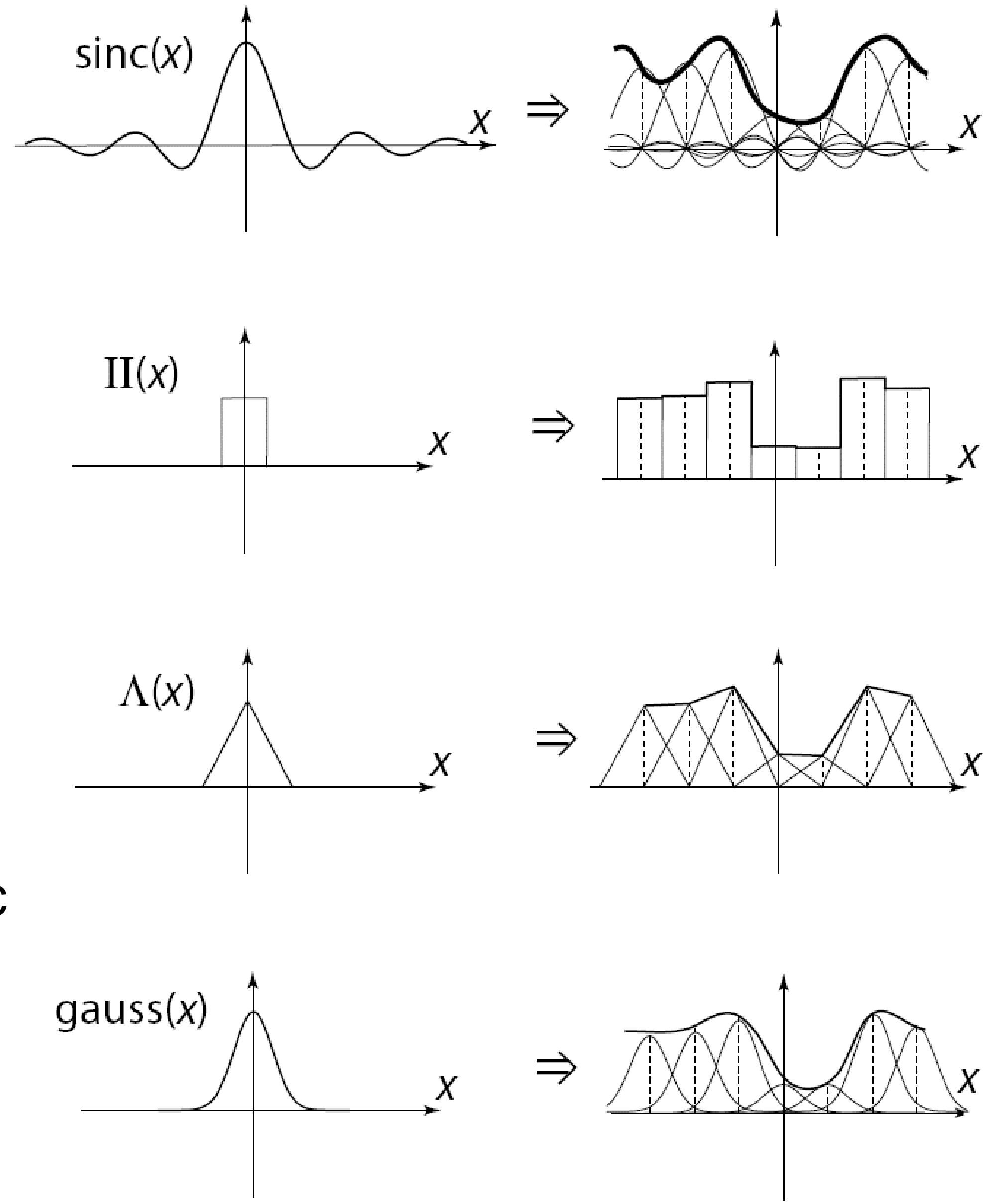
The reconstructed function is obtained by interpolating among the samples in some manner

Reconstruction filters

The sinc filter, while ideal, has two drawbacks:

- It has large support (slow to compute)
- It introduces ringing in practice



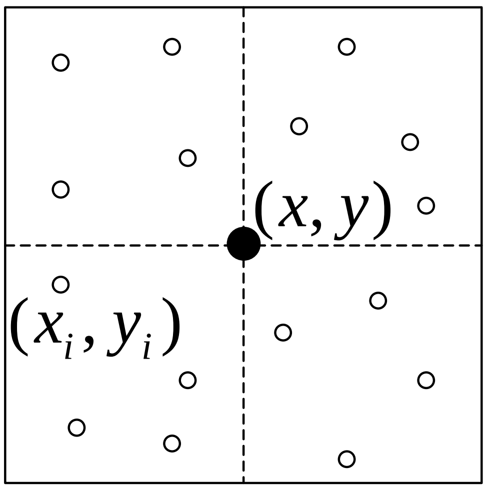


The box filter is bad because its Fourier transform is a sinc filter which includes high frequency contribution from the infinite series of other copies.

Reconstruction filters

- Given image samples, we can do the following to compute pixel values.
  1. reconstruct continuous function L' from samples
  2. prefilter L' to remove frequency higher than Nyquist limit
  3. sample L' at pixel locations
- Instead, we consider an interpolation problem

$$I(x,y)=\frac{\sum_i f(x-x_i,y-y_i)L(x_i,y_i)}{\sum_i f(x-x_i,y-y_i)}$$



Filter

- provides an interface to *f(x,y)*
- **`Film`** stores a pointer to a filter and use it to filter the output before writing it to disk.

width, half of support

```
Filter::Filter(float xw, float yw)
Float Evaluate(float x, float y);
```

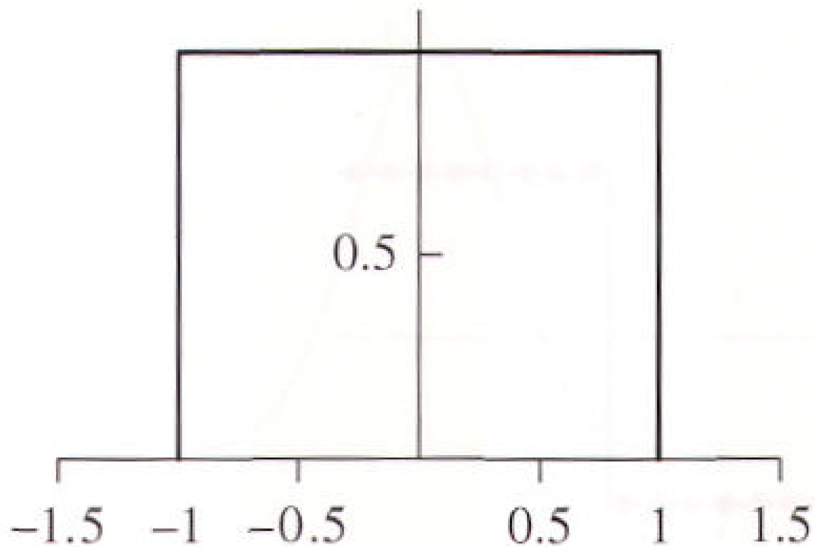
- **`filters/*`**

Box filter

- Most commonly used in graphics. It’s just about the worst filter possible, incurring postaliasing by high-frequency leakage.

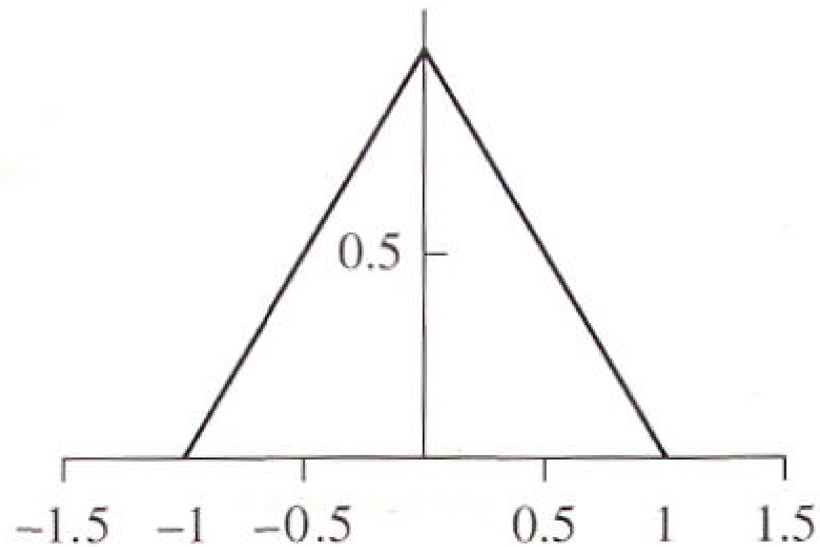
```
Float BoxFilter::Evaluate(float x, float
  y)
{
  return 1.;
}
```

▪**Note:** input always in Range **-1 to 1**



Triangle filter

```
Float TriangleFilter::Evaluate(float x,
  float y)
{
  return max(0.f, xWidth-fabsf(x)) *
         max(0.f, yWidth-fabsf(y));
}
```



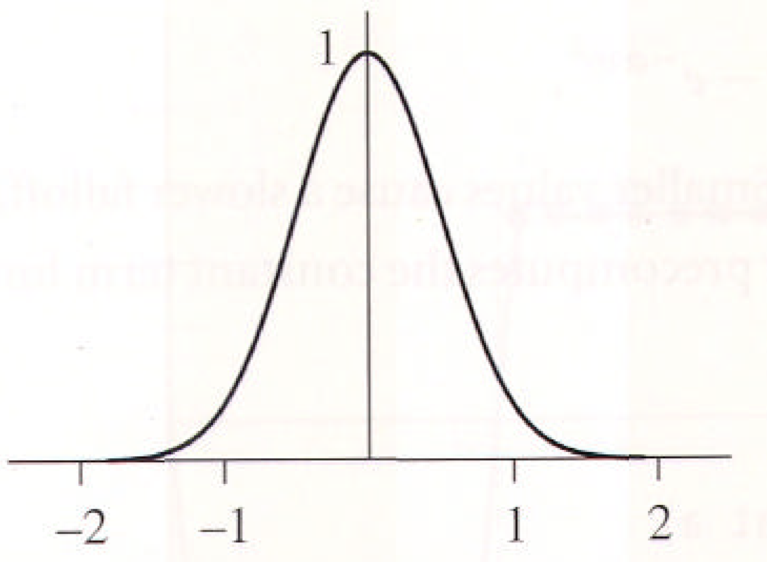
Gaussian filter

- Gives reasonably good results in practice

```
Float GaussianFilter::Evaluate(float x,
  float y)
{
  return Gaussian(x, expX)*Gaussian(y, expY);
}
```

1D Gaussian filter:

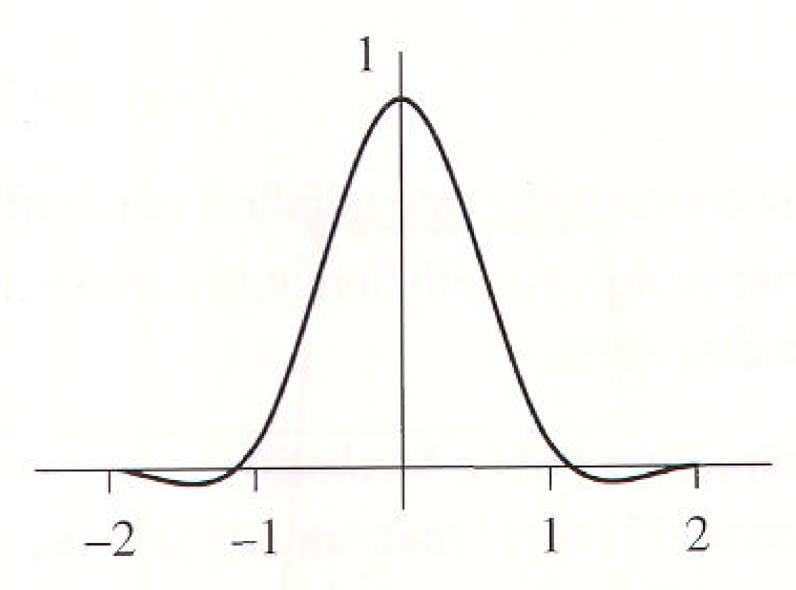
$$f(x) = e^{-ax^2} - e^{-aw}$$



Mitchell filter

- parametric filters, tradeoff between ringing and blurring

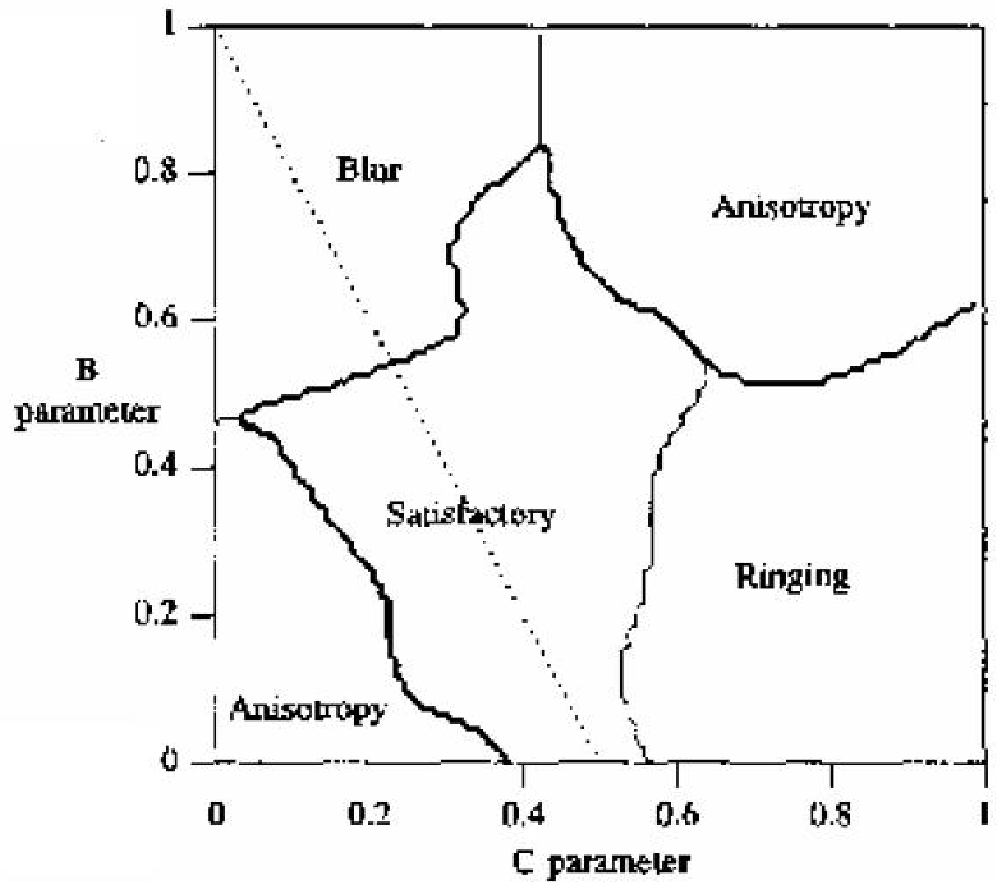
- Negative lobes



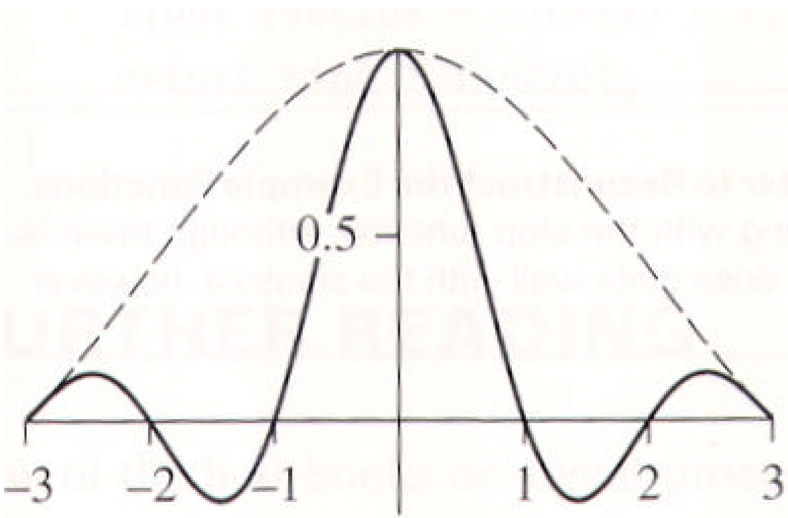
Mitchell filter

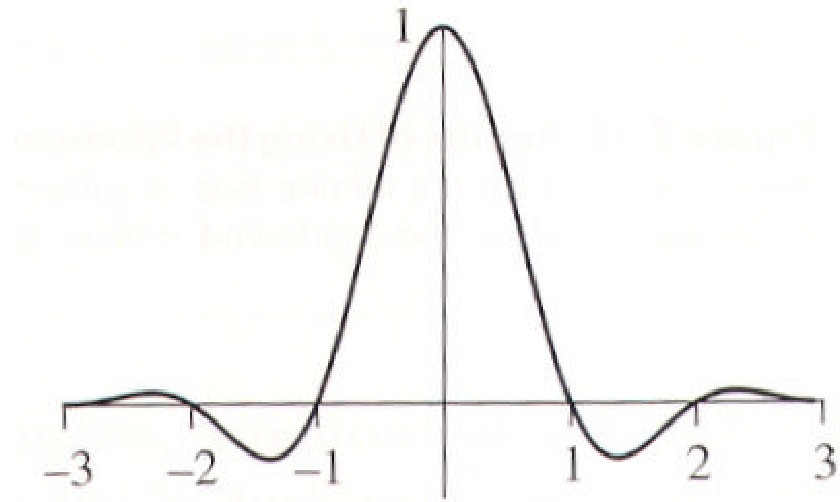
$$h(x) = \frac{1}{6}\begin{cases} (12-9B-6C)x^3 + (-18+12B+6C)x^2 + (6-2B) & |x| < 1 \\ (-B-6C)x^3 + (6B+30C)x^2 + (-12B-48C)x + (8B+24C) & 1 < |x| < 2 \\ 0 & otherwise \end{cases}$$

- Separable filter
- Two parameters, B and C, B+2C=1 suggested



Windowed sinc filter





$$w(x) = \frac{\sin px / t}{px / t}$$

References

- Yung-Yu Chuang, Image Synthesis, class slides, National Taiwan University, Fall 2005
- Rick Parent, 782: Advanced 3D Image Generation
- Pat Hanrahan, CS 348B, Spring 2005 class slides