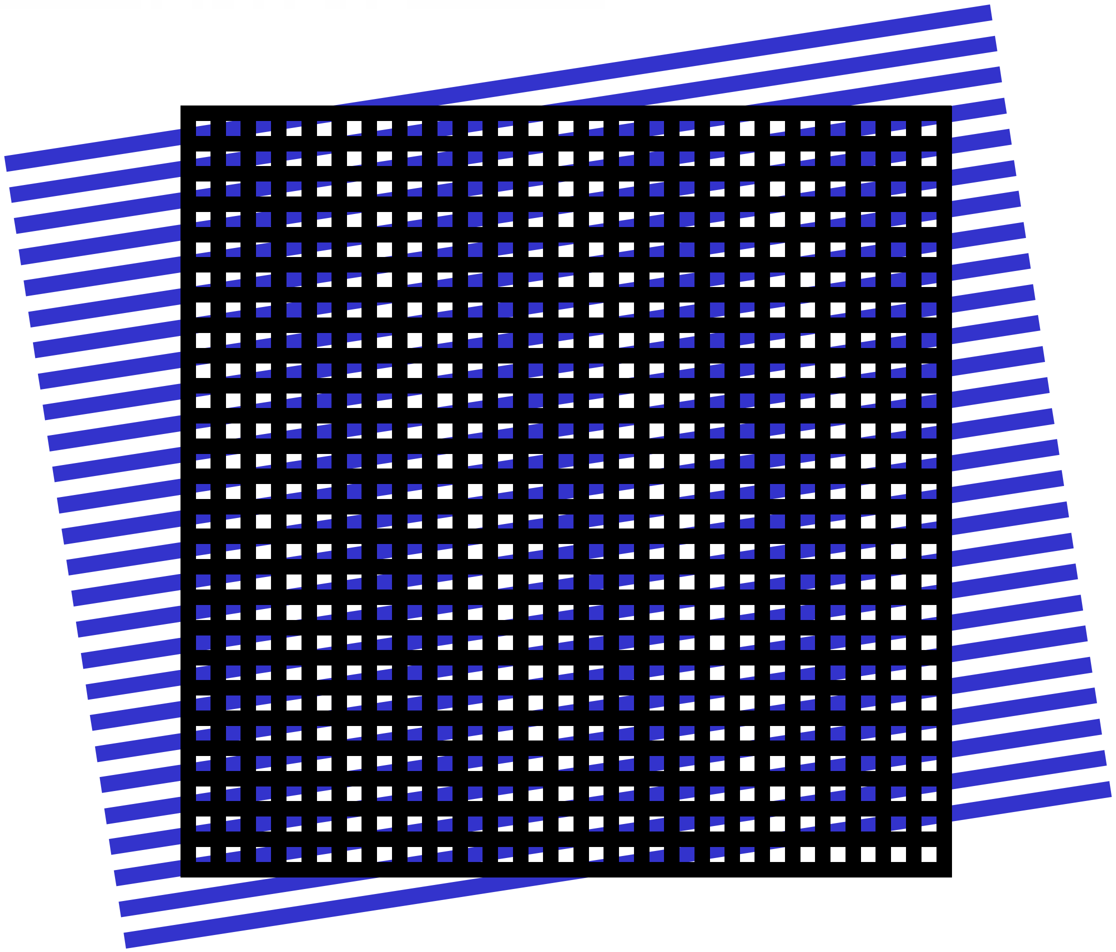
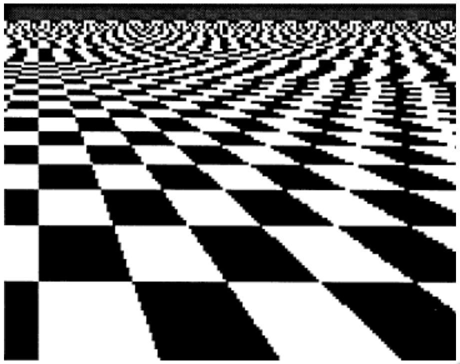

8. Anti Aliasing

Motivation

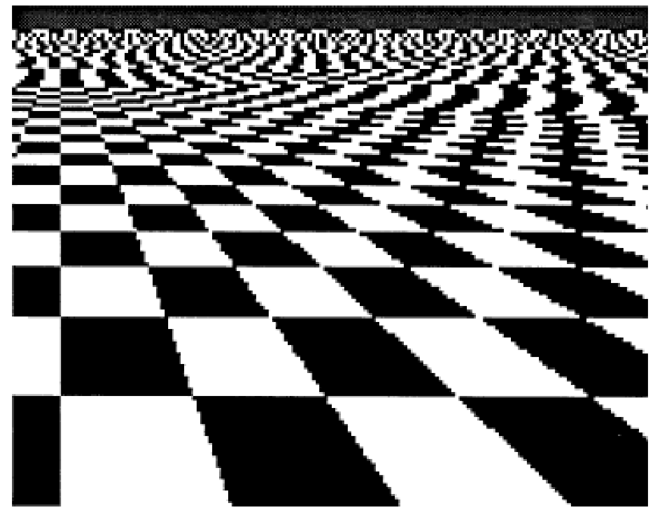
- The main goal of Computer Graphics is to generate 2D images
- 2D images are continuous 2D functions (or signals)
  - monochrome $f(x,y)$
  - or color $r(x,y)$, $g(x,y)$, $b(x,y)$
- These functions are represented by a 2D set of discrete samples (pixels)
- Sampling can cause artifacts (=Aliasing)

Examples - Moiré Patterns



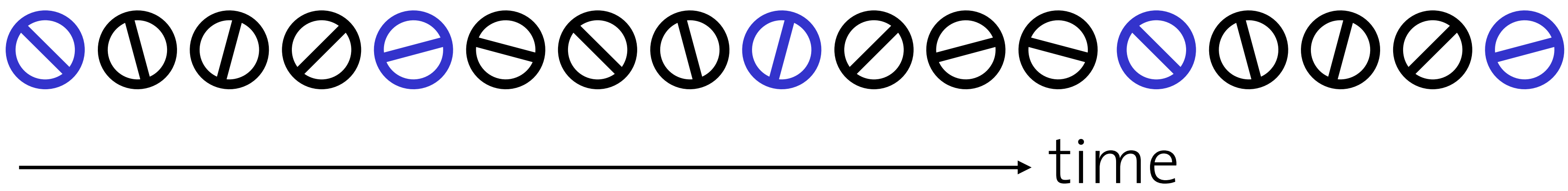


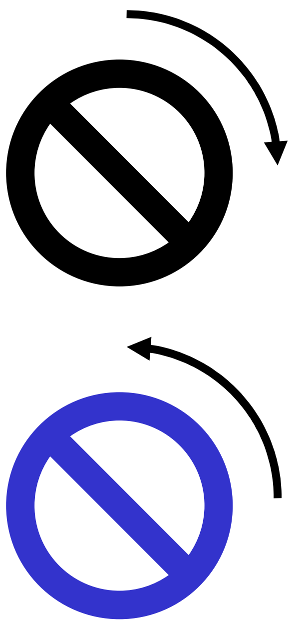
Examples - Jaggies



- Staircase effect at borders

Temporal Aliasing





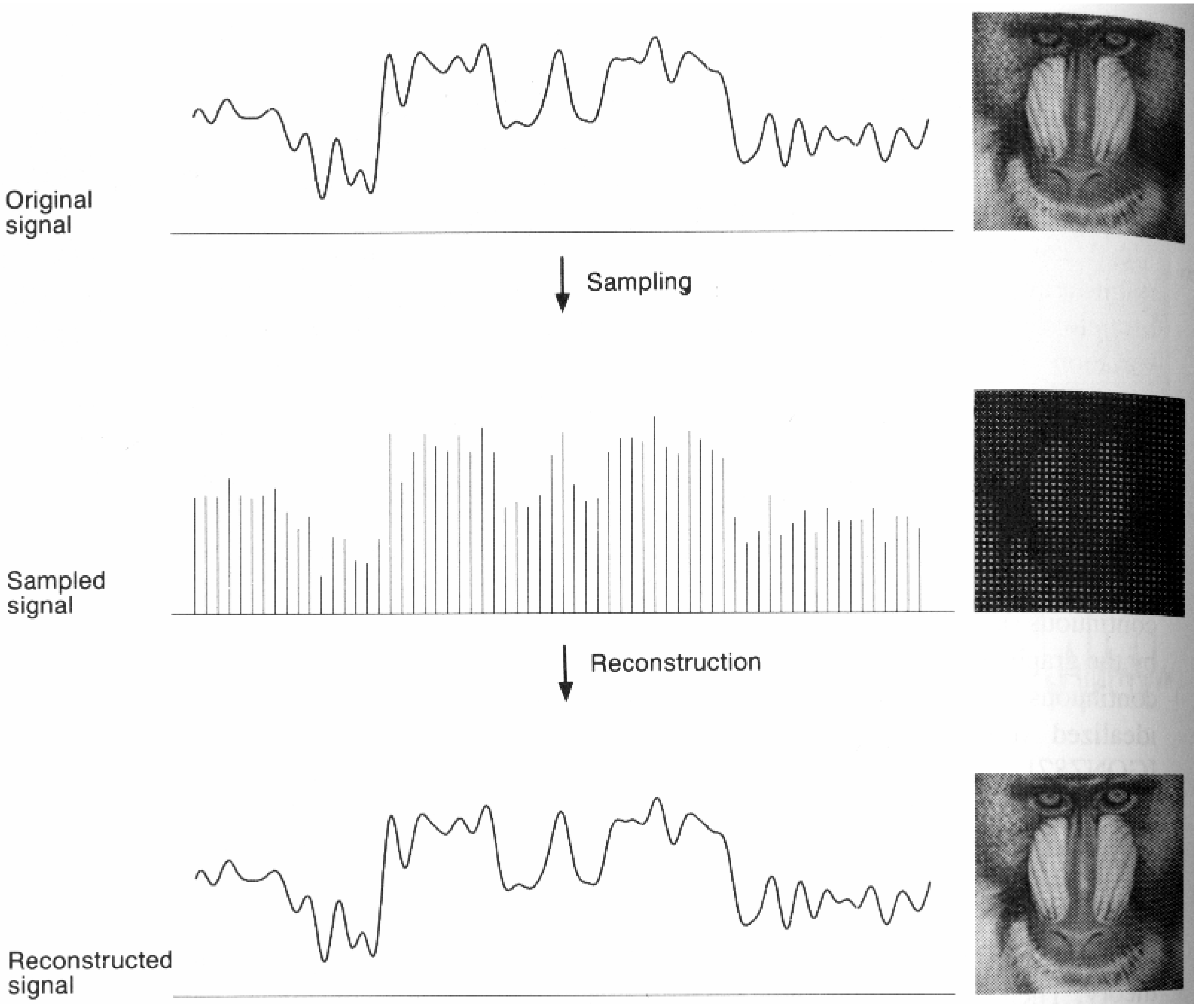
real (continuous) motion

sampled (perceived) motion

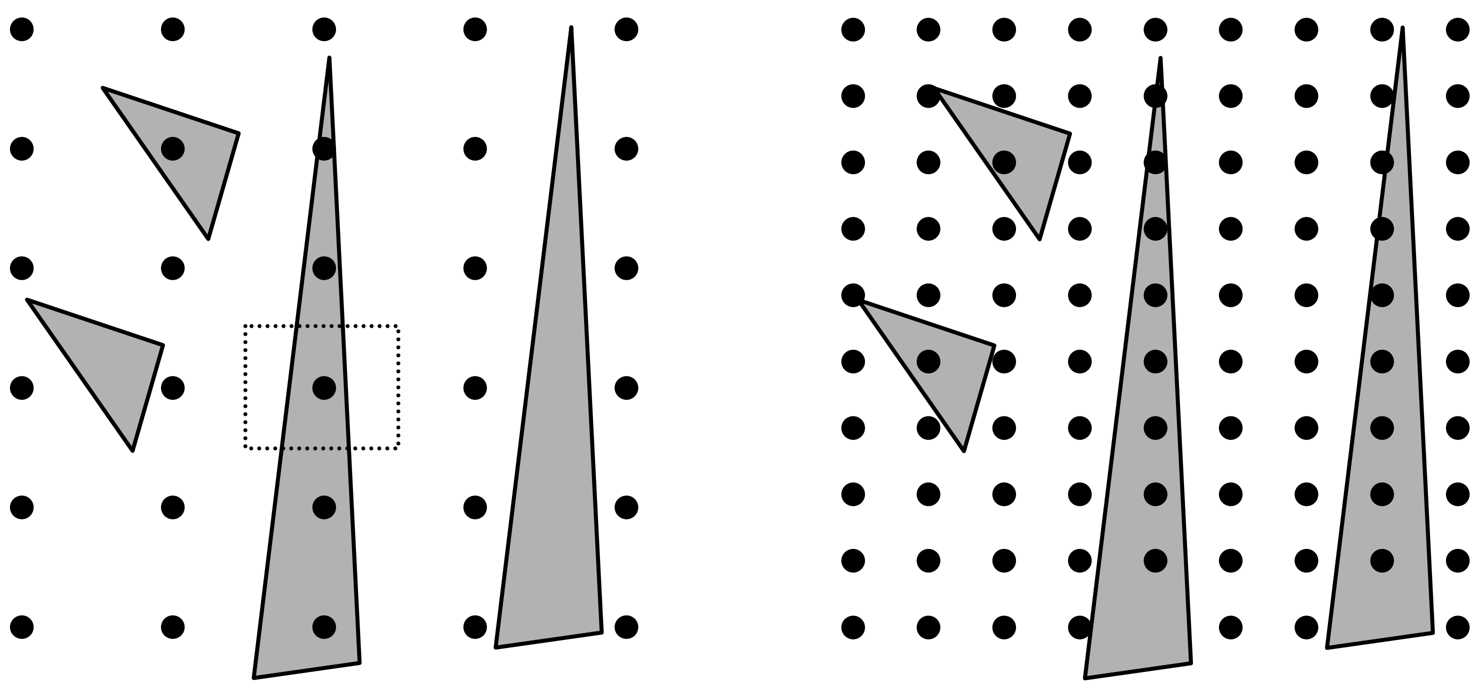
Aliasing in Computer Graphics

- Aliasing effects:
  - loss of detail
  - Moiré patterns
  - jaggies
- Appear in
  - texture mapping
  - scan conversion of geometry
  - raytracing

Sampling and Reconstruction

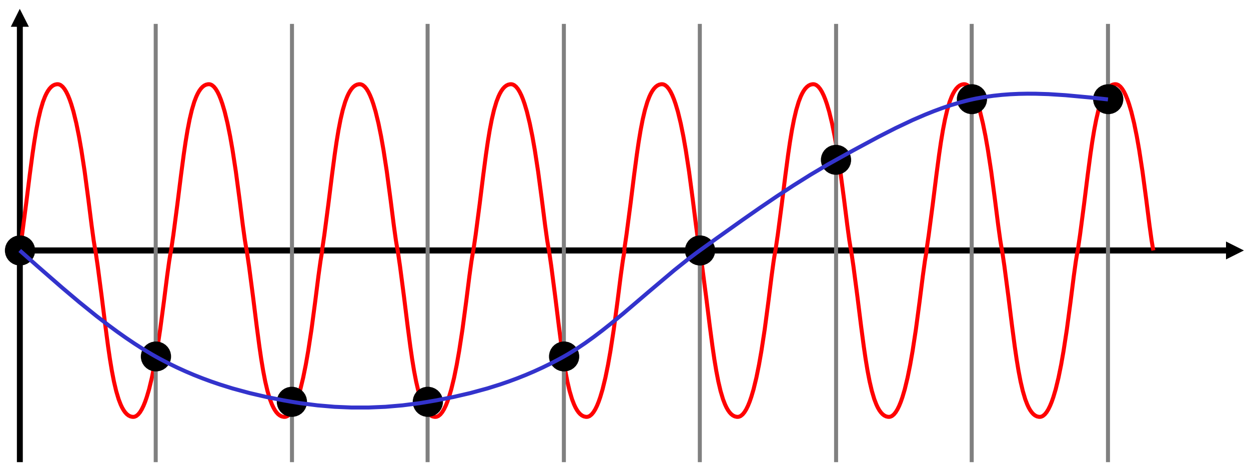


Example – Point Sampling



Signal Processing

- Aliasing is well understood in signal processing
- Interpret images as 2D signals
- Aliasing = sampling of L^2-functions below the Nyquist frequency $u_{Nyquist} = 2\, u_{signal}$



Spectrum of an Image

- What is u_{signal} of an image $f(x,y)$?
- Use Fourier analysis (1D first)
- Represent $f(x)$ as a sum of harmonic waves:

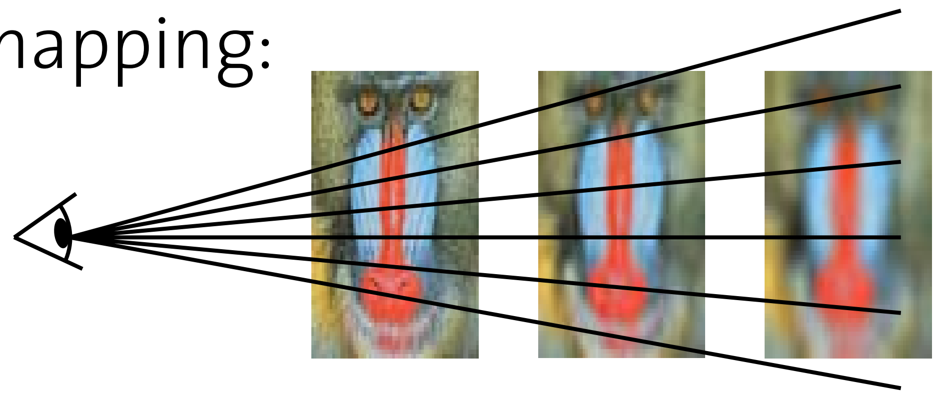
$$f(x) = \int_{-\infty}^{\infty} F(u)\, e^{j2\pi u x}\, du$$

- The amplitudes $F(u)$ of waves with frequency u (spectrum) are computed as

$$F\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-j2\pi u x}\, dx$$

Avoiding Aliasing

- Let W be the maximum u for which $|F(u)|>0$
- Either choose $u_{sampling} > 2W$
- Or zero all $F(u)$ for $u > \frac{1}{2}\, u_{sampling}$
- i.e. low pass filter the signal
- Smoothing of image before sampling!
- e.g. Mip mapping:



decreasing sampling rate, increased smoothing

1D Fourier Transform

- Fourier transform

$$F\left\{f(x)\right\} = F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-j2\pi u x}\, dx$$

- Inverse transform

$$F^{-1}\left\{F(u)\right\} = f(x) = \int_{-\infty}^{\infty} F(u)\, e^{j2\pi u x}\, du$$

1D Discrete Fourier Transform

- Discrete transform $F(k) = \frac{1}{N} \sum_{i=0}^{N-1} f(i)\, e^{-\frac{j 2\pi k i}{N}}$

- Discrete inverse $f(i) = \sum_{k=0}^{N-1} F(k)\, e^{\frac{j 2\pi k i}{N}}$

$$x = i \cdot \Delta x,$$
$$u = k \cdot \Delta u$$

- Heisenberg resolution bounds $\Delta x \cdot \Delta u \geq \frac{1}{4\pi}$

2D Fourier Transforms

$$F\left\{f(x,y)\right\} = F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-j2\pi(ux+vy)}\, dx\, dy$$

$$F^{-1}\left\{F(u,v)\right\} = f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{j2\pi(ux+vy)}\, du\, dv$$

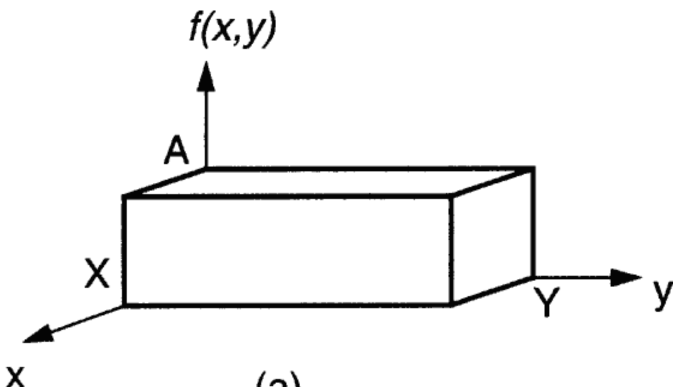
- Discrete setting

$$F(u,v) = \frac{1}{M\,N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

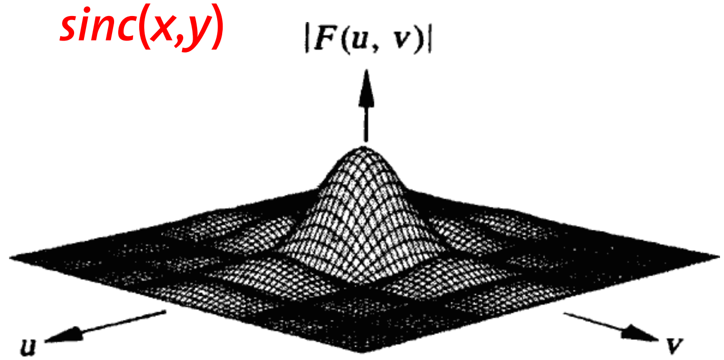
Example: 2D Fourier Transforms

rect(x,y)

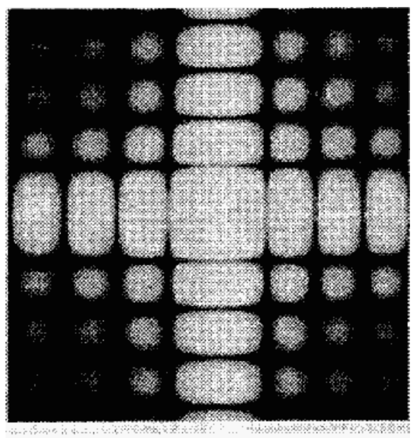


(a)

sinc(x,y)



(b)

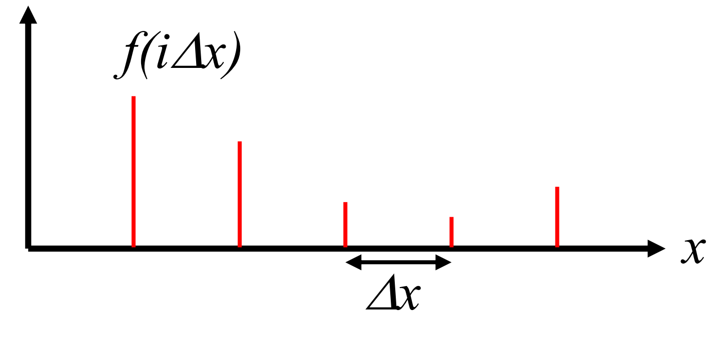


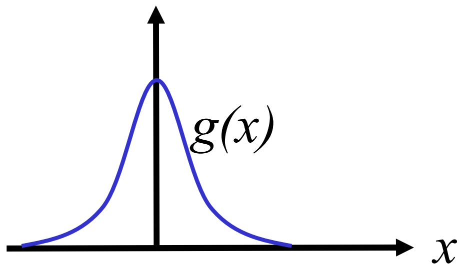
(c)

sine cardinal:

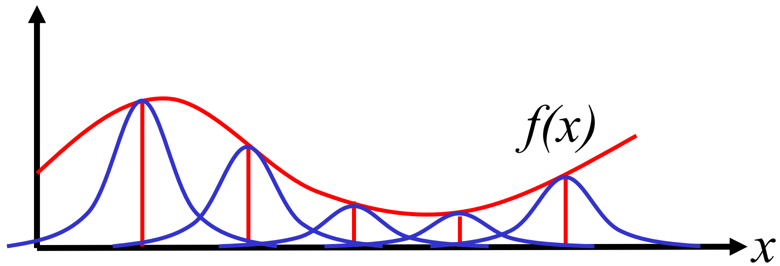
$$\text{sinc}(x) = \begin{cases} 1 & x = 0 \\ \sin(x)/x & \text{otherwise} \end{cases}$$

Reconstruction





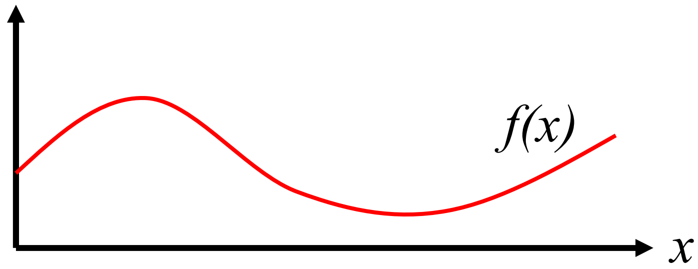
reconstruction filter

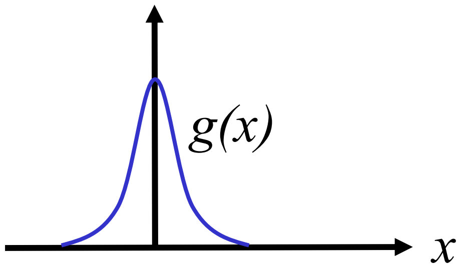


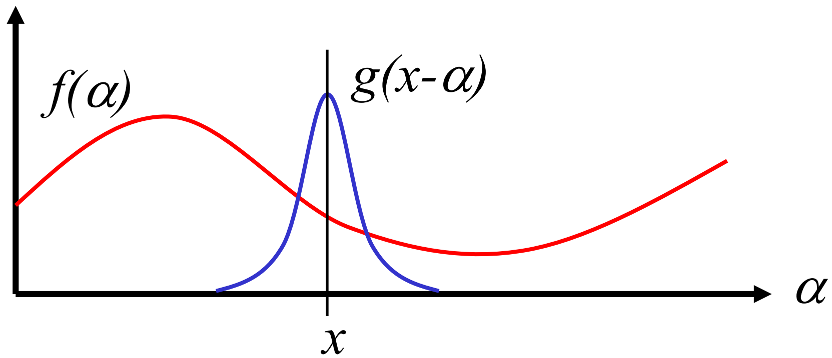
$$f(x) = \sum_{i=1}^{N} f(i\Delta x) \cdot g(x - i\Delta x) \cdot \Delta x$$

Convolutions

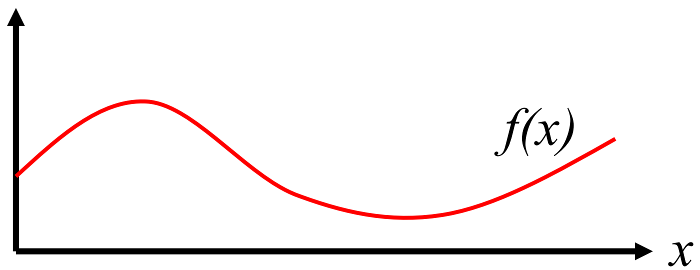
$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha)\, d\alpha$$

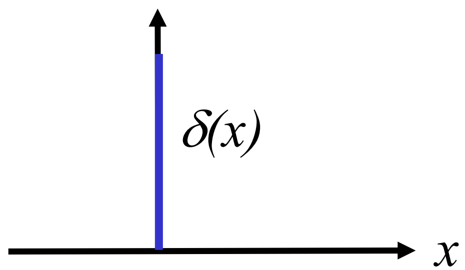






Convolutions





$$f(x) * \delta(x) = \int_{-\infty}^{\infty} f(\alpha)\delta(x-\alpha)\, d\alpha = f(x)$$

- Discrete setting

$$f(x) * g(x) = \sum_{m=0}^{M-1} f(m)\, g(x-m)$$

Convolutions

- 2D convolution as a separable TP-extension

$$f(x,y) * g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta)\, g(x-\alpha, y-\beta)\, d\alpha\, d\beta$$

- Discrete form

$$f(x,y) * g(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\, g(x-m, y-n)$$

Convolutions

- Convolution theorem

$$f(x) * g(x) \equiv F(u)\,G(u)$$

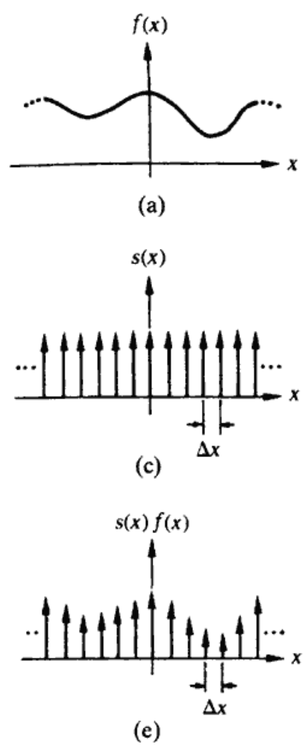
$$f(x)g(x) \equiv F(u) * G(u)$$

- For function of ***finite energy*** (L^2)

$$\|f\|^2 = \langle f,f \rangle = \int_{-\infty}^{\infty} |f(x)|^2\,dx < \infty$$

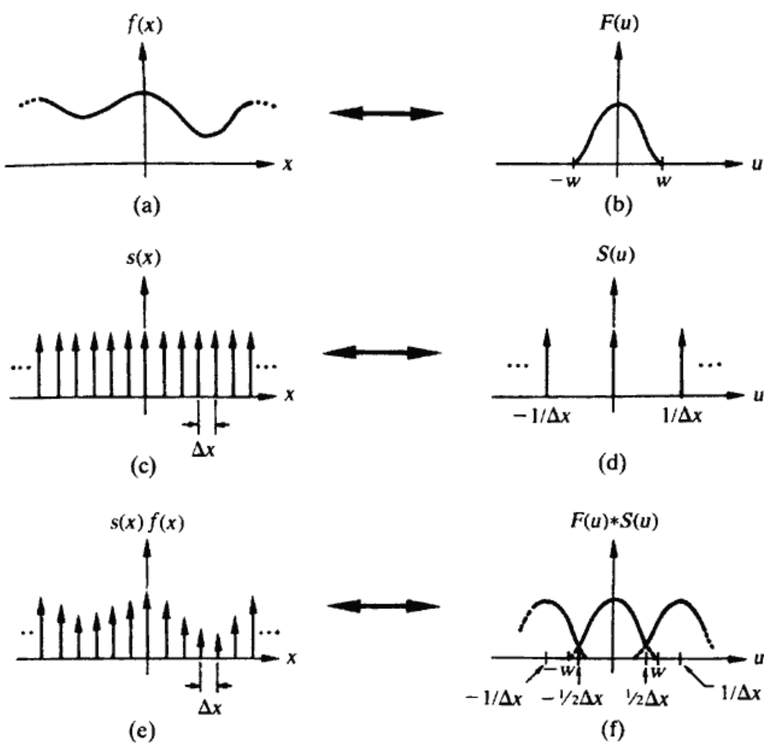
Aliasing

- Sampling = multiplication with sequence of delta functions (impulse train)



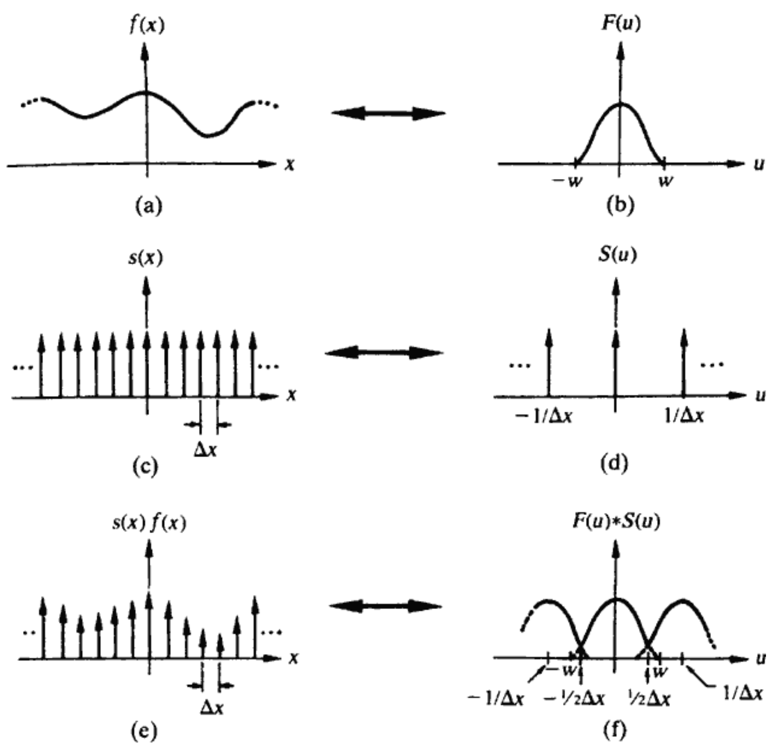
Aliasing

- Multiplication converts to convolution in Fourier domain



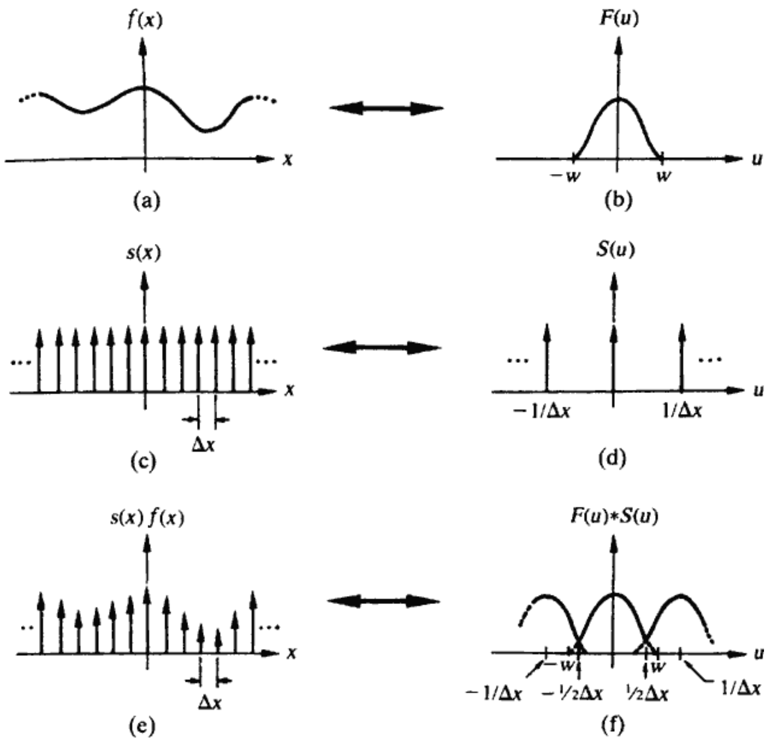
Aliasing

- Convolution with sequence of delta functions
  = periodization



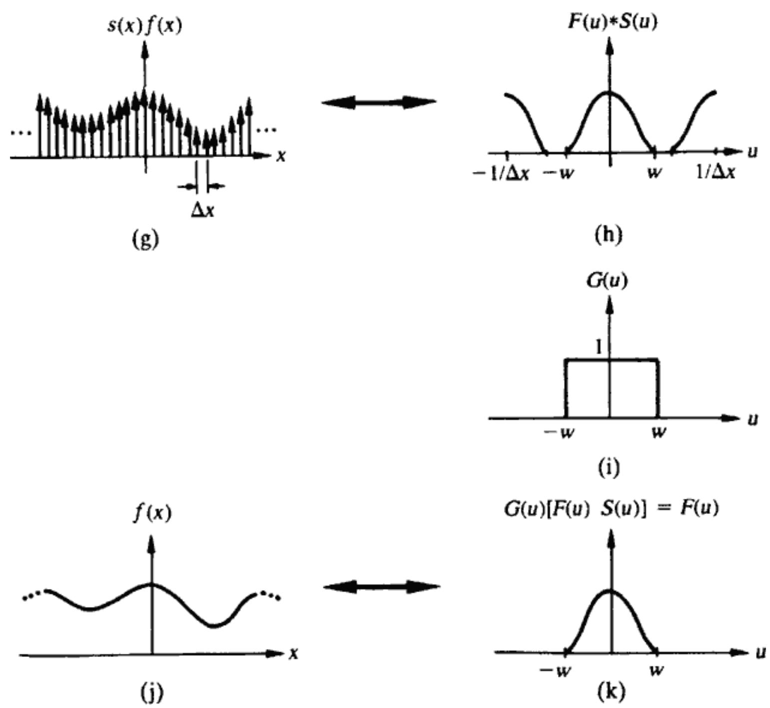
Aliasing

- Overlap of Fourier transforms leads to aliasing



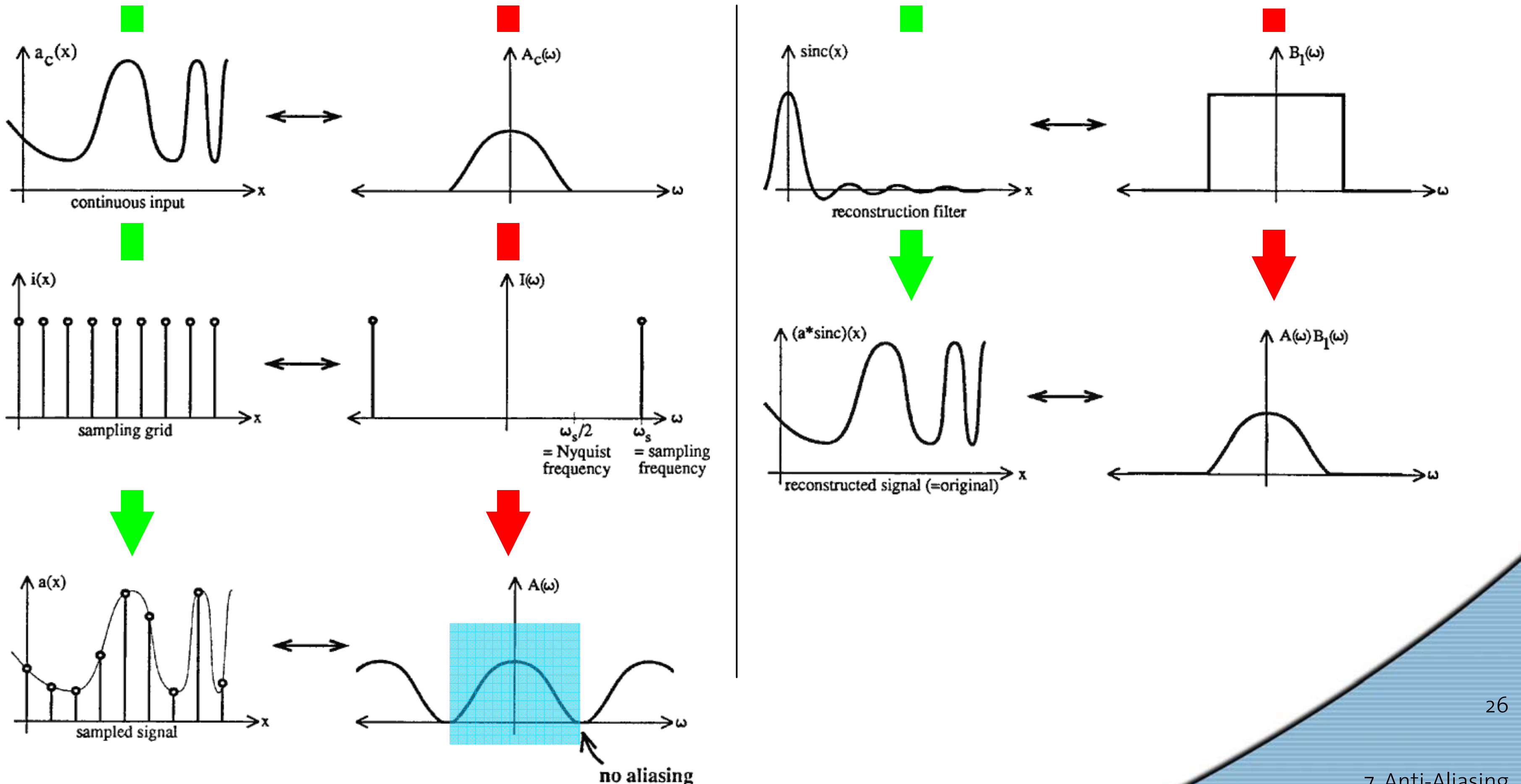
Aliasing

- Reconstruction = Low pass filtering



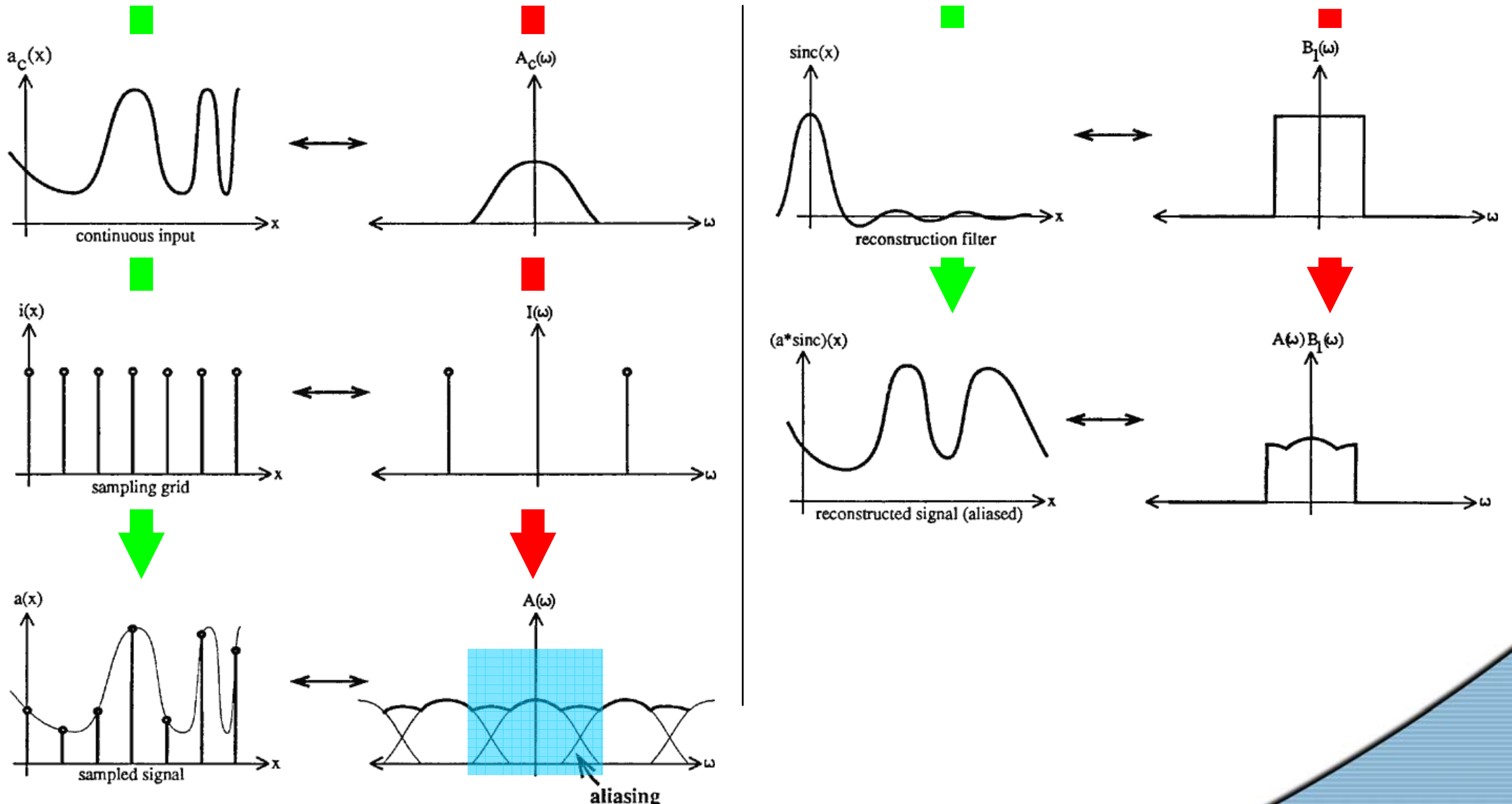
Aliasing-free Reconstruction

Spatial Domain Frequency Domain Spatial Domain Frequency Domain



Occurrence of Aliasing

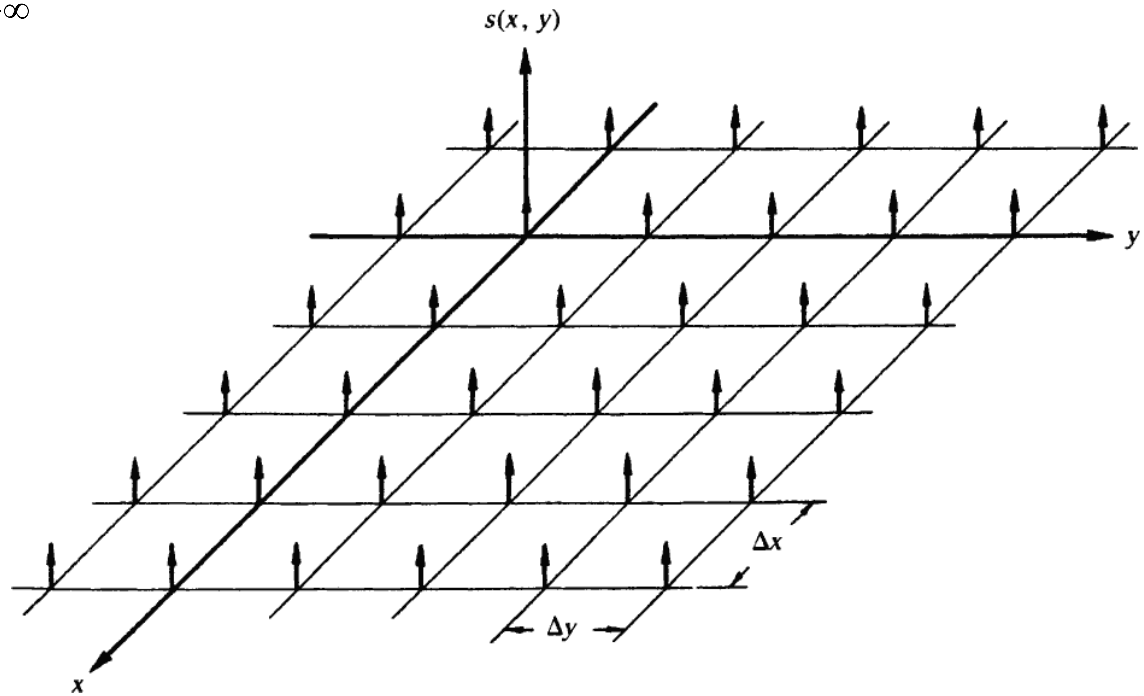
Spatial Domain | Frequency Domain | Spatial Domain | Frequency Domain



2D Sampling

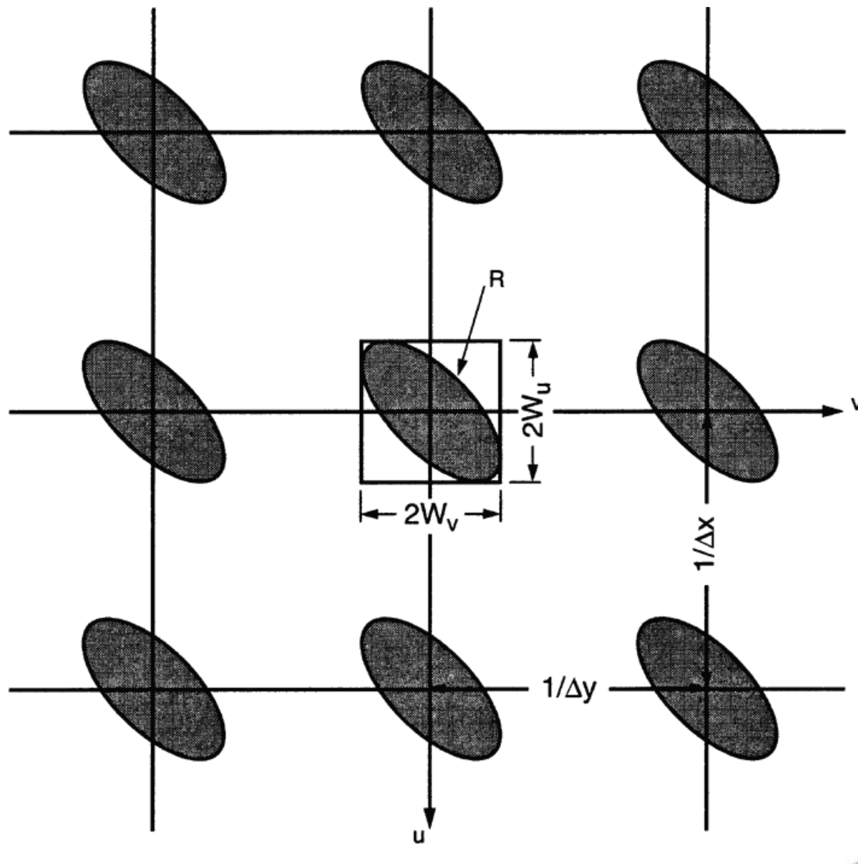
- 2D impulse fields

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(x-x_0,y-y_0)\,dx\,dy = f(x_0,y_0)$$



Fourier Domain

- Periodic spectrum of band limited sampled function



Reconstruction – Antialiasing

- Windowing spectrum using filters
- Simple

$$f(x,y) = G(u,v)\left[S(u,v) * F(u,v)\right]$$

where

$$G(u,v) = \begin{cases} 1 & (u,v) \text{ within Bounding Box of } R \\ 0 & \text{else} \end{cases}$$

2D Sampling Theorem

- Sampling rate is bounded by

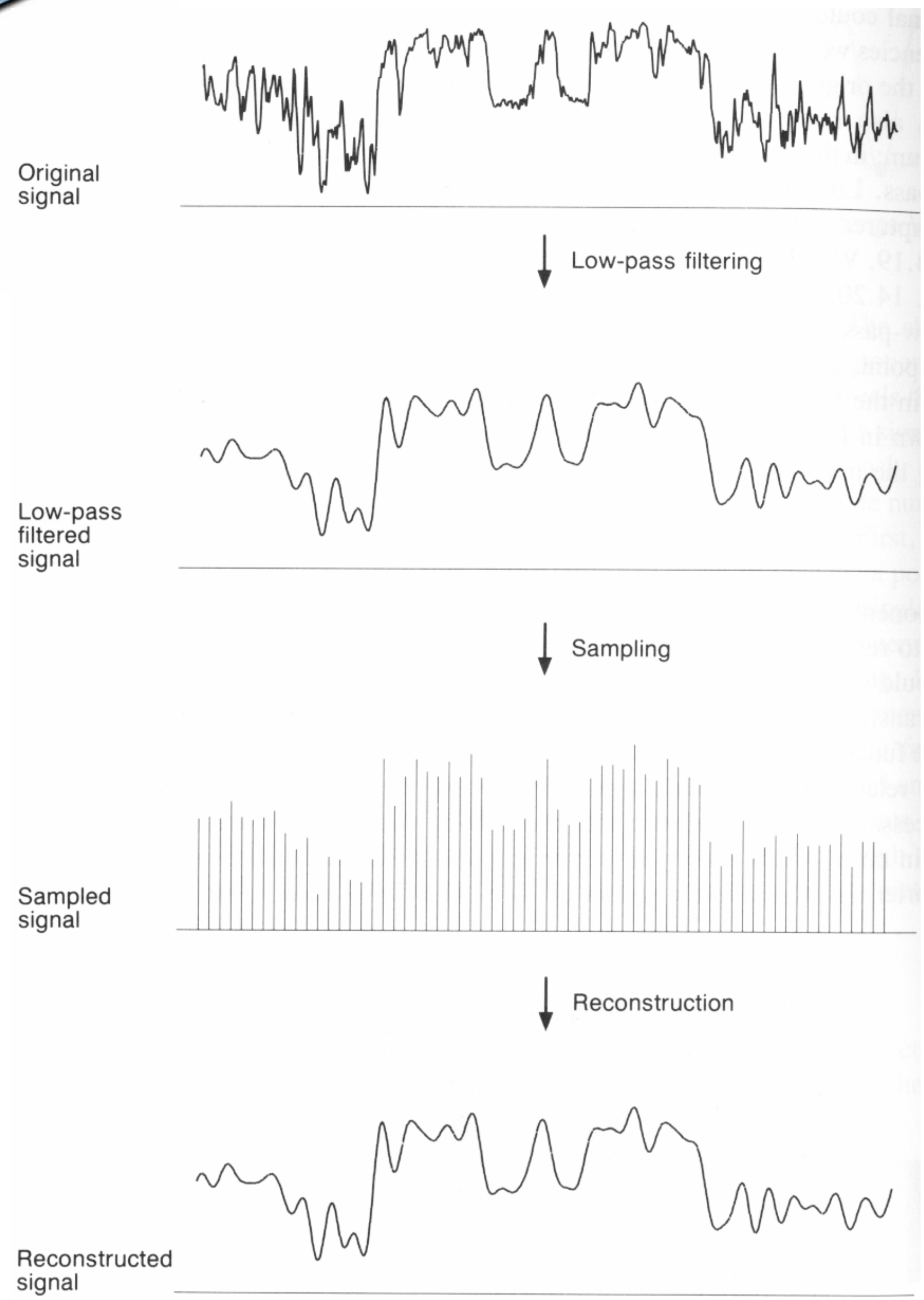
$$\Delta u = \frac{1}{N\,\Delta x}$$

$$\Delta v = \frac{1}{N\,\Delta y}$$

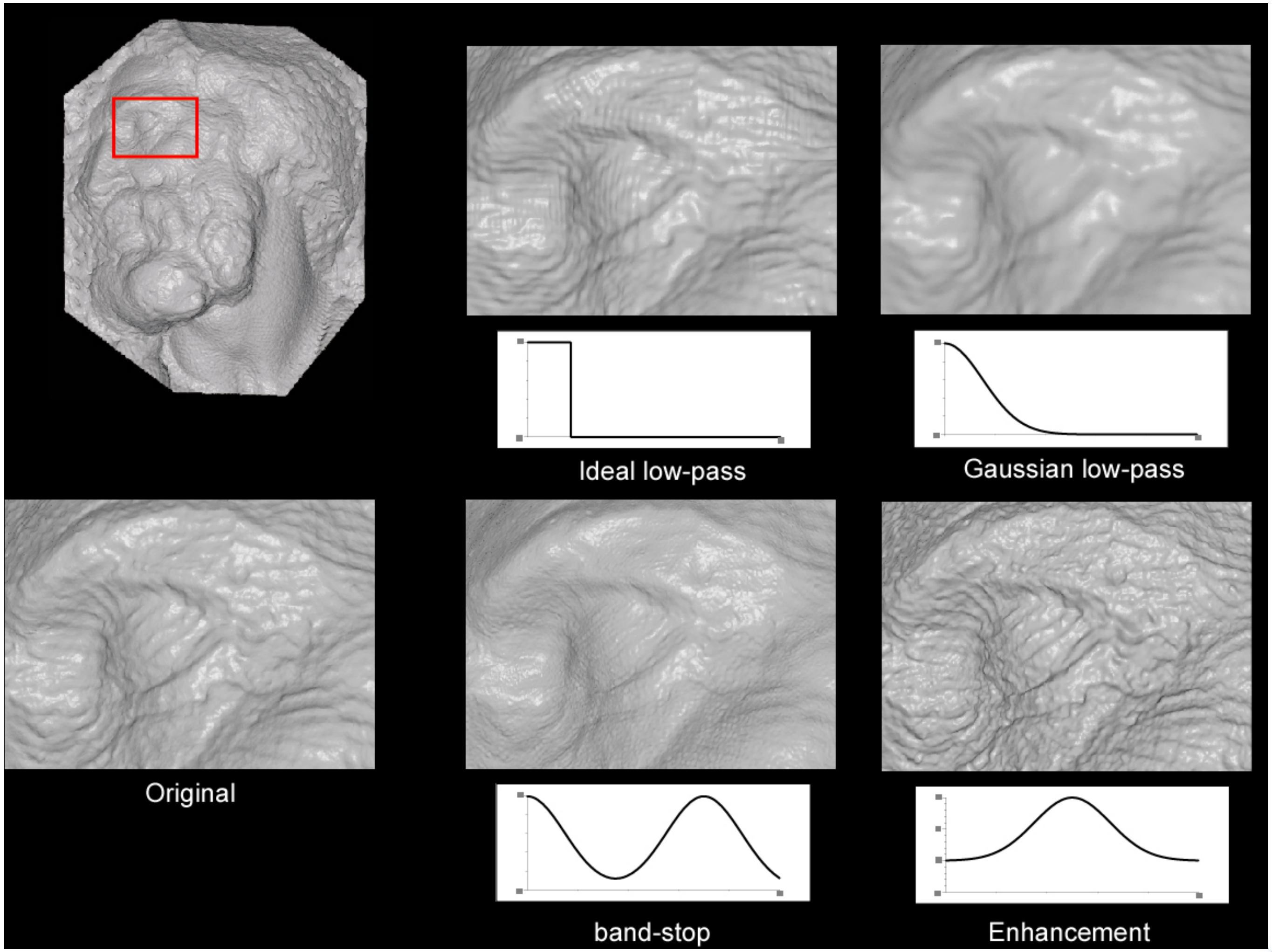
- Finite, discrete setting

$$\Delta x \leq \frac{1}{2W_u}$$

$$\Delta y \leq \frac{1}{2W_v}$$

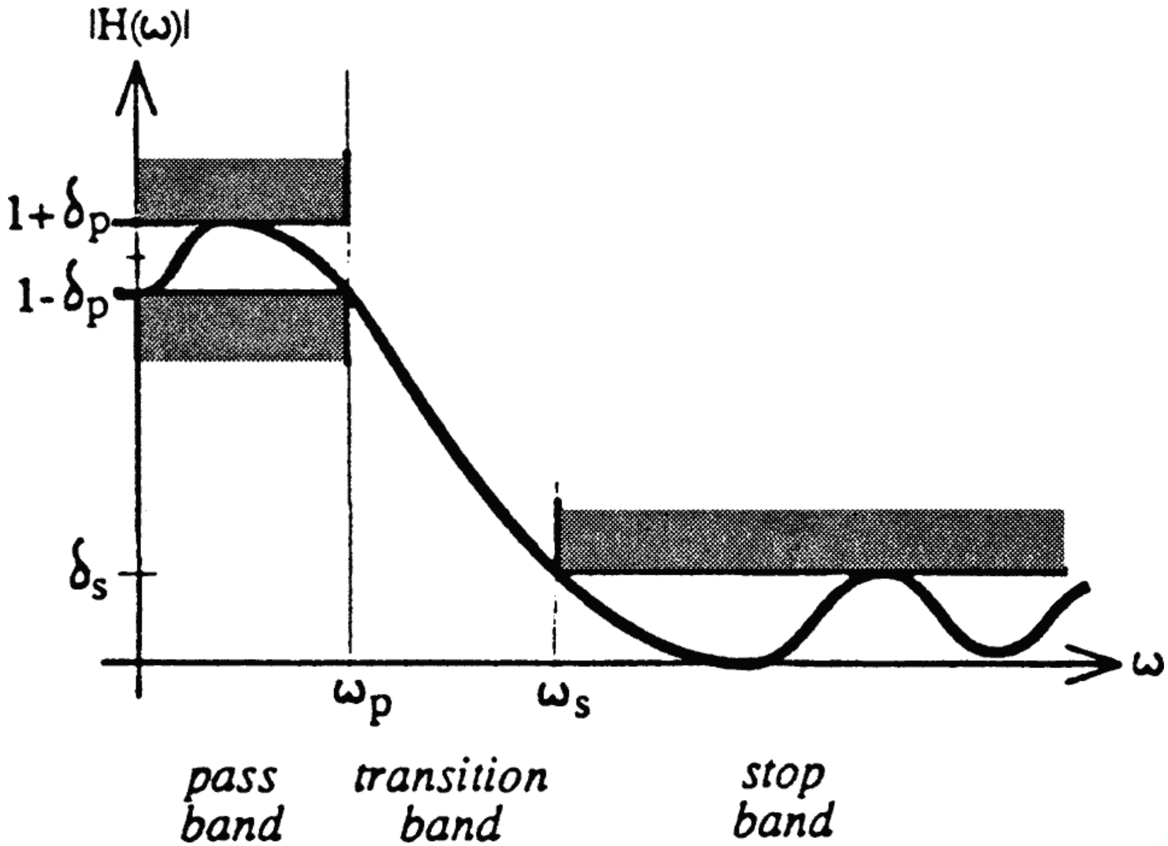

Original signal
Low-pass filtering
Low-pass filtered signal
Sampling
Sampled signal
Reconstruction
Reconstructed signal

Spectral Analysis Geometry



Antialiasing Filters in Practice

- Properties of a good low pass filter



Antialiasing Filters

- B-Spline filters of order n

$$g_1(x) = \begin{cases} 1 & |x| \le \frac{1}{2} \\ 0 & |x| > \frac{1}{2} \end{cases} \quad \leftrightarrow \quad \frac{\sin \omega/2}{\omega/2} = \frac{\sin \pi f}{\pi f} = \sin c\, f$$

- Increase order by repeated convolution

$$g_n(x) = g_1(x) * g_1(x) * \ldots * g_1(x) \quad \leftrightarrow \quad \sin c^n f$$

Antialiasing Filters

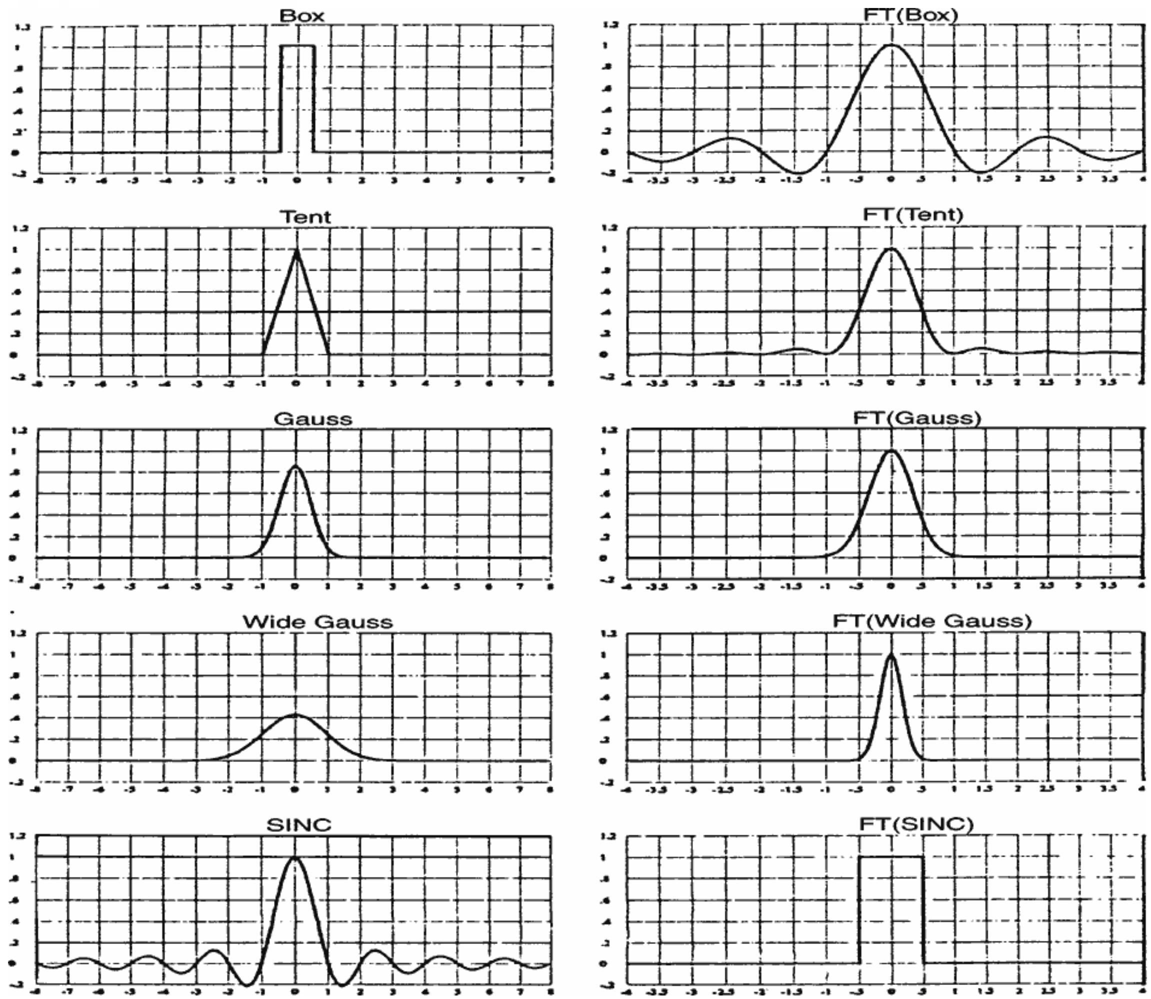
- Gaussian filters

$$g_{\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} \quad \leftrightarrow \quad G_{\sigma^2}(\omega) = e^{-\sigma^2\omega^2/2} = \frac{\sqrt{2\pi}}{\sigma} g_{1/\sigma^2}(\omega)$$

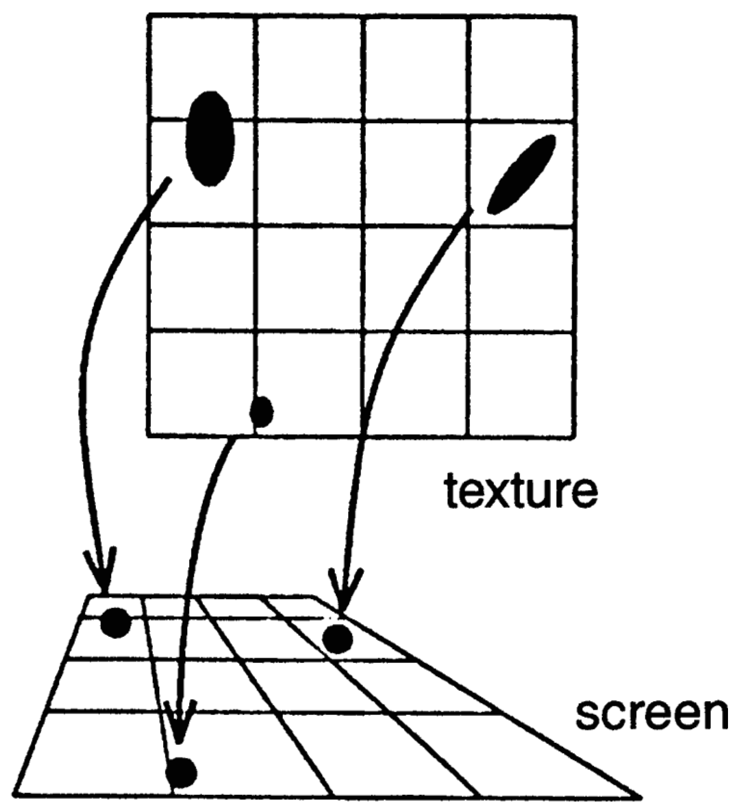
- Sinc-filter

$$sinc\left(\frac{\omega_c x}{\pi}\right) = \frac{\sin(\omega_c x)}{\pi x} \quad \leftrightarrow \quad g_1\left(\frac{\omega}{2\omega_c}\right) = \begin{cases} 1 & |\omega| \le \omega_c \\ 0 & |\omega| > \omega_c \end{cases}$$

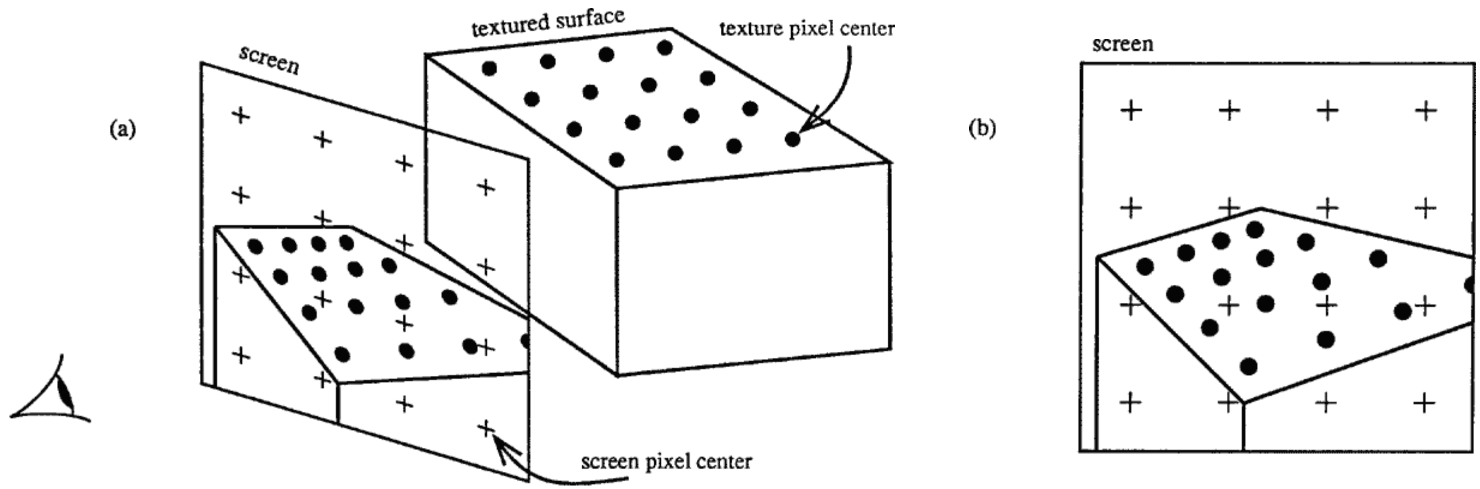
Filters and Fourier Transforms



Filtering in Texture Space and Screen Space



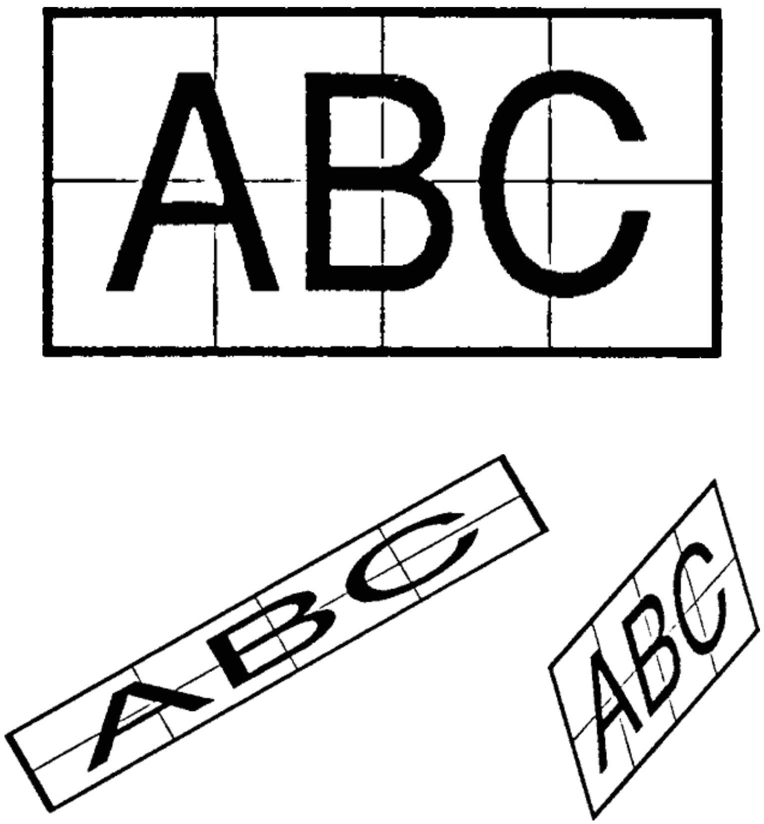
The Concept of Resampling Filters



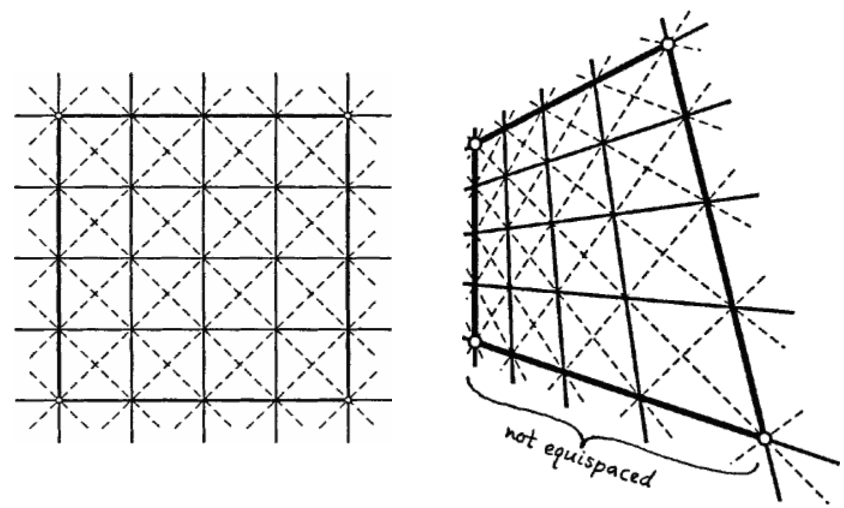
- Perspective projection of a textured surface
- Non-uniform sampling pattern on screen
- Optimal resampling filter is spatially variant

Projection and Image Warping

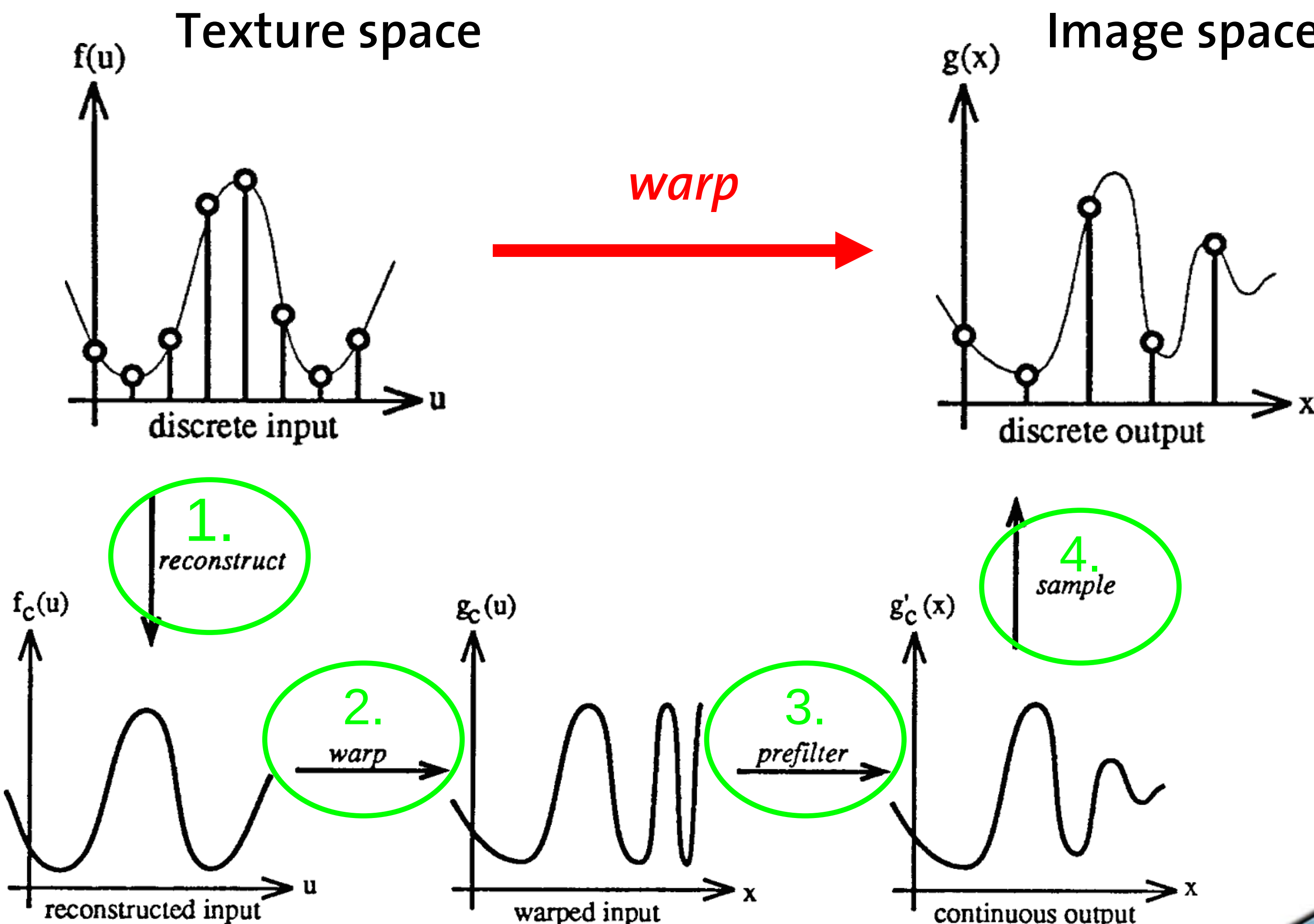
Affine Mapping



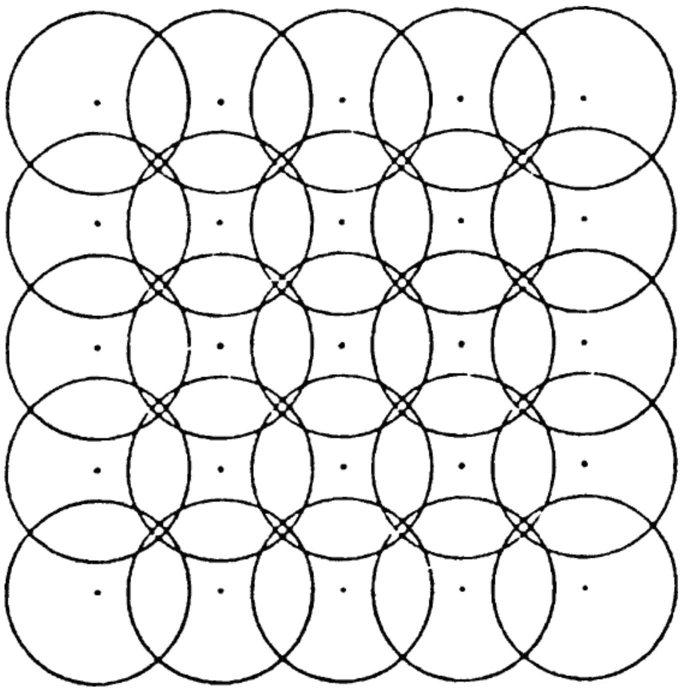
Projective Mapping



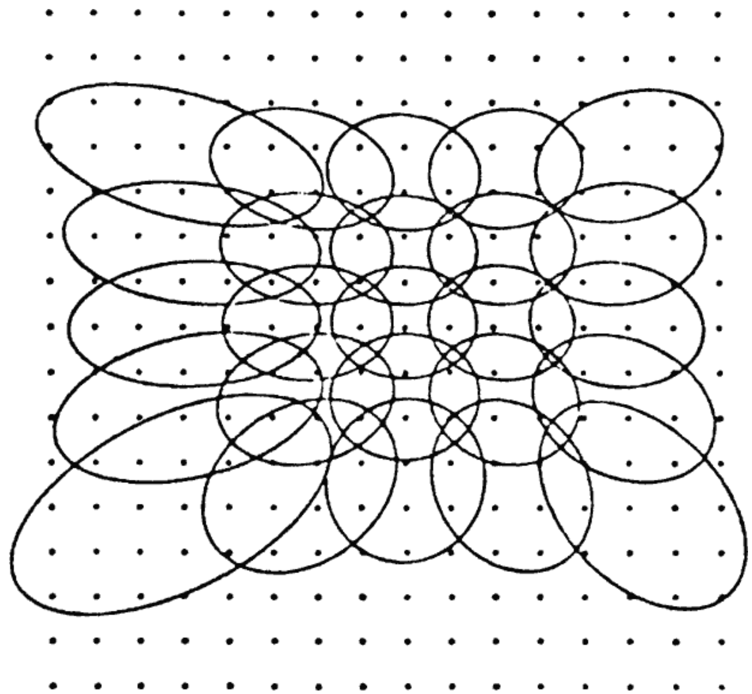
Relations between Texture and Image Space



Spatially Variant Filtering



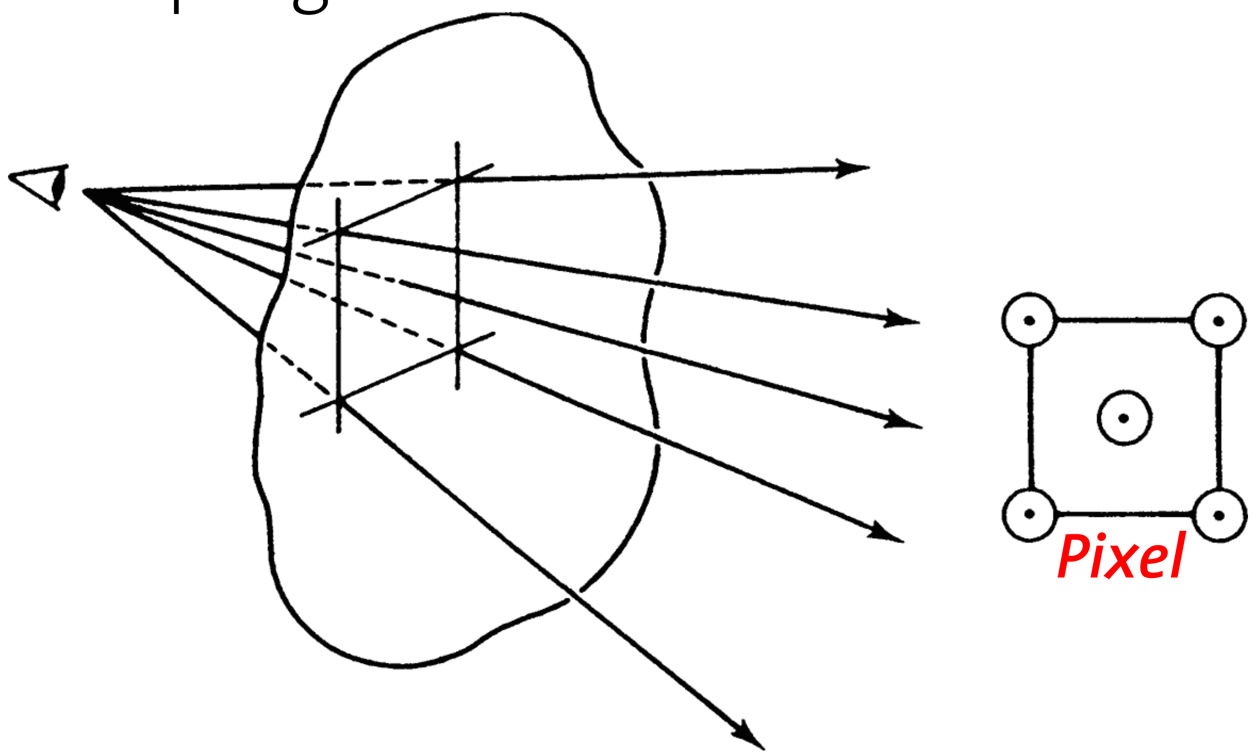
Screen Space



Texture Space

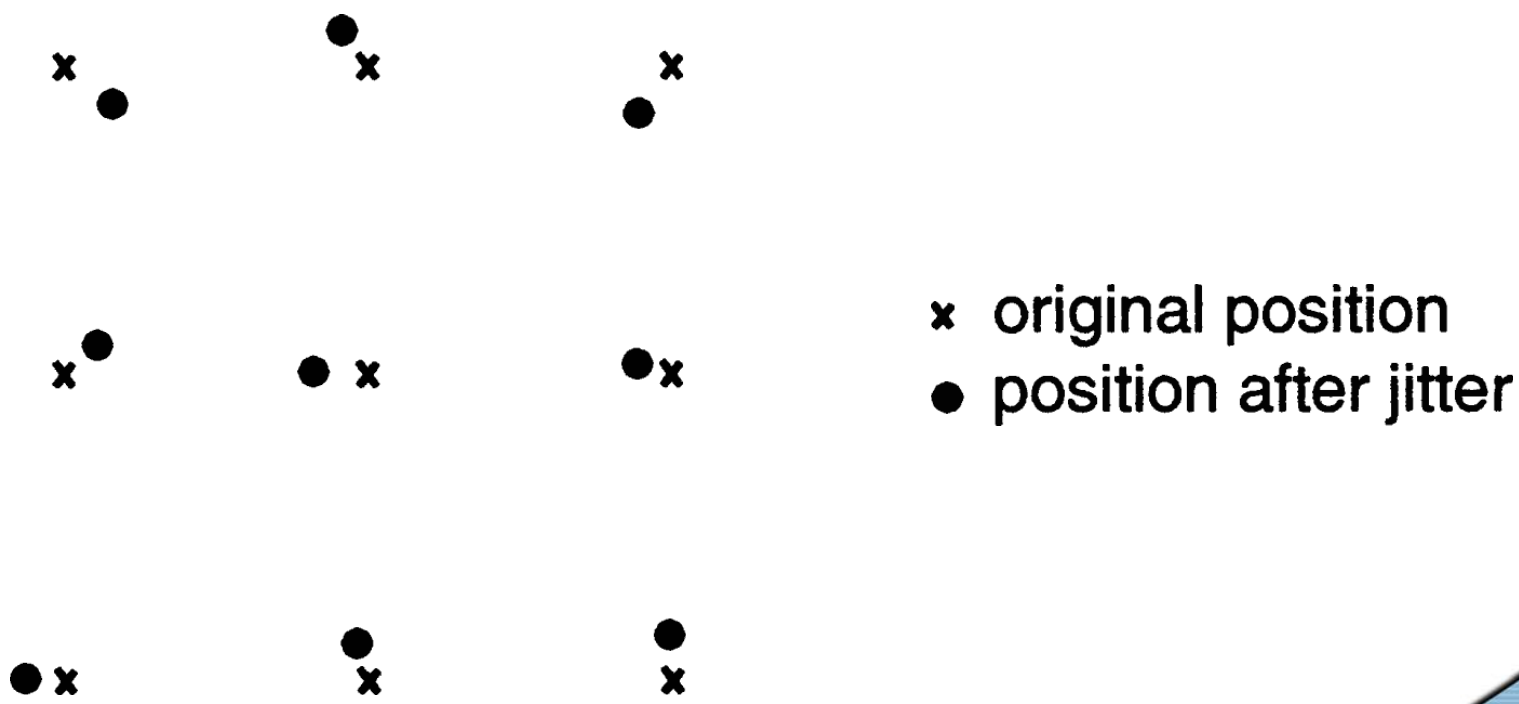
Antialiasing in Raytracing

- Supersampling

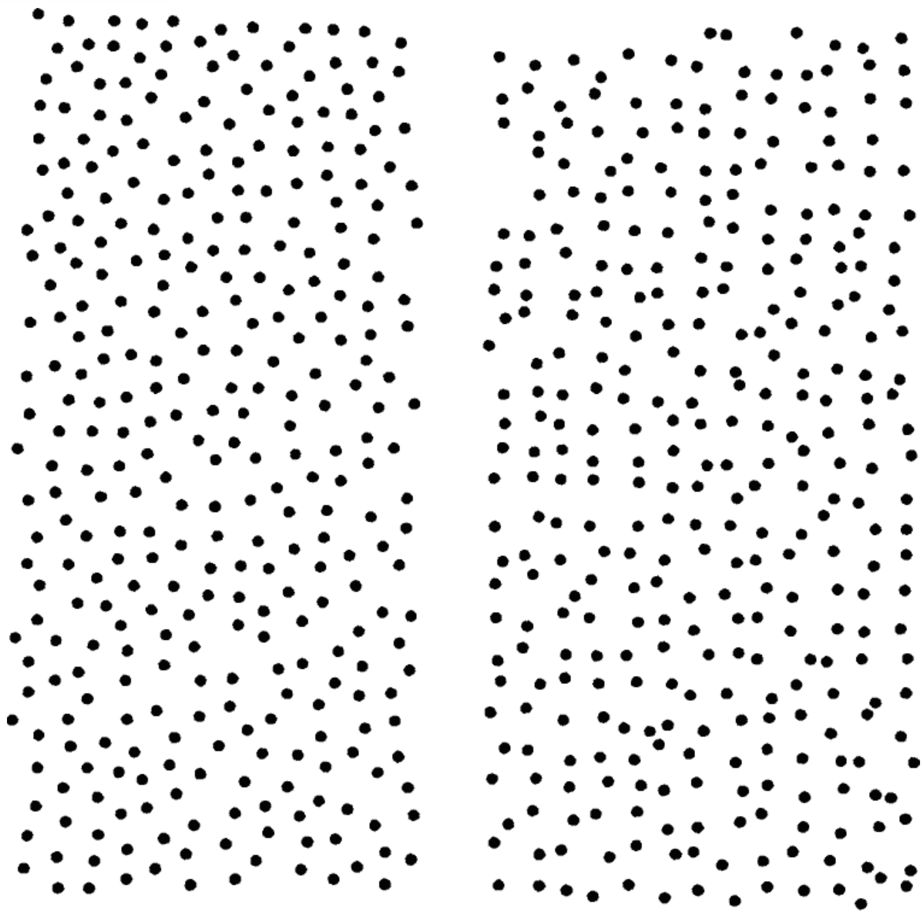


Jittering

- Random Perturbation of Sampling Positions



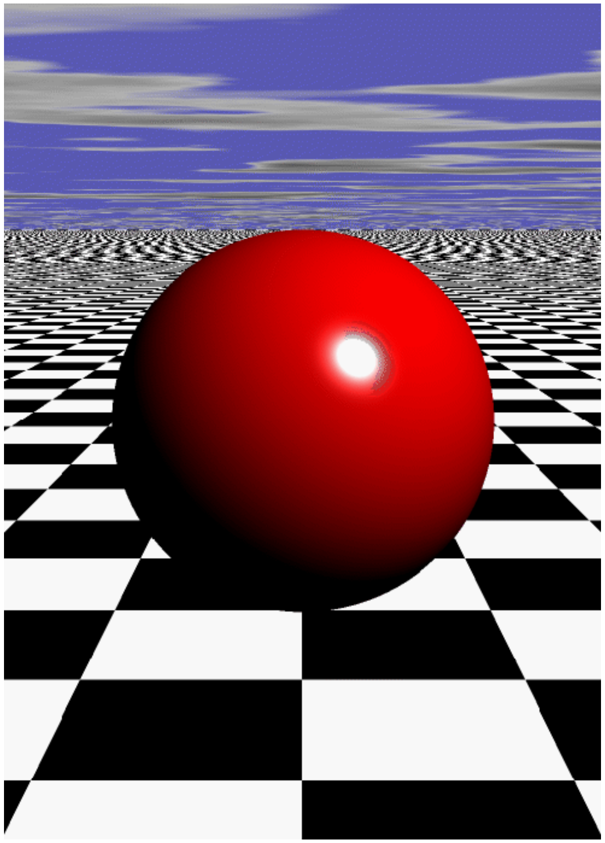
Poisson Sampling vs. Jittering



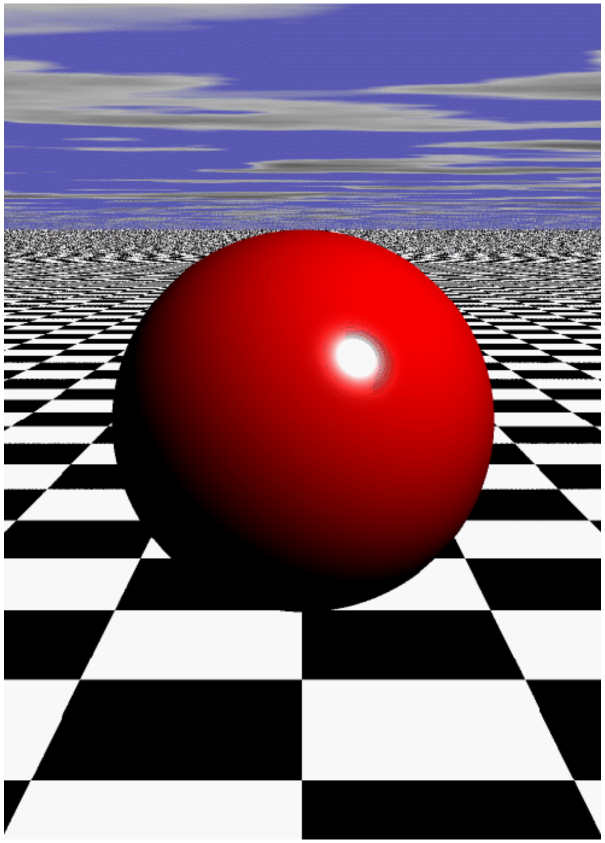
(a) (b)

Supersampling & Jittering

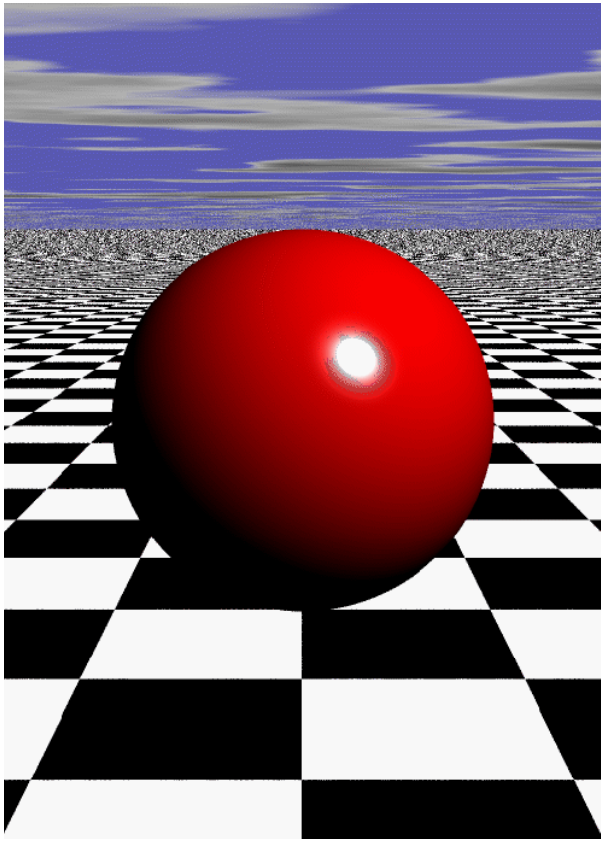
4 Rays/Pixel

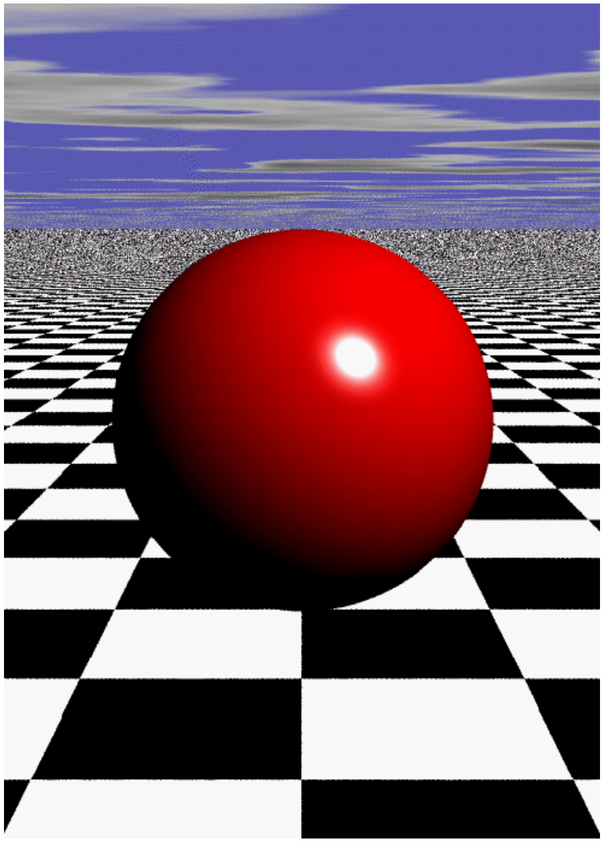


Jitter=0.3



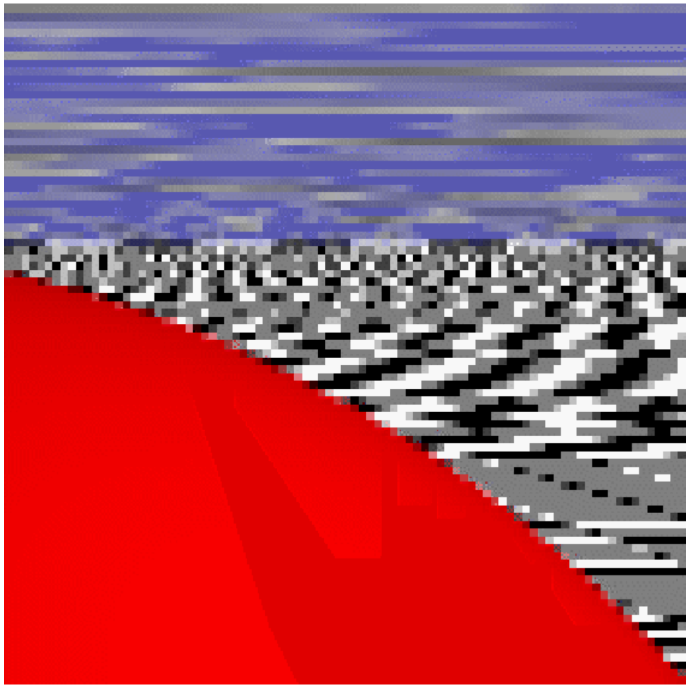
Supersampling & Jittering



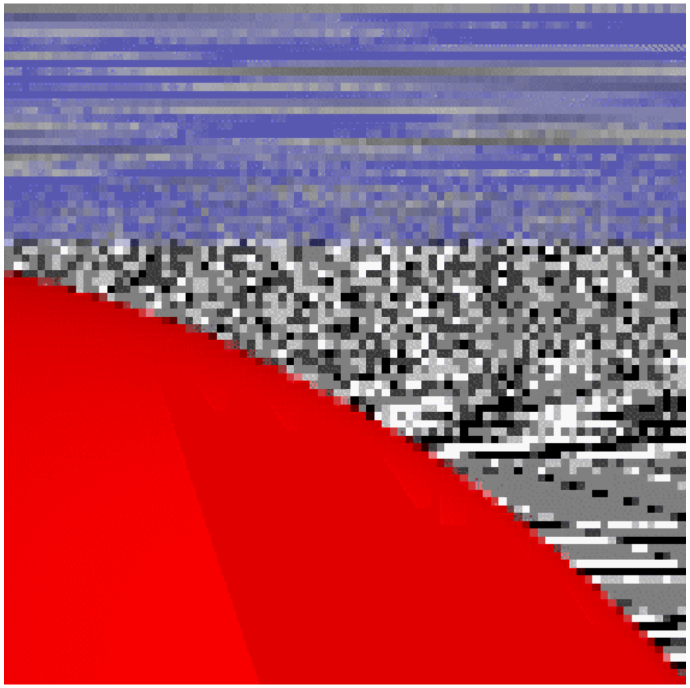


Supersampling & Jittering

4 Rays/Pixel

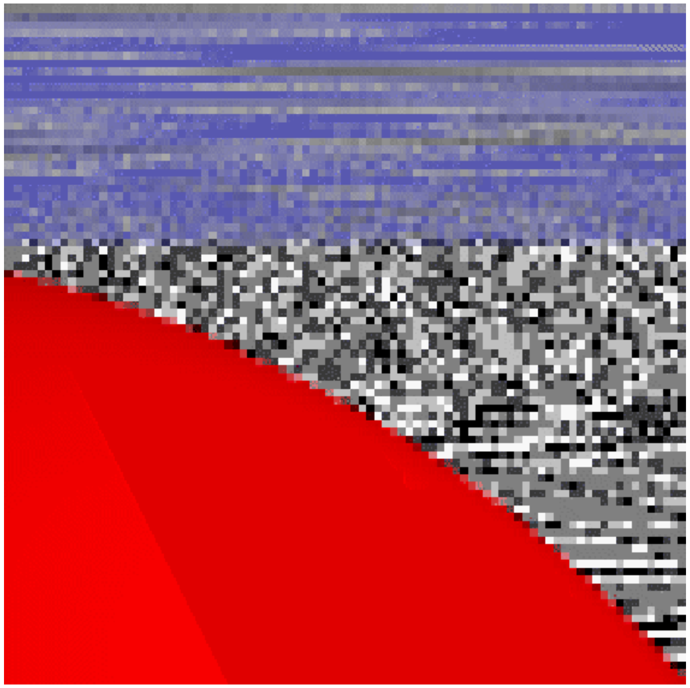


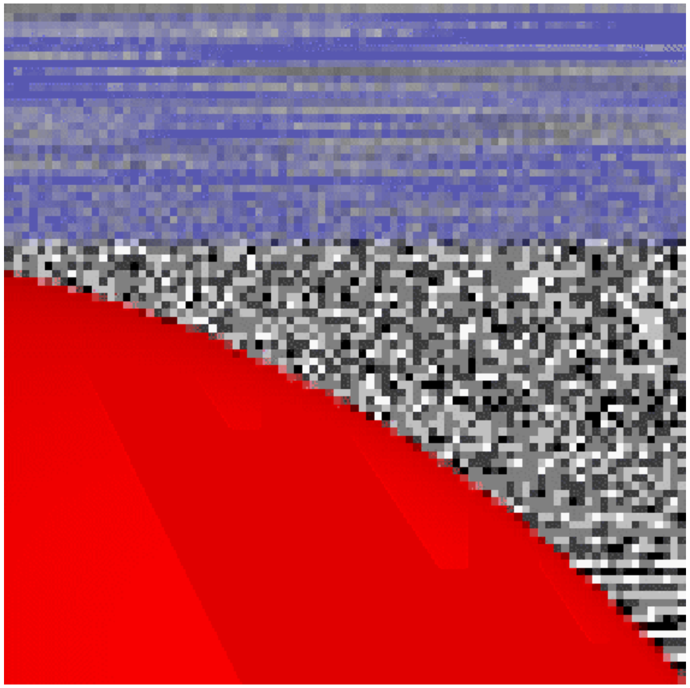
Jitter=0.3



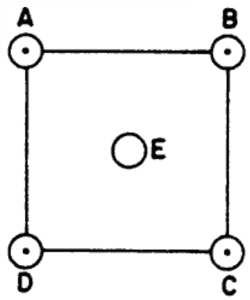
Supersampling & Jittering

Jitter=0.5 *Jitter=1.0*



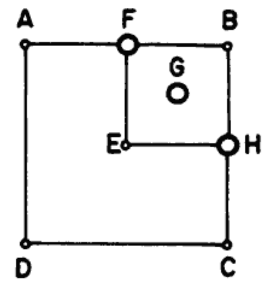


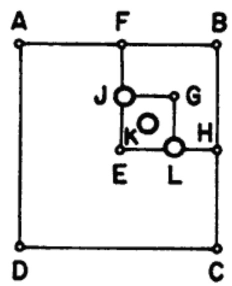
Adaptive Supersampling



When we start a pixel, we trace rays through the four corners and the center. We then compare the colors of rays **AE**, **BE**, **CE**, and **DE**. Suppose **A** and **E** are similar and so are **D** and **E**, but both **BE** and **CE** are too different.

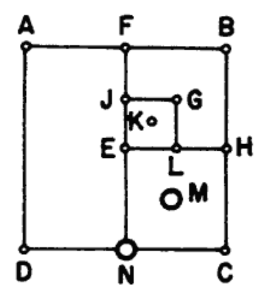
We'll start by looking more closely at the region bounded by **B** and **E**. We fire new rays **F**, **G**, **H** to find all four corners and the center of this region. We now compare **FG**, **BG**, **HG**, and **EG**. Suppose each pair is very similar, except **G** and **E**. So we look more closely at the region bounded by **G** and **E**.



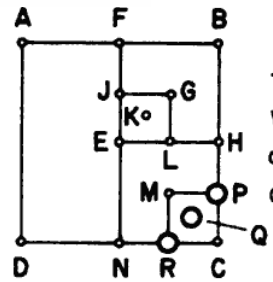


So now we fill in the square region bounded by **BE** with the three new rays **J**, **K**, and **L**. Let's suppose they're all sufficiently similar.

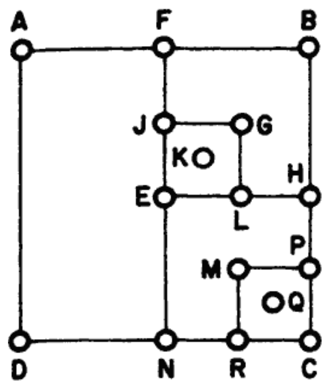
Now we return to the pair **CE** which we identified earlier. Since we already have **H**, we trace the new rays **M** and **N**. We compare the colors between **EM**, **HM**, **CM**, and **NM**. Suppose they are all similar except **CM**.



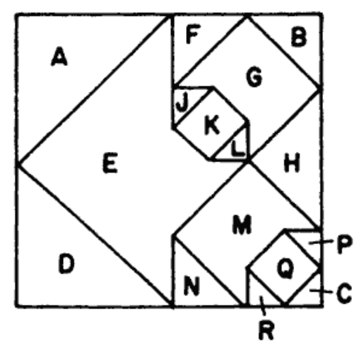
Adaptive Supersampling



To complete the region we trace the new rays P, Q, and R. We compare MQ, PQ, CQ, and RQ. At this point we'll assume they're all sufficiently similar. These are no pairs of colors left to examine, so we're now done.



So now its time to determine the final color. The rays on the left will end up with relative weights indicated by the diagram on the right. Basically, for each quadrant we average its four subquadrants recursively. The final formula for this example could then be expressed as:



$$\frac{1}{4}\left(\frac{A+E}{2}+\frac{D+E}{2}+\frac{1}{4}\left[\frac{F+G}{2}+\frac{B+G}{2}+\frac{H+G}{2}+\frac{1}{4}\left\{\frac{J+K}{2}+\frac{G+K}{2}+\frac{L+K}{2}+\frac{E+K}{2}\right\}\right]\right.$$

$$\left.+\frac{1}{4}\left[\frac{E+M}{2}+\frac{H+M}{2}+\frac{N+M}{2}+\frac{1}{4}\left\{\frac{M+Q}{2}+\frac{P+Q}{2}+\frac{C+Q}{2}+\frac{R+Q}{2}\right\}\right]\right)$$