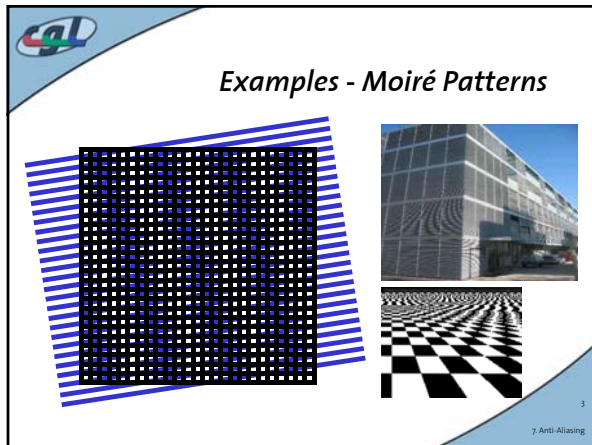


### Motivation

- 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 - Jaggies

- Staircase effect at borders

### Temporal Aliasing

time

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

Original signal

↓ Sampling

Sampled signal

↓ Reconstruction

Reconstructed signal

7 Anti-Aliasing

### Example – Point Sampling

8 Anti-Aliasing

### Signal Processing

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

9 Anti-Aliasing

### 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

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

10 Anti-Aliasing

### 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

11 Anti-Aliasing

### 1D Fourier Transform

- Fourier transform

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

- Inverse transform

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

12 Anti-Aliasing

### 1D Discrete Fourier Transform

- Discrete transform  $F(k) = \frac{1}{N} \sum_{i=0}^{N-1} f(i) e^{-\frac{j2\pi ki}{N}}$
- Discrete inverse  $f(i) = \sum_{k=0}^{N-1} F(k) e^{\frac{j2\pi ki}{N}}$ 
  - $x = i \cdot \Delta x$ ,
  - $u = k \cdot \Delta u$
- Heisenberg resolution bounds  $\Delta x \cdot \Delta u \geq \frac{1}{4\pi}$

13 Anti-Aliasing

### 2D Fourier Transforms

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

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

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

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

14 Anti-Aliasing

### Example: 2D Fourier Transforms

rect(x,y)  $f(x,y)$

sinc(x,y)  $|F(u,v)|$

(a) (b) (c)

sine cardinal:

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

15 Anti-Aliasing

### Reconstruction

$f(i\Delta x)$   $g(x)$  reconstruction filter

$f(x)$

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

16 Anti-Aliasing

### Convolutions

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

$f(x)$   $g(x)$

$f(\alpha)$   $g(x-\alpha)$   $x$   $\alpha$

17 Anti-Aliasing

### Convolutions

$f(x)$   $\delta(x)$

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

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

18 Anti-Aliasing

**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

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

7 Anti-Aliasing

**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** ( $L^2$ )

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

7 Anti-Aliasing

**Aliasing**

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

7 Anti-Aliasing

**Aliasing**

- Multiplication converts to convolution in Fourier domain

7 Anti-Aliasing

**Aliasing**

- Convolution with sequence of delta functions = periodization

7 Anti-Aliasing

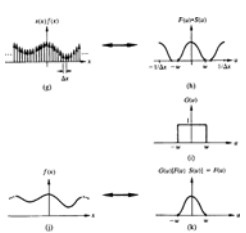
**Aliasing**

- Overlap of Fourier transforms leads to aliasing

7 Anti-Aliasing

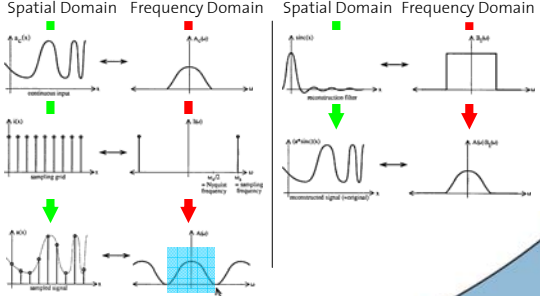
### Aliasing

- Reconstruction = Low pass filtering



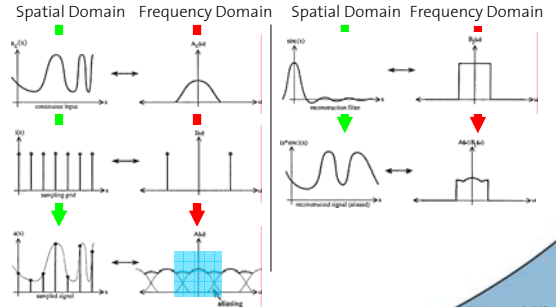
25  
7 Anti-Aliasing

### Aliasing-free Reconstruction



26  
7 Anti-Aliasing

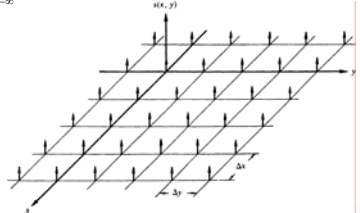
### Occurrence of Aliasing



27  
7 Anti-Aliasing

### 2D Sampling

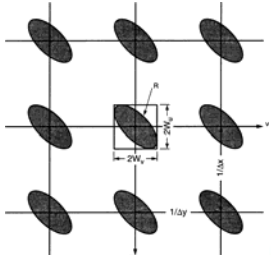
- 2D impulse fields

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


28  
7 Anti-Aliasing

### Fourier Domain

- Periodic spectrum of band limited sampled function



29  
7 Anti-Aliasing

### Reconstruction – Antialiasing

- Windowing spectrum using filters
- Simple

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

where

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

30  
7 Anti-Aliasing

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

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

31  
7 Anti-Aliasing

32  
7 Anti-Aliasing

## Spectral Analysis Geometry

33  
7 Anti-Aliasing

## Antialiasing Filters in Practice

- Properties of a good low pass filter

34  
7 Anti-Aliasing

## Antialiasing Filters

- B-Spline filters of order n
 
$$g_1(x) = \begin{cases} 1 & |x| \leq 1/2 \\ 0 & |x| > 1/2 \end{cases} \leftrightarrow \frac{\sin \omega/2}{\omega/2} = \frac{\sin \pi f}{\pi f} = \text{sinc } f$$
- Increase order by repeated convolution
 
$$g_n(x) = g_1(x) * g_1(x) * \dots * g_1(x) \leftrightarrow \text{sinc}^n f$$

35  
7 Anti-Aliasing

## Antialiasing Filters

- Gaussian filters
 
$$g_{\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} \leftrightarrow G_{\sigma^2}(\omega) = e^{-\sigma^2 \omega^2/2} = \frac{\sqrt{2\pi}}{\sigma} g_{1/\sigma^2}(\omega)$$
- Sinc-filter
 
$$\text{sinc}\left(\frac{\omega_c x}{\pi}\right) = \frac{\sin(\omega_c x)}{\pi x} \leftrightarrow g_1\left(\frac{\omega}{2\omega_c}\right) = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & |\omega| > \omega_c \end{cases}$$

36  
7 Anti-Aliasing

### Filters and Fourier Transforms

37  
7 Anti-Aliasing

### Filtering in Texture Space and Screen Space

38  
7 Anti-Aliasing

### The Concept of Resampling Filters

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

39  
7 Anti-Aliasing

### Projection and Image Warping

Affine Mapping

Projective Mapping

40  
7 Anti-Aliasing

### Relations between Texture and Image Space

41  
7 Anti-Aliasing

### Spatially Variant Filtering

42  
7 Anti-Aliasing

**Antialiasing in Raytracing**

- Supersampling

43  
7 Anti-Aliasing

**Jittering**

- Random Perturbation of Sampling Positions

44  
7 Anti-Aliasing

**Poisson Sampling vs. Jittering**

(a) (b)

45  
7 Anti-Aliasing

**Supersampling & Jittering**

4 Rays/Pixel Jitter=0.3

46  
7 Anti-Aliasing

**Supersampling & Jittering**

Jitter=0.5 Jitter=1.0


47  
7 Anti-Aliasing

**Supersampling & Jittering**

4 Rays/Pixel Jitter=0.3

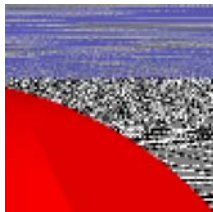
48  
7 Anti-Aliasing



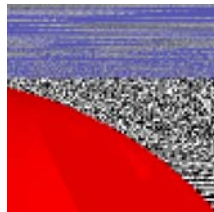


## Supersampling & Jittering


Jitter=0.5



Jitter=1.0



49 7 Anti-Aliasing



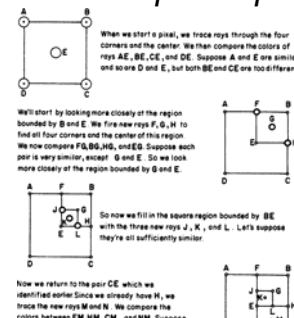
## Adaptive Supersampling

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, 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.



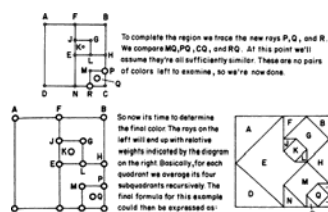
50 7 Anti-Aliasing



## 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. There are no pairs of colors left to examine, so we're now done.

So now it's time to determine the final color. The rays on the left will add up with 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:



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

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

51 7 Anti-Aliasing