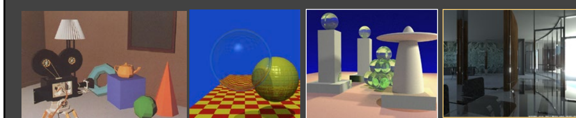
Computer Graphics II: Rendering

CSE 168[Spr 21],Lecture 11: Fourier Analysis, Sampling
Ravi Ramamoorthi

http://viscomp.ucsd.edu/classes/cse168/sp21

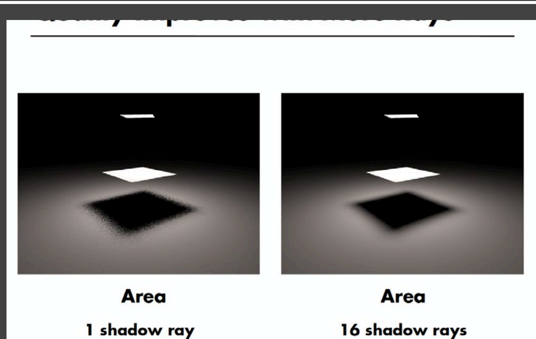


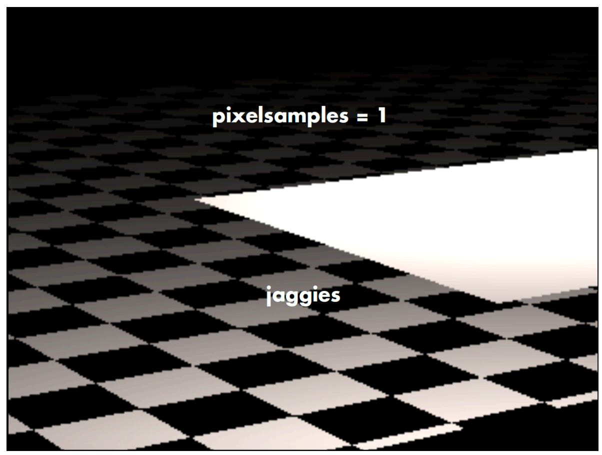
To Do

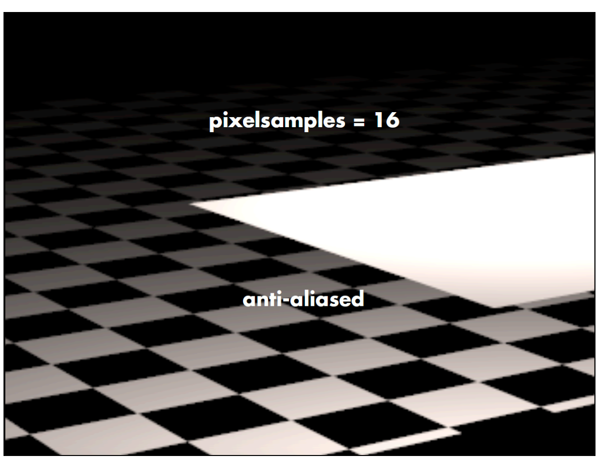
- Start immediately on homework 4.
- Start thinking about final project
- This lecture gives core background on sampling and signal-processing (bear in mind image processing)

Some slides courtesy Pat Hanrahan

Quality Improves with More Rays

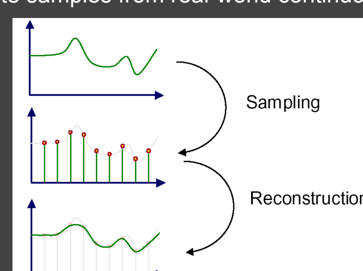






Sampling and Reconstruction

- An image is a 2D array of samples
- Discrete samples from real-world continuous signal



Sampling and Reconstruction

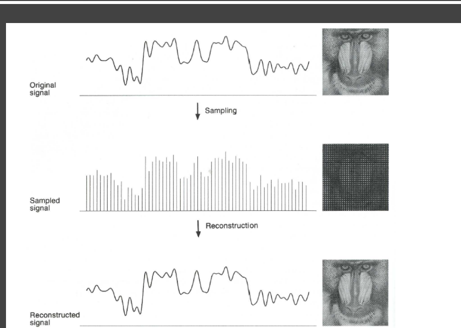
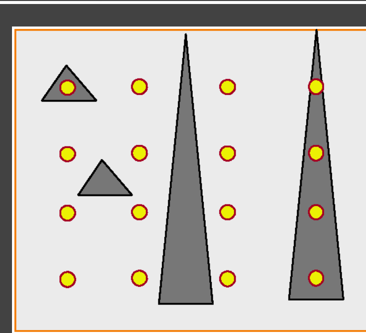


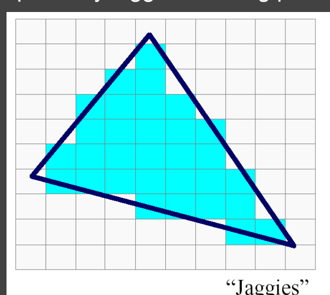
Figure 19.9 FvDFH

(Spatial) Aliasing

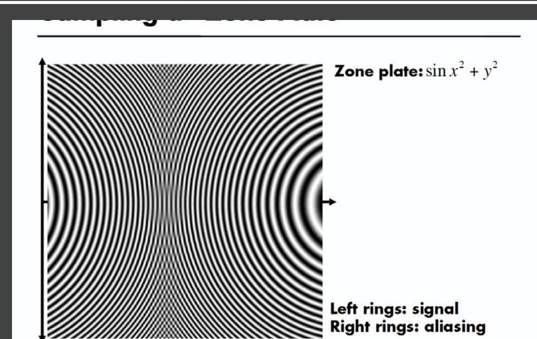


(Spatial) Aliasing

- Jaggies probably biggest aliasing problem



Sampling a Zone Plate



Sampling and Aliasing

- Artifacts due to undersampling or poor reconstruction
- Formally, high frequencies masquerading as low
- E.g. high frequency line as low freq jaggies

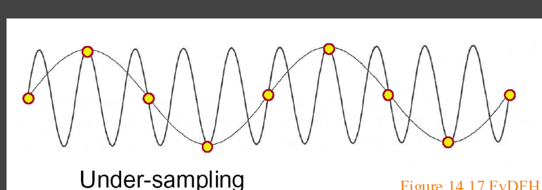
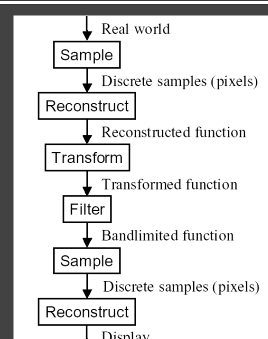


Figure 14.17 FvDFH

Image Processing pipeline



Motivation

- Formal analysis of sampling and reconstruction
- Important theory (signal-processing) for graphics
- Also relevant in rendering, modeling, animation

- Note: Fourier Analysis useful for understanding, but image processing often done in spatial domain

Ideas

- Signal (function of time generally, here of space)
- Continuous: defined at all points; discrete: on a grid
- High frequency: rapid variation; Low Freq: slow variation
- Images are converting continuous to discrete. Do this sampling as best as possible.
- Signal processing theory tells us how best to do this
- Based on concept of frequency domain Fourier analysis

Sampling Theory

Analysis in the frequency (not spatial) domain

- Sum of sine waves, with possibly different offsets (phase)
- Each wave different frequency, amplitude

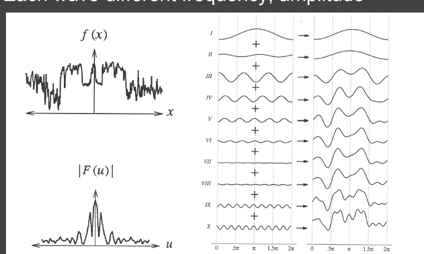


Figure 2.6 Wolberg

Fourier Transform

- Tool for converting from spatial to frequency domain

$$f(x) = \sum_{u=-\infty}^{+\infty} F(u) e^{2\pi iux}$$

$$e^{2\pi iux} = \cos(2\pi ux) + i \sin(2\pi ux)$$

- Or vice versa $\qquad i = \sqrt{-1}$
- One of most important mathematical ideas
- Computational algorithm: Fast Fourier Transform
  - One of 10 great algorithms scientific computing
  - Makes Fourier processing possible (images etc.)
  - Not discussed here, but look up if interested

Fourier Transform

- Simple case, function sum of sines, cosines

$$f(x) = \sum_{u=-\infty}^{+\infty} F(u) e^{2\pi iux}$$

$$F(u) = \int_0^1 f(x) e^{-2\pi iux} dx$$

- Continuous infinite case

Forward Transform: $F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi iux} dx$

Inverse Transform: $f(x) = \int_{-\infty}^{+\infty} F(u) e^{2\pi iux} du$

Fourier Transform

- Simple case, function sum of sines, cosines

$$f(x) = \sum_{u=-\infty}^{+\infty} F(u) e^{2\pi iux}$$

$$F(u) = \int_0^1 f(x) e^{-2\pi iux} dx$$

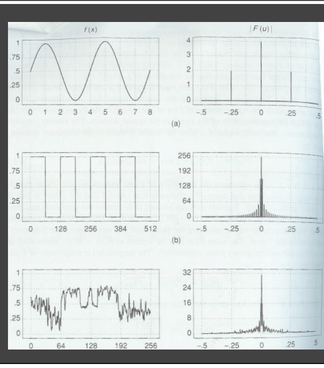
- Discrete case

$$F(u) = \sum_{x=0}^{x=N-1} f(x) \left[\cos(2\pi ux/N) - i \sin(2\pi ux/N) \right], \qquad 0 \le u \le N-1$$

$$f(x) = \frac{1}{N} \sum_{u=0}^{u=N-1} F(u) \left[\cos(2\pi ux/N) + i \sin(2\pi ux/N) \right], \qquad 0 \le x \le N-1$$

Fourier Transform: Examples 1

Single sine curve
(+constant DC term)



$$f(x) = \sum_{u=-\infty}^{+\infty} F(u) e^{2\pi iux}$$

$$F(u) = \int_0^1 f(x) e^{-2\pi iux} dx$$

Fourier Transform Examples 2

Forward Transform: $F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi iux} dx$

Inverse Transform: $f(x) = \int_{-\infty}^{+\infty} F(u) e^{2\pi iux} du$

- Common examples

$f(x)$	$F(u)$
$\delta(x - x_0)$	$e^{-2\pi iux_0}$
1	$\delta(u)$
e^{-ax^2}	$\sqrt{\pi / a}\, e^{-\pi^2 u^2 / a}$

Fourier Transform Properties

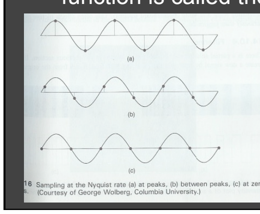
Forward Transform: $F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi iux} dx$

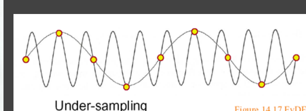
Inverse Transform: $f(x) = \int_{-\infty}^{+\infty} F(u) e^{2\pi iux} du$

- Common properties
  - Linearity: $F(af(x) + bg(x)) = aF(f(x)) + bF(g(x))$
  - Derivatives: [integrate by parts] $F(f'(x)) = \int_{-\infty}^{\infty} f'(x) e^{-2\pi iux} dx = 2\pi iuF(u)$
  - 2D Fourier Transform
    Forward Transform: $F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{-2\pi iux} e^{-2\pi ivy} dxdy$
    Inverse Transform: $f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{+\infty} F(u,v) e^{2\pi iux} e^{2\pi ivy} dudv$
- Convolution (next)

Sampling Theorem, Bandlimiting

- A signal can be reconstructed from its samples, if the original signal has no frequencies above half the sampling frequency – Shannon
- The minimum sampling rate for a bandlimited function is called the Nyquist rate





Sampling Theorem, Bandlimiting

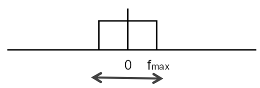
- A signal can be reconstructed from its samples, if the original signal has no frequencies above half the sampling frequency – Shannon
- The minimum sampling rate for a bandlimited function is called the Nyquist rate
- A signal is bandlimited if the highest frequency is bounded. This frequency is called the bandwidth
- In general, when we transform, we want to filter to bandlimit before sampling, to avoid aliasing

Antialiasing

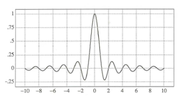
- Sample at higher rate
  - Not always possible
  - Real world: lines have infinitely high frequencies, can't sample at high enough resolution
- Prefilter to bandlimit signal
  - Low-pass filtering (blurring)
  - Trade blurriness for aliasing

Ideal bandlimiting filter

- Formal derivation is homework exercise
  - Frequency domain



  - Spatial domain



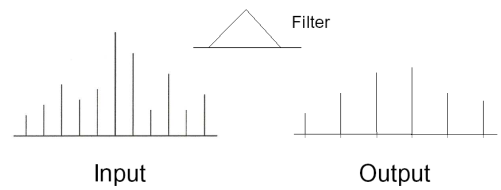
if full width $f_{max} = 1$

$$Sinc(x) = \frac{\sin \pi x}{\pi x}$$

Figure 4.5 Wolberg

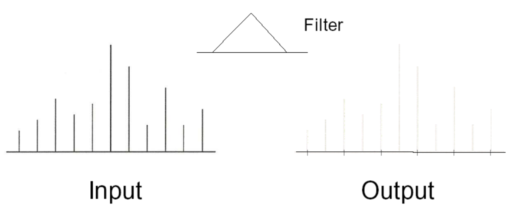
Convolution 1

- Spatial domain: output pixel is weighted sum of pixels in neighborhood of input image
  - Pattern of weights is the "filter"



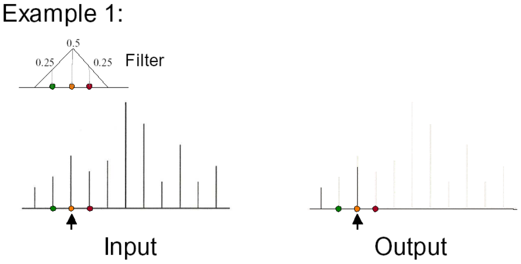
Convolution 2

- Example 1:



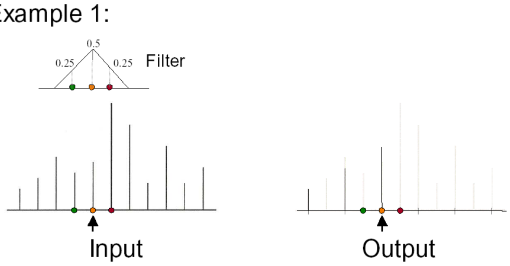
Convolution 3

- Example 1:



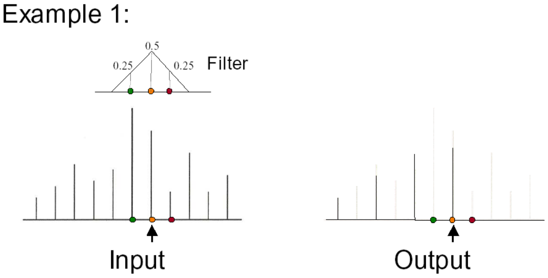
Convolution 4

- Example 1:



Convolution 5

- Example 1:



Convolution in Frequency Domain

Forward Transform: $F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} dx$

Inverse Transform: $f(x) = \int_{-\infty}^{+\infty} F(u) e^{2\pi i u x} du$

- Convolution (f is signal ; g is filter [or vice versa])

$$h(y) = \int_{-\infty}^{+\infty} f(x) g(y-x) dx = \int_{-\infty}^{+\infty} g(x) f(y-x) dx$$

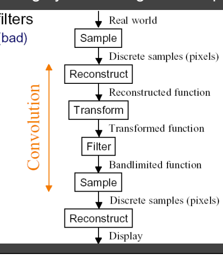
$$h = f * g \ \text{ or } \ f \otimes g$$

- Fourier analysis (frequency domain multiplication) $H(u) = F(u)G(u)$

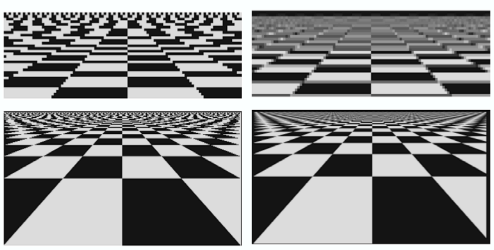
Practical Image Processing

- Discrete convolution (in spatial domain) with filters for various digital signal processing operations
- Easy to analyze, understand effects in frequency domain
  - E.g. blurring or bandlimiting by convolving with low pass filter

- Finite low-pass filters
  - Point sampling (bad)
  - Triangle filter
  - Gaussian filter



Point vs Area Sampling



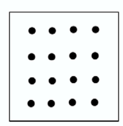
Point **Exact Area**

Checkerboard sequence by Tom Duff

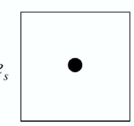
Uniform Supersampling

Increasing the number of samples moves each copy of the spectra further apart, thus there is less overlap

This reduces, but does not eliminate, aliasing



$$Pixel = \sum_s w_s \cdot Sample_s$$



Samples **Pixel**

Non-uniform Sampling

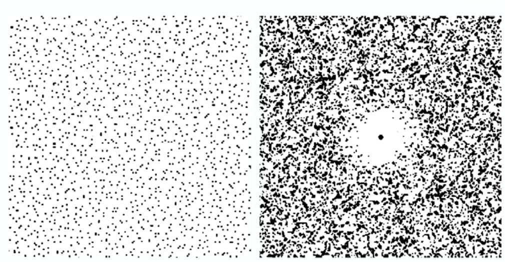
Uniform sampling

- **The spectrum of uniformly spaced samples is also a set of uniformly spaced spikes**
- **Multiplying the signal by the sampling pattern corresponds to placing a copy of the spectrum at each spike (in freq. space)**
- **Aliases are coherent, and very noticable**

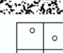
Non-uniform sampling

- **Samples at non-uniform locations have a different spectrum; a single spike plus noise**
- **Sampling a signal in this way converts aliases into broadband noise**
- **Noise is incoherent, and much less objectionable**
- **May cause error in the integral**

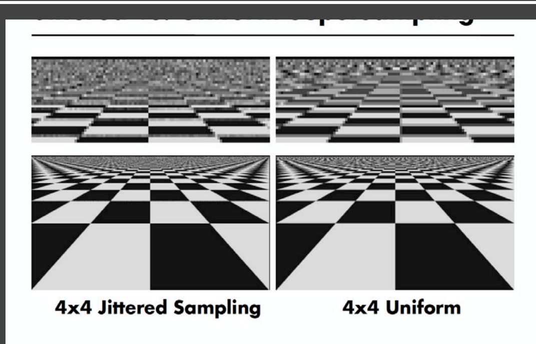
Jittered Sampling



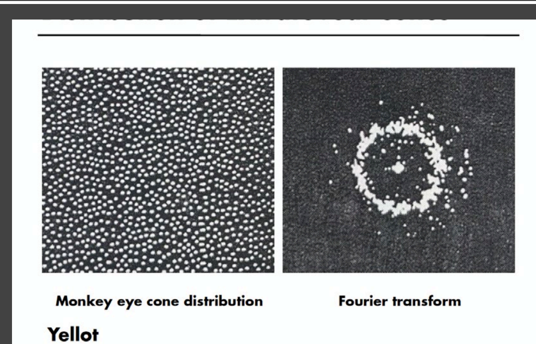
Add uniform random jitter to each sample



Jittered vs Uniform Supersampling



Distribution of Extrafoveal Cones



Poisson Disk Sampling