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

pixelsamples = 1
jaggies

Area
1 shadow ray
16 shadow rays

anti-aliased

pixelsamples = 16
Sampling and Reconstruction

(Spatial) Aliasing

- Jaggies probably biggest aliasing problem

Sampling a Zone Plate

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

Image Processing pipeline

- Real world
- Sample
- Discrete samples (pixels)
- Reconstruct
- Reconstructed function
- Transform
- Transformed function
- Filter
- Bandlimited function
- Sample
- Discrete samples (pixels)
- Reconstruct
- Display
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

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
\[ i = \sqrt{-1} \]
- 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

Simple case, function sum of sines, cosines
\[ f(x) = \sum_{u=-\infty}^{\infty} F(u)e^{2\pi iux} \]
\[ F(u) = \int_{-\infty}^{\infty} f(x)e^{-2\pi iux} \text{d}x \]

Continuous infinite case

Forward Transform:
\[ F(u) = \int_{-\infty}^{\infty} f(x)e^{-2\pi iux} \text{d}x \]

Inverse Transform:
\[ f(x) = \int_{-\infty}^{\infty} F(u)e^{2\pi iux} \text{d}u \]

Discrete case
\[ F(u) = \sum_{x=0}^{N-1} f(x)\left[\cos\left(2\pi ux / N\right) - i\sin\left(2\pi ux / N\right)\right], \quad 0 \leq u \leq N-1 \]
\[ f(x) = \frac{1}{N} \sum_{u=0}^{N-1} F(u)\left[\cos\left(2\pi ux / N\right) + i\sin\left(2\pi ux / N\right)\right], \quad 0 \leq x \leq N-1 \]
Fourier Transform: Examples 1

Single sine curve (+constant DC term)

\[ f(x) = \sum_{u} F(u) e^{2\pi i u x} \]

\[ F(u) = \int_{x} f(x) e^{-2\pi i u x} dx \]

Fourier Transform Examples 2

Forward Transform:

\[ F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} dx \]

Inverse Transform:

\[ f(x) = \int_{-\infty}^{\infty} F(u) e^{2\pi i u x} du \]

Common examples:

\[ \delta(x - x_0)e^{-2\pi i u x} \]

\[ \delta(u) \]

\[ e^{-ax^2} \]

\[ e^{-\pi^2 u^2 / a} \]

Fourier Transform Properties

Forward Transform:

\[ F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} dx \]

Inverse Transform:

\[ f(x) = \int_{-\infty}^{\infty} F(u) e^{2\pi i u x} du \]

Common properties:

- Linearity: \[ F(a f(x) + b g(x)) = a F(f(x)) + b F(g(x)) \]
- Derivatives: [integrate by parts]
  \[ F(f'(x)) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} dx \]
  \[ = 2\pi i u F(f) \]
- 2D Fourier Transform
  \[ F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{-2\pi i u x} e^{-2\pi i v y} dy dx \]
- Convolution (next)
  \[ F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{-2\pi i (u x + v y)} dy dx \]

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

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

*Figure 4.5 Websg*  

**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 2 Example](image)

**Convolution 3**

- Example 1:

  ![Convolution 3 Example](image)

**Convolution 4**

- Example 1:

  ![Convolution 4 Example](image)

**Convolution 5**

- Example 1:

  ![Convolution 5 Example](image)
Convolution in Frequency Domain

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

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

- Convolution (f is signal; g is filter [or vice versa])
  \[ h(y) = \int f(x) g(y-x) dx = \int g(x) f(y-x) dx \]
  \[ h = f \ast g \text{ or } f \odot g \]
- 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

Point vs Area Sampling

- 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
  \[ Pixel = \sum_{i} w_i \cdot Sample_i \]

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 noticeable
- 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