Overview
- So far:
  - ...
  - Image synthesis
    - Ray tracing, texture mapping
- Today
  - Antialiasing
    - Prefiltering
    - Supersampling
    - Postfiltering
    - Stochastic and adaptive sampling

Antialiasing
- In computer graphics an analog signal is point sampled
  - Image is just a continuous analog signal that is sampled at discrete pixel positions
  - Pixel spacing determines the frequencies (the size of details) that can be reconstructed
  - Undersampling the image causes aliasing artifacts
  - Note: sampling frequency decreases with increasing distance to the viewpoint

Antialiasing
- Original scene and brightness distribution along a scan line
Antialiasing
- Point sampling the scene at pixel centers

Antialiasing
- The rendered image

Antialiasing
- Jagged profiles

Antialiasing
- Loss of details

Antialiasing
- Disintegrated texture mapping

Antialiasing
- Cause of aliasing
  - The sampling frequency is not high enough to cover all details
  - It is below the Nyquist limit:
    - Shannon's Sampling theorem: "the signal has to be sampled at a frequency that is equal to or higher than two times the highest frequency in the signal"
  - Overlap between replicated copies in frequency domain
    - High frequencies appear as low frequency regular patterns
  - Understand the basics of sampling theory
Fourier transform

- Two different approaches to describe a function
  - Spatial domain vs. frequency domain
  - Every periodic function can be represented as
    
    \[
    f(x) = \frac{1}{2} a_0 + \sum_{k=1}^{\infty} \left( a_k \cos(2\pi k x) + b_k \sin(2\pi k x) \right)
    \]
  - With
    
    \[
    a_k = \frac{1}{\pi} \int_{0}^{\pi} f(x) \cos(2\pi k x) dx, \quad b_k = \frac{1}{\pi} \int_{0}^{\pi} f(x) \sin(2\pi k x) dx
    \]

- Example from http://mathworld.wolfram.com/FourierSeries.html

- Non-periodic functions
  - Every reasonable function \( s(x) \) can be represented as a superposition of harmonic (sin/cos) functions
    
    \[
    s(x) = \text{IFT} (S(f)) = \int S(f) e^{2\pi i f'} dx
    \]
    
    \[
    S(f) = \text{FT} (s(x)) = \int s(x) e^{-2\pi i f x} dx
    \]
  - \( e^{i\theta} = \cos \theta + i \sin \theta \)
  - FT: Fourier transform
  - IFT: Inverse Fourier transform

- Some functions and their frequency representation

- Some functions and their frequency representation

- Some functions and their frequency representation
Fourier transform

- Some functions and their frequency representation

\[ s(x) = 1 \quad S(f) = \delta(f) \]

\[ \delta(f) = \begin{cases} \infty & \text{if } f = 0 \\ 0 & \text{else} \end{cases} \]

see exercise

Fourier transform

- Some functions and their frequency representation

\[ s(x) \quad S(f) \]

The comb function: many regularly spaced copies of the Dirac delta function

\[ s(x) = \sum_{k=-\infty}^{\infty} \delta(x - kT) \]

Spatial domain

\[ \xi(f) = \sum_{k=-\infty}^{\infty} \delta(f - kT) \]

Frequency domain

Frequency space operations

- Operations between signals can either be performed in spatial or frequency domain
- Transform signals to frequency space
- Perform operation in frequency space
- Transform result back to spatial domain

\[ g(x) \text{ op } h(x) = IFT(FT(g(x)) \text{ OP } FT(g(x))) \]

Convolution

- Sliding function \( v(x) \) along function \( u(x) \)

\[ s(x) = u(x) * v(x) = \int_{-\infty}^{\infty} u(x') \cdot v(x - x') dx' \]

http://mathworld.wolfram.com/Convolution.html
Convolution

- Convolution with the delta function \( \delta(x) \)
  \[
  s(x) = u(x) * \delta(x) = u(x)
  \]
  \[
  s(x) = u(x) * \delta(x - x_0) = u(x - x_0)
  \]
- Convolution with comb function
  \[
  s(x) = u(x) * \sum_{k=-\infty}^{\infty} \delta(x - kT) = \sum_{k=-\infty}^{\infty} u(x - kT)
  \]

In frequency domain: convolution becomes multiplication

\[
FT(u(x) * v(x)) = FT(u(x)) \cdot FT(v(x))
\]

... and multiplication becomes convolution

\[
FT(u(x) \cdot v(x)) = FT(u(x)) \ast FT(v(x))
\]

Sampling

- Question
  - What is sampling, i.e. evaluation of a continuous function at evenly spaced positions?
- Answer
  - Multiplication of the function with an appropriately spaced comb function

In frequency domain

- Multiplication with a comb function becomes convolution with a comb function of different spacing
- Example: given spectrum \( S(f) \) of a signal \( s(x) \)

Convolution with comb-function produces multiple shifted copies of \( S(f) \)
- If \( 1/T \) is large enough, the individual copies do not overlap
- Depends on maximum frequency \( f_0 \) in \( s(t) \)

- If \( T \) is too large (1/T becomes too small), overlap occurs
Reconstruction

- Question
  - Can we obtain the original function \( s(t) \) if we are only given the discrete samples?
- Answer
  - Only if sampling frequency \( T \) has been chosen large enough, so that the copies of \( S(f) \) do not overlap

THIS IS CALLED ALIASING

Reconstruction

- What happens if \( T \) was not small enough?
  - Copies of \( S(f) \) overlap
  - Cannot clearly separate the individual copies
  - High frequencies of one copy get interpreted as low frequencies of a neighboring copy

Sampling and Reconstruction

Antialiasing

- Cause of aliasing
  - Point sampling is the multiplication of the analog signal with an impulse train
  - Look at discrete sampling in the frequency domain
    - The Fourier Transform of an impulse function with spacing \( T \) in the spatial domain gives an impulse function with spacing \( 1/T \) in the frequency domain
    - The multiplication of functions in one domain corresponds to the convolution of both functions in the other domain

Antialiasing

- Aliasing artefacts
  - Moiré interference pattern
  - Aliasing

- Aliasing artefacts
  - Moiré interference pattern
  - Aliasing

- Aliasing artefacts
  - Moiré interference pattern
  - Aliasing

- Aliasing artefacts
  - Moiré interference pattern
  - Aliasing

- Aliasing artefacts
  - Moiré interference pattern
  - Aliasing

- Aliasing artefacts
  - Moiré interference pattern
  - Aliasing
Antialiasing

- Aliasing artefacts
  - Stair cases – Jaggies
    - Abrupt change in intensity
    - Edges, texture, shadows, highlights
  - Not always aliasing artefacts
  - Rather reconstruction artefacts
    - Result from orientation of the pixel grid

Spatial aliasing

- Supersampling - increase sampling frequency
  - OK, but doesn’t eliminate aliasing
- Prefiltering
  - Antialiasing before sampling
  - Analytic low-pass filtering of geometry
    - Hard to implement
    - Ideal eliminates aliasing completely
    - Sampling the shape of an object very densely within a pixel region
    - Supersampling
  - Postfiltering after reconstruction
    - NO, just blurs the image

Temporal aliasing

- Increase frame rate
  - OK
- Prefiltering (motion blur)
  - Yes, but only for simple geometries
  - Problems with textures etc.
- Postfiltering - averaging several frames
  - NO, generates “double images”

Antialiasing by prefiltering

- Analytic
- Supersampling
  - Higher frequencies
- Ideal reconstruction
  - Convolution with sinc
- Real reconstruction
  - Convolution with box or sphere
  - Sampling frequency must be much higher than Nyquist-frequency

Analytic low-pass filtering

- Ideally eliminates aliasing completely
  - Only works for polygon edges with constant color
  - Weighted or unweighted area sampling
  - Compute distance from pixel to edge
  - Doesn’t work for corners
- Over/Supersampling
  - Very easy to implement
  - Doesn’t eliminate aliasing
    - Sharp edges contain infinitely high frequencies
Antialiasing

- Prefiltering
  - Treat a pixel as an area rather than as a point
  - Determine the color of the part of the object that is covered by that area
  - Use this color as the pixel color

- Prefiltering combines color contributions into a pixel

Example

- Texture filtering
  - Combining texels to determine pixel color
    - Minification: pixels are larger than one texel
    - Magnification: pixels are smaller than one texel
  - Filtering methods
    - Nearest: choose the texel closest to the pixel center
    - Linear: (bi/tri)-linear texture interpolation
    - MipMap: (bi/tri)-linear interpolation in a stack of textures of decreasing resolution
Antialiasing

- Texture filtering

![Texture filtering diagram]

Antialiasing

- Mip-Mapping
  - Only used for minification
  - Copies of the texture are generated that contain the data at every coarser resolution
  - Match between texture resolution and pixel resolution can be achieved

![Mip-Mapping diagram]

Antialiasing

- Supersampling
  - Uses several samples from the scene
  - Averages these samples to get the pixel color
  - Virtually increases image resolution
  - Down-filter the high resolution image to the original size
  - Use different kinds of filters for down-filtering

![Supersampling diagram]
Antialiasing

- Filtering example

Antialiasing

- Instead of regular supersampling patterns use stochastic sampling -> visually less noticeable

Antialiasing

- Regular sampling
  - Visibility of aliases is due to regular sampling grid
  - Human visual system:
    - Sensitive against regular structures
    - But rather insensitive against high frequency noise
- Stochastic sampling
  - Alias frequencies are converted to noise

Antialiasing

- Stochastic supersampling
  - Uniform distribution
  - Minimal correlation between samples
  - Poisson-disk sampling
    - Minimal distance between samples
    - Random generation of samples
  - Jittered sampling
    - Random jittering from regular grid points
  - Stratified sampling
    - Regular partitioning of pixel region
    - One random sample per partition

Antialiasing

- Poisson-disk sampling
  - Distribution of optical receptors on retina

Antialiasing

- Adaptive supersampling
  - Performs supersampling in regions of high frequencies
    - Edges, shadows etc.
  - Has to weight contributions appropriately
Antialiasing

- Example:

[Image: Antialiasing example 1]

[Image: Antialiasing example 2]