Antialiasing

Han-Wei Shen
What is alias?

- **Alias** - A false signal in telecommunication links from beats between signal frequency and sampling frequency (from dictionary.com)
  - Not just telecommunication, alias is everywhere in computer graphics because rendering is essentially a sampling process
- Examples:
  - Jagged edges
  - False texture patterns
Alias caused by under-sampling

- 1D signal sampling example

---

Actual signal

Sampled signal
Alias caused by under-sampling

- 2D texture display example

Minor aliasing

worse aliasing
How often is enough?

- What is the right sampling frequency?
- **Sampling theorem (or Nyquist limit)** - the sampling frequency has to be at least twice the *maximum* frequency of the signal to be sampled.

Need two samples in this cycle.
Reconstruction

- After the (ideal) sampling is done, we need to reconstruct back the original continuous signal.
- The reconstruction is done by a reconstruction filter.
Reconstruction Filters

- Common reconstruction filters:
  - Box filter
  - Tent filter
  - Sinc filter = $\frac{\sin(\pi x)}{\pi x}$
Anti-aliased texture mapping

- Two problems to address –
  - Magnification
  - Minification
Re-sampling

- Minification and Magnification – resample the signal to a different resolution

(note the minification is done badly here)
Magnification

- Simpler to handle, just resample the reconstructed signal
Magnification

- Common methods: nearest neighbor (box filter) or linear interpolation (tent filter)
Minification

- Harder to handle
- The signal’s frequency is too high to avoid aliasing
- A possible solution:
  - Increase the low pass filter width of the ideal sinc filter – this effectively blur the image
  - Blur the image first (using any method), and then sample it
Minification

- Several texels cover one pixel (under sampling happens)

Solution:

Either increase sampling rate or reduce the texture frequency.

We will discuss:

1. Mip-mapping
2. Rip-mapping
3. Sum Area Table
Mip-mapping

- By Lance Williams (SIGGRAPH ’83)
- Mip – many things in a small place
- The original texture is augmented with a set of low resolution textures
- Which resolution to use depends on the screen projection size – try to maintain pixel to texel ratio close to 1
Mipmapping

- Two common measures are used to compute $d$.
  - Use the longer edge of the quadrilateral
  - Use the largest absolute value of the four differentials ($\partial u/\partial x, \partial v/\partial x, \partial u/\partial y, \partial v/\partial y$)
- Interpolate between levels
Mipmapping problem

- Overblurring! – When a pixel covers many $u$ texels but few $v$ texels, we always choose the largest pixel coverage to decide the level
Ripmapping

- Correct the over-blurring problem of mipmapping – separating x and y filtering

- Scale by half by x across a row.
- Scale by half in y going down a column.
- The diagonal has the equivalent mip-map.
- Four times the amount of storage is required.
- Four different texel values to average
Summed Area Table

- Another way to perform anisotropic filtering - can be used to compute the average color for any arbitrary rectangular region in the texture space at a constant speed.
- A two dimensional array that has the same size as the texture.

Every pixel stores the sum of all the texel colors in the lower left corner.
Summed Area Table (SAT)

- How to calculate the sum of texels in the area between A and B?


The average value can be computed by

\[ \frac{R}{\text{num of texels in } R} \]
Summed Area Table (SAT)

- is used to filter the texture

Compute the rectangular bounding box of the pixel area in the texture and then use summed area table to compute the average color (i.e., filter the texels when magnification takes place)
Summed Area Table

- Less blurry than Mipmapping
- Still blurry when the texture is viewed along its diagonal – this is because the bounding box of the pixel area is larger than the actual pixel area
Comparison

Non filtering

Mipmapping

Summed area table

Note that only mipmap is implemented by hardware and supported by OpenGL