Anti-Aliasing Techniques

How do we remove aliasing?
- Cheaper solution: take multiple samples for each pixel and average them together → supersampling.
- Can weight them towards the centre → weighted average sampling
- Stochastic sampling

Removing aliasing is called antialiasing

Antialiasing Strategies

- Pixel needs to represent average color over its entire area
- Prefiltering
  - Averages the image function so a single sample represents the average color
  - Limits bandwidth of image signal to avoid overlap
- Supersampling
  - Supersampling averages together many samples over pixel area
  - Moves the spectral replicas farther apart in frequency domain to avoid overlap

Cone Tracing

- Amanatides SIGGRAPH 84
- Replace rays with cones
- Cone samples pixel area
- Intersect cone with objects
  - Analytic solution of cone-object intersection similar to ray-object intersection
  - Expensive

Beam Tracing

- Heckbert & Hanrahan SIGGRAPH 84
- Replace rays with generalized pyramids
- Intersection with polygonal scenes
  - Plane-plane intersections easy, fast
  - Existing scan conversion antialiasing
- Can perform some recursive beam tracing
  - Scene transformed to new viewpoint
  - Result clipped to reflective polygon

Covers
Supersampling
- Trace at higher resolution, average results
- Adaptive supersampling
  - trace at higher resolution only where necessary
- Problems
  - Does not eliminate aliases (e.g. moire patterns)
  - Makes aliases higher-frequency
  - Due to uniformity of samples

Stochastic Sampling
- Eye is extremely sensitive to patterns
- Remove pattern from sampling
- Randomize sampling pattern
- Result: pattern becomes noise
- Some noises better than others
- Jitter: Pick a random points in sample space
  - Easiest, but samples cluster
- Uniform Jitter: Subdivide sample space into \( n \) regions, and randomly sample in each region
  - Easier, but can still cluster
- Poisson Disk: Pick \( n \) random points, but not too close to each other
  - Samples can’t cluster, but may run out of room

Stochastic Sampling
- Poisson
- Poisson Disk
- Jitter

OpenGL Aliases
- Aliasing due to rasterization
- Opposite of ray casting
- New polygons-to-pixels strategies
- Prefiltering
  - Edge aliasing
    - Analytic Area Sampling
    - A-Buffer
  - Texture aliasing
    - MIP Mapping
    - Summed Area Tables
- Postfiltering
  - Accumulation Buffer

Analytic Area Sampling
- Ed Catmull, 1978
- Eliminates edge aliases
- Clip polygon to pixel boundary
- Sort fragments by depth
- Clip fragments against each other
- Scale color by visible area
- Sum scaled colors
### A-Buffer
- Loren Carpenter, 1984
- Subdivides pixel into 4x4 bitmasks
- Clipping = logical operations on bitmasks
- Bitmasks used as index to lookup table

![A-Buffer Diagram](image)

### Texture Aliasing
- Image mapped onto polygon
- Occur when screen resolution differs from texture resolution
- Magnification aliasing
  - Screen resolution finer than texture resolution
  - Multiple pixels per texel
- Minification aliasing
  - Screen resolution coarser than texture resolution
  - Multiple texels per pixel

### Magnification Filtering
- Nearest neighbor
  - Equivalent to spike filter
- Linear interpolation
  - Equivalent to box filter

![Magnification Filtering Diagram](image)

### Minification Filtering
- Multiple texels per pixel
- Potential for aliasing since texture signal bandwidth greater than framebuffer
- Box filtering requires averaging of texels
- Precomputation
  - MIP Mapping
  - Summed Area Tables

### MIP Mapping
- Lance Williams, 1983
- Create a resolution pyramid of textures
  - Repeatedly subsample texture at half resolution
  - Until single pixel
  - Need extra storage space
- Accessing
  - Use texture resolution closest to screen resolution
  - Or interpolate between two closest resolutions

![MIP Mapping Diagram](image)

### Summed Area Table
- Frank Crow, 1984
- Replaces texture map with summed-area texture map
  - \( S(x,y) = \) sum of texels \( \leq x,y \)
  - Need double range (e.g. 16 bit)
- Creation
  - Incremental sweep using previous computations
  - \( S(x,y) = T(x,y) + S(x-1,y) + S(x,y-1) - S(x-1,y-1) \)
- Accessing
  - \( \sum T([x_1,x_2],[y_1,y_2]) = S(x_2,y_2) - S(x_2,y_1) - S(x_1,y_2) + S(x_1,y_1) \)
  - \( \text{Ave} \, \sum T([x_1,x_2],[y_1,y_2]) = (x_2 - x_1)(y_2 - y_1) \)
Accumulation Buffer

- Increases OpenGL’s resolution
- Render the scene 16 times
- Shear projection matrices
- Samples in different location in pixel
- Average result
- Jittered, but same jitter sampling pattern in each pixel

Aliased vs. Antialiased

Polygon Edges

Line Antialiasing

Polygon Antialiasing

Close-up
Sample at 4x4 subpixels and color the pixel according to the portion of the coverage.
Texture Mapping

Two special cases:

- Magnification: No real need in prefiltering. The main decision is what kind of reconstruction (interpolation) to use.
- Minification: No real need in reconstruction. The main problem is proper prefiltering.
Nearest neighbor sampling

Filtered Texture:

Magnification Filtering
- Nearest neighbor
  - Equivalent to spike filter
- Linear interpolation
  - Equivalent to box filter

Texture Pre-Filtering
- Problem: filtering the texture during rendering is too slow for interactive performance.
- Solution: pre-filter the texture in advance
  - Summed area tables - gives the average value of each axis-aligned rectangle in texture space
  - Mip-maps (tri-linear interpolation) - supported by most of today’s texture mapping hardware

MIP-Maps
- Precompute a set of prefiltered textures (essentially an image pyramid).
- Based on the area of the pre-image of the pixel:
  - Select two “best” resolution levels
  - Use bilinear interpolation inside each level
  - Linearly interpolate the results
- Referred to as trilinear interpolation
MIP Mapping

• Lance Williams, 1983
• Create a resolution pyramid of textures
  – Repeatedly subsample texture at half resolution
  – Until single pixel
• Accessing
  – Use texture resolution closest to screen resolution
  – Or interpolate between two closest resolutions

Texture Aliasing

• Image mapped onto polygon
• Occur when screen resolution differs from texture resolution
• Magnification aliasing
  – Screen resolution finer than texture resolution
  – Multiple pixels per texel
• Minification aliasing
  – Screen resolution coarser than texture resolution
  – Multiple texels per pixel

Minification Filtering

• Multiple texels per pixel
• Potential for aliasing since texture signal bandwidth greater than framebuffer
• Box filtering requires averaging of texels
• Precomputation
  – MIP Mapping
  – Summed Area Tables

Summed Area Table

• Frank Crow, 1984
• Replaces texture map with summed-area texture map
  – \( S(x,y) = \text{sum of texels } \leq x,y \)
  – Need double range (e.g. 16 bit)
• Creation
  – Incremental sweep using previous computations
    \( S(x,y) = T(x,y) + S(x-1,y) + S(x,y-1) - S(x-1,y-1) \)
• Accessing
  – \( \sum T([x_1,x_2],[y_1,y_2]) = S(x_2,y_2) - S(x_1,y_2) - S(x_2,y_1) + S(x_1,y_1) \)
  – \( \text{Ave } T([x_1,x_2],[y_1,y_2]) = \frac{(x_2-x_1)(y_2-y_1)}{((x_2-x_1)(y_2-y_1))} \)

Summed Area Tables

• A 2D table the size of the texture. At each entry \((i,j)\), store the sum of all texels in the rectangle defined by \((0,0)\) and \((i,j)\).
• Given any axis aligned rectangle, the sum of all texels is easily obtained from the summed area table:
  \[
  \text{area} = A - B - C + D
  \]

Quality considerations

• Pixel area maps to “weird” (warped) shape in texture space
Mip-maps

• Find level of the mip-map where the area of each mip-map pixel is closest to the area of the mapped pixel.

2x2 pixels level selected

Summed Area Table (SAT)

• Determining the rectangle:
  – Find bounding box and calculate its aspect ratio

Elliptical Weighted Average (EWA) Filter

• Treat each pixel as circular, rather than square.
• Mapping of a circle is elliptical in texel space.

Texture Domain

Elliptical Weighted Average

Summed Area Table (SAT)

• Determine the rectangle with the same aspect ratio as the bounding box and the same area as the pixel mapping.
Accumulation Buffer

- Increases OpenGL’s resolution
- Render the scene 16 times
- Shear projection matrices
- Samples in different location in pixel
- Average result
- Jittered, but same jitter sampling pattern in each pixel

tests