Prefiltered Antialiased Lines Using Half-Plane Distance Functions

Joel McCormack, Bob McNamara &
Norman P. Jouppi

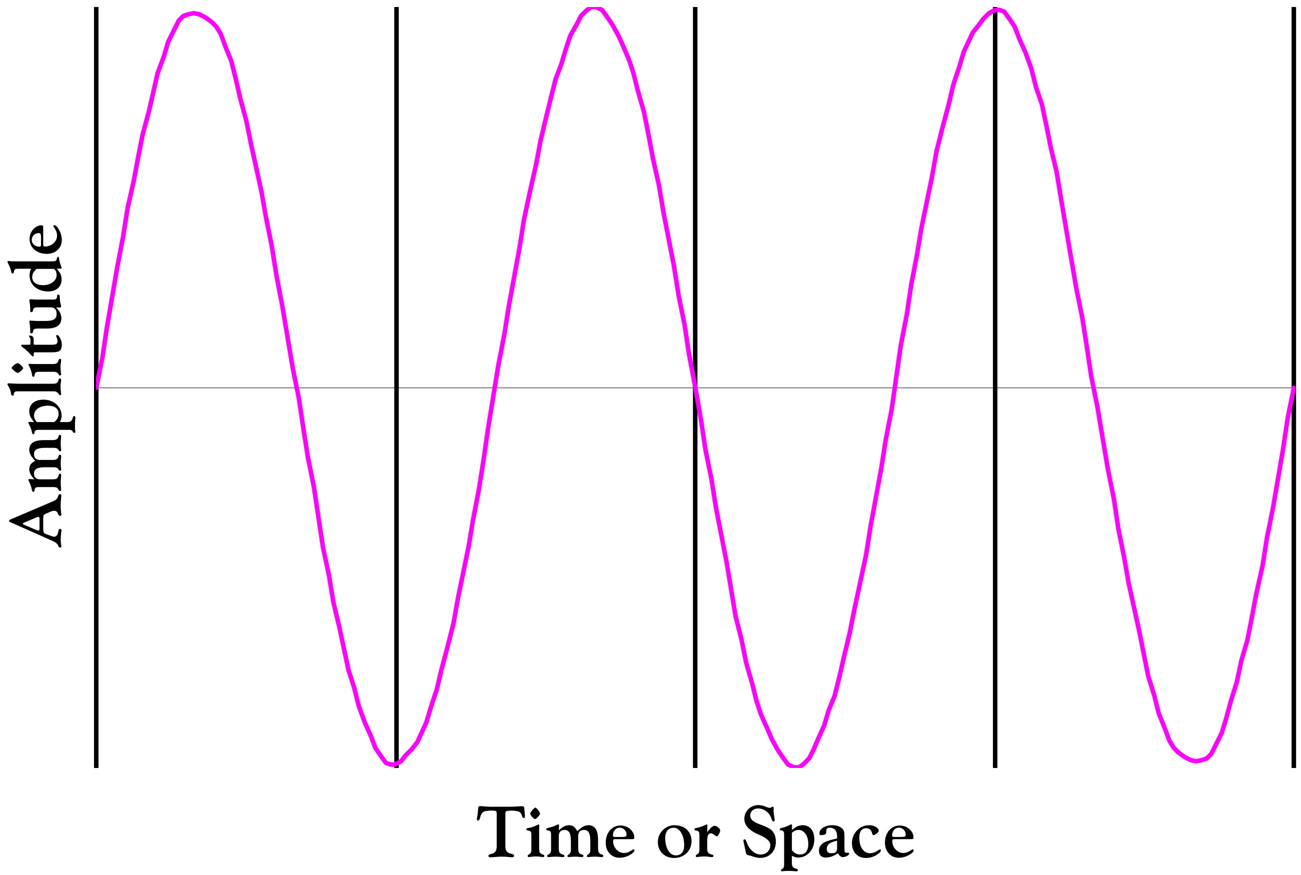
Motivation

- Simple (one sample/pixel) frame buffer
- Beautiful antialiased lines
- Mesh well with half-plane equation based rasterizer

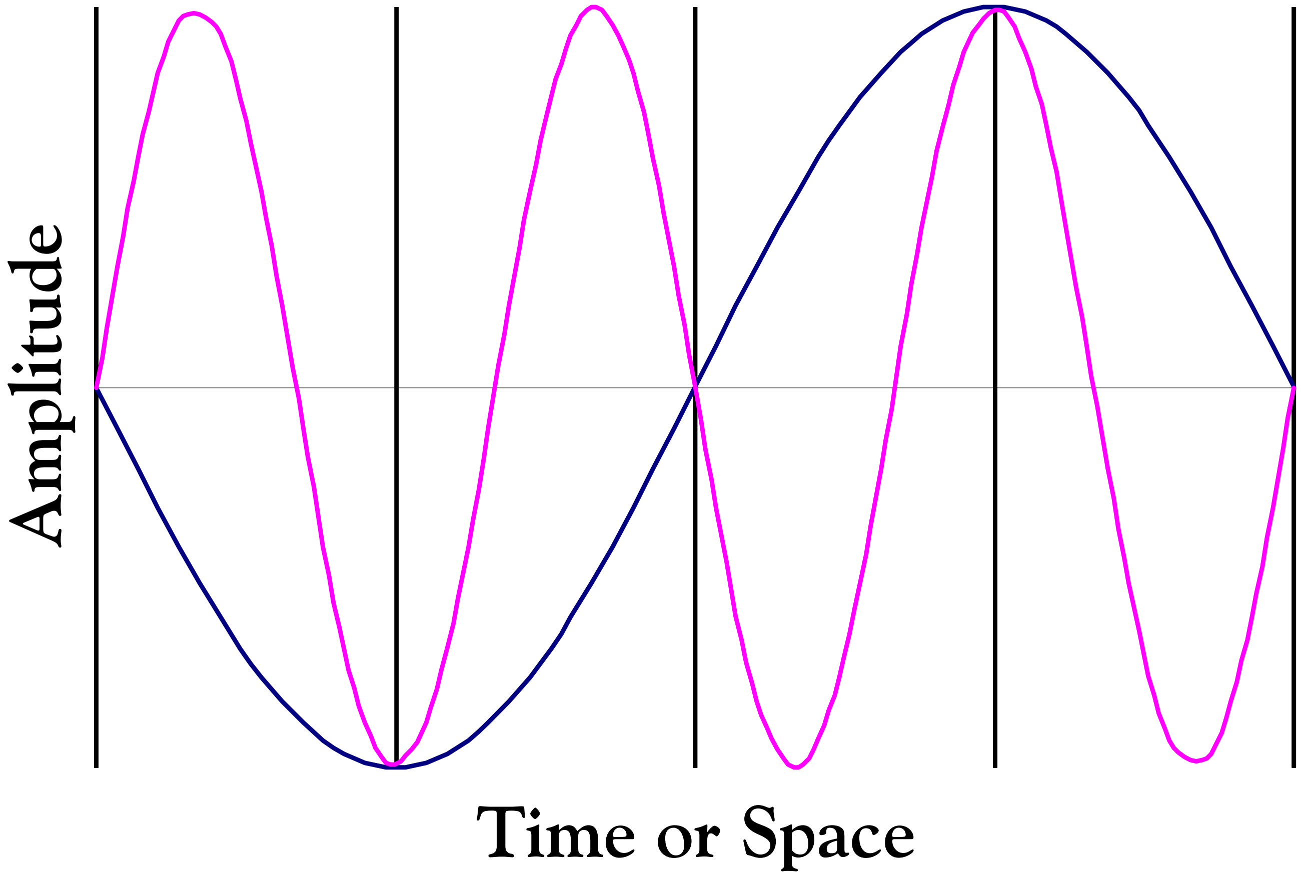
Outline

- Aliasing
- Filtering
- Prefiltering
- Edge and distance functions
- Distance to intensity mapping
- Image comparisons
- Conclusions

Aliasing: Insufficiently Sampled High Frequency…

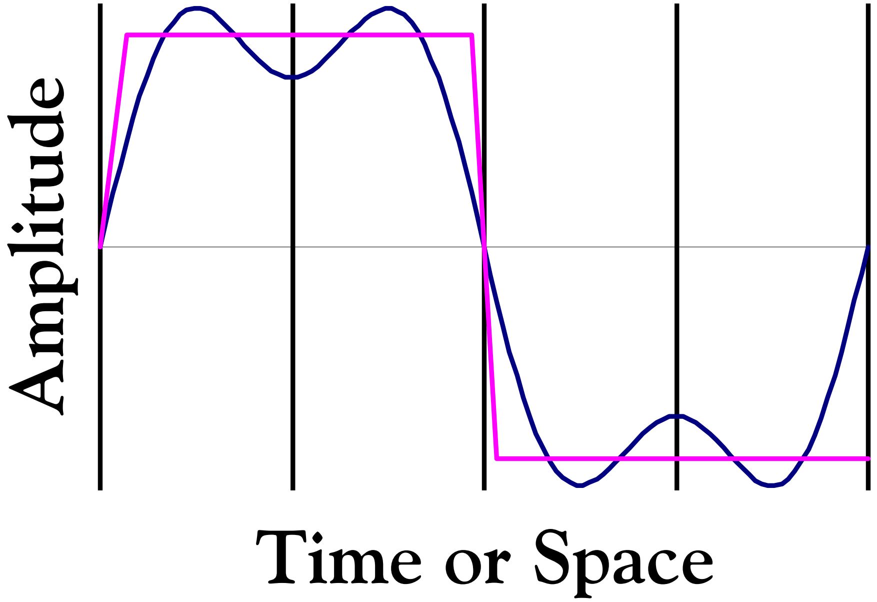


…Gets Reconstructed (Aliases) As Lower Frequency



Aliasing: Visual Artifacts

- Sharp edges require infinite frequency response
  - But… finite number of pixels limits frequency response
  - Reconstruction filter further limits frequency response
- High frequencies alias as stairstepped lines
- Solution: filter out high frequencies *before* sampling
- Cost: blurring

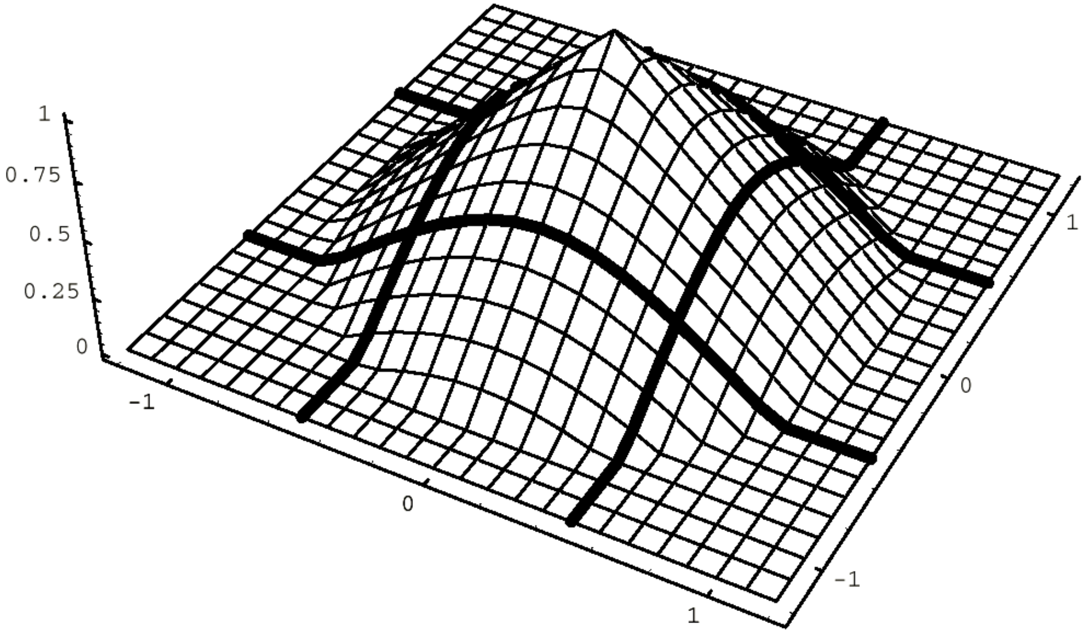


Filtering: Theory vs. Reality

- Mathematically "ideal" filter:
  - Removes all irreproducible frequencies (else aliasing)
  - Leaves all reproducible frequencies (else blurring)
  - Creates visible "ghosts" (ringing)
  - Requires infinite extent
- Realistic filter:
  - Strongly attenuates irreproducible frequencies
  - Weakly attenuates reproducible frequencies near cut-off
  - Small footprint, so efficient

Filtering: A Simple Filter

- $\mathit{weight}(x, y) = \max(1 - \mathrm{sqrt}(x^2 + y^2), 0)$
- Footprint radius of 1 pixel
- Compromise between blurring, aliasing, and efficiency
- Much better than hardware super-sampling box filter with ½ pixel radius

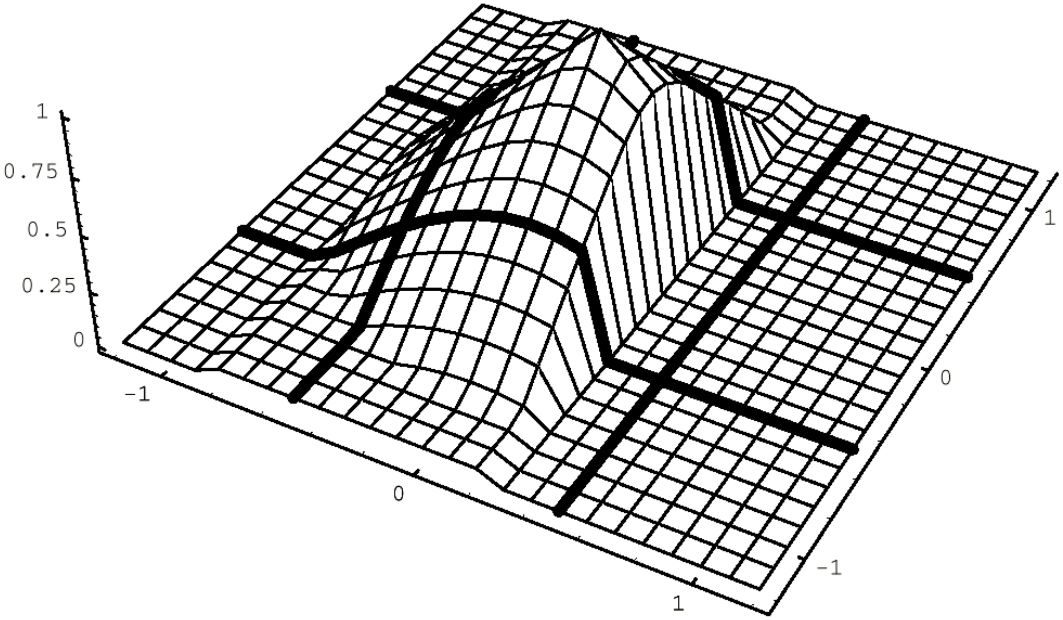


Filtering: Convolution With Object

- To compute a pixel's filtered (*red*, *green*, *blue*):
  - Place (*x*, *y*, *weight*) filter on pixel center
  - At each point in (*x*, *y*, *red*, *green*, *blue*)scene, multiply filter weight by object colors
  - Integrate each set of weighted colors

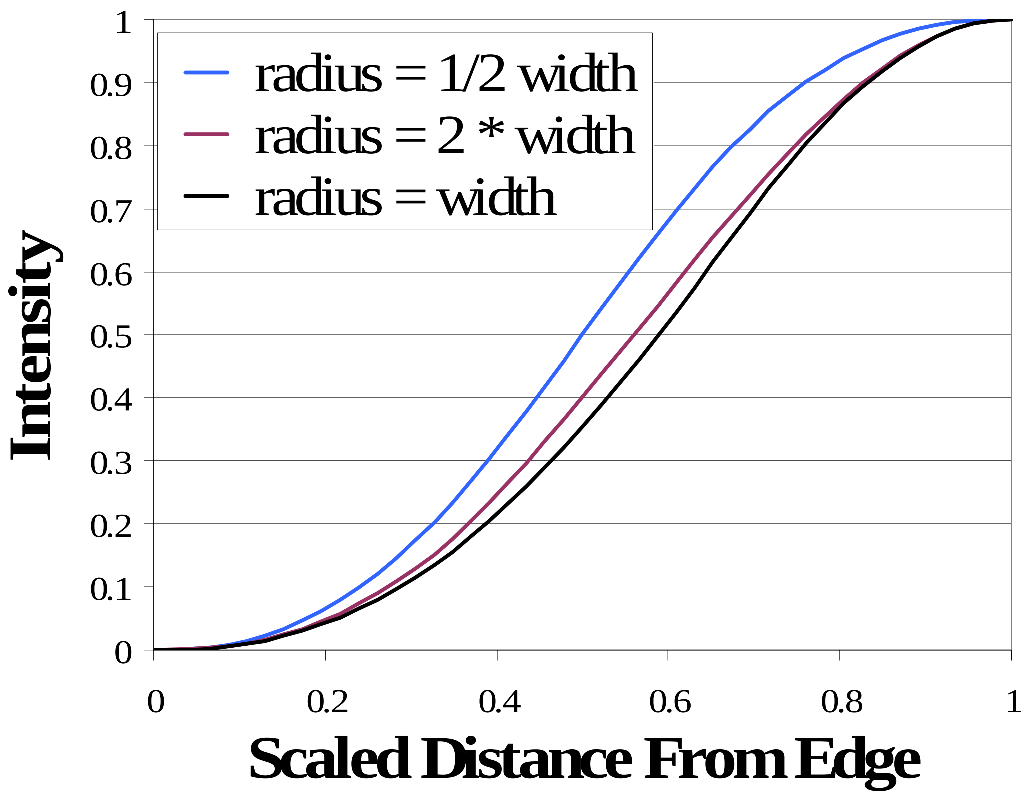
Filtering: Gross Simplifications

- Assume background color constant (no other objects)
  - Intersections with other objects may look weird
- Assume line color constant (*x*, *y*, 1)
  - Endpoint colors of depth-cued lines will be slightly off
- Convolution reduces to computing intersection volume of filter and line



Simplifications Allow Prefiltering

- Distance from pixel center to line edge determines volume
- Construct distance-to-intensity table
- Table depends upon ratio of filter radius to line width

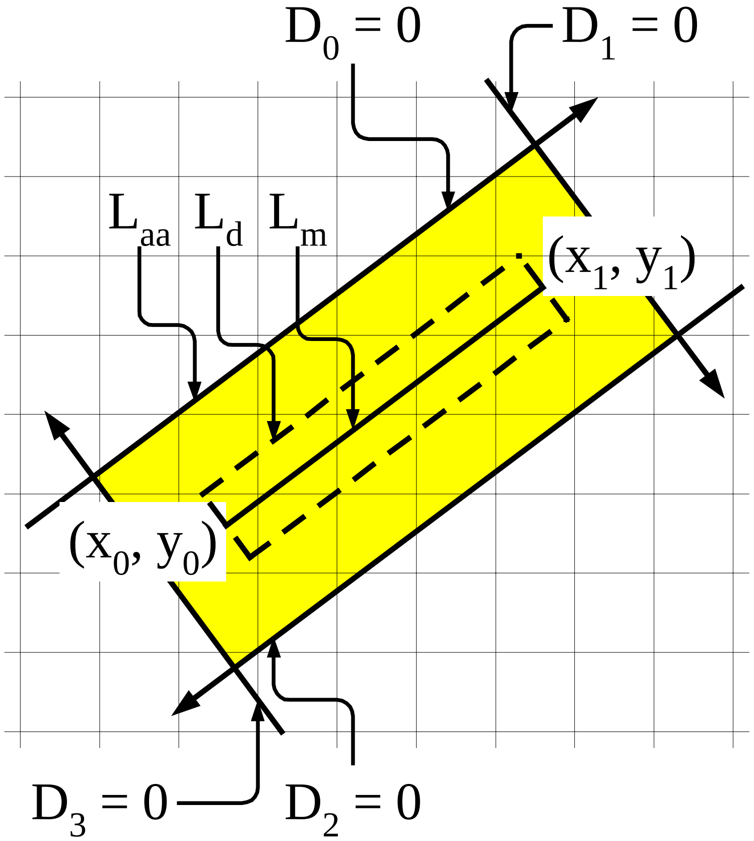


Edge Functions

- Many rasterizers traverse polygons by surrounding them with edge functions
- $E_{01}(x, y) = (x - x_0)(y_1 - y_0) - (y - y_0)(x_1 - x_0)$ splits 2D plane into two halves:
  - Non-negative on or to right of edge
  - Negative to left of edge
- If all edge functions at (x, y) have same sign, then (x, y) is within polygon

Surrounding An Antialiased Line with Edge Functions

- Construct L_m from supplied endpoints (x_0, y_0) and (x_1, y_1)
- Desired line L_d sides pushed out by ½ line width from L_m
- Antialiased line L_{aa} pushed out by filter radius from L_d

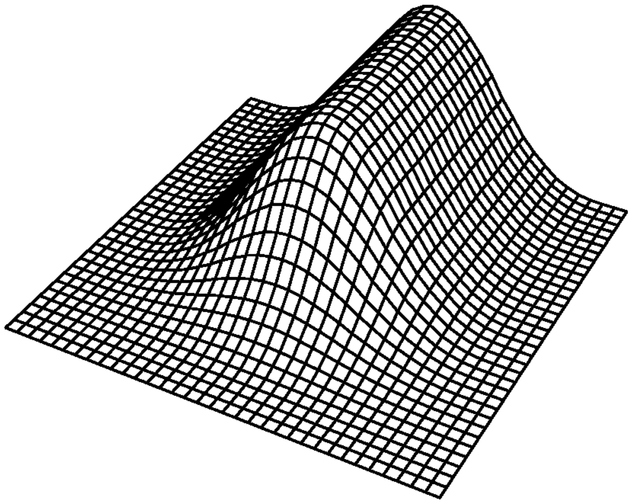


Distance Functions

- $D_{01}(x, y) = E_{01}(x, y) / \mathrm{sqrt}((x_1 - x_0)^2 + (y_1 - y_0)^2)$
- Implement as multiply by reciprocal root
- Further scale $D(x, y)$ so distance in [0..1] maps to intensity of [0..1]
- This affects only *setup* of edge functions…traversal algorithm is unchanged

But What About Line Ends?

- Distance to intensity table only works in middle of line
- Pixels near line ends can use:
  - Product of intensities from side and end distances, or
  - Two-dimensional distance to intensity table



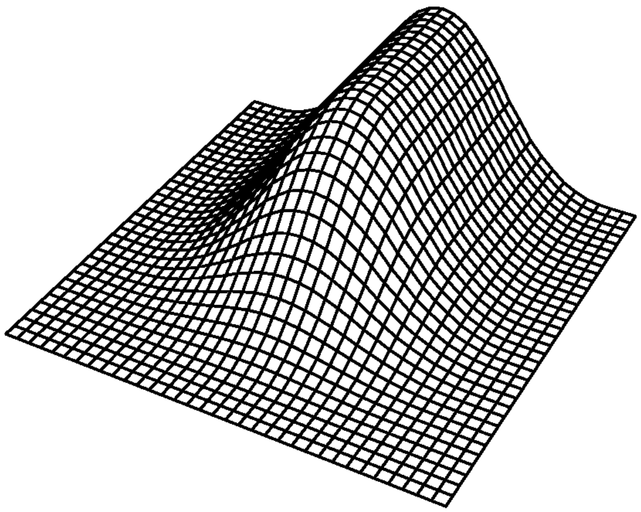
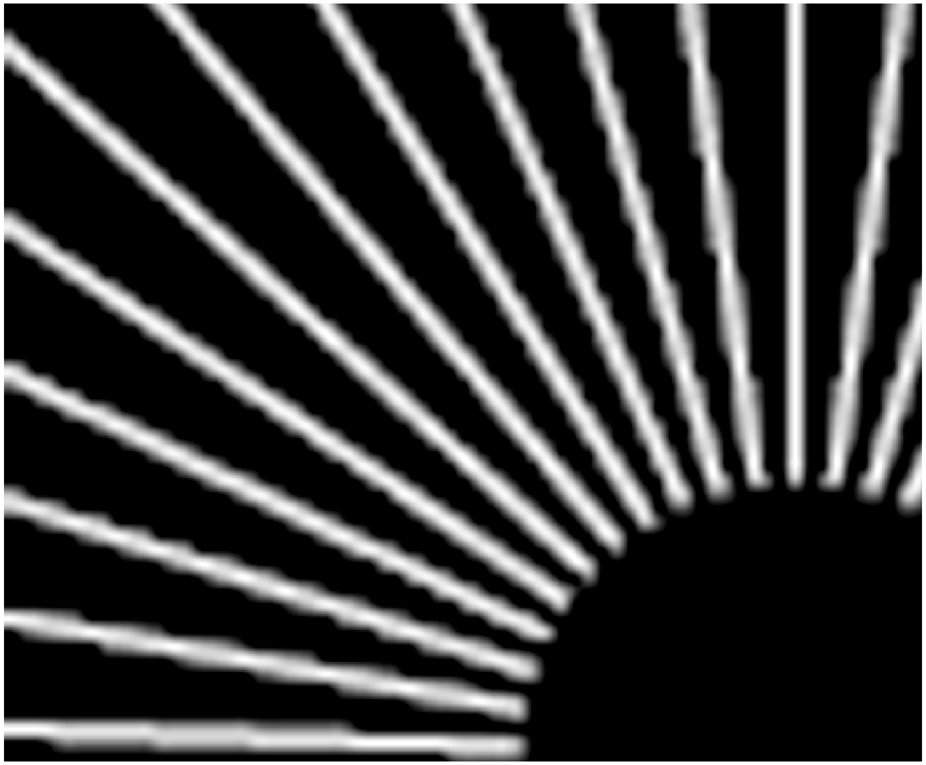
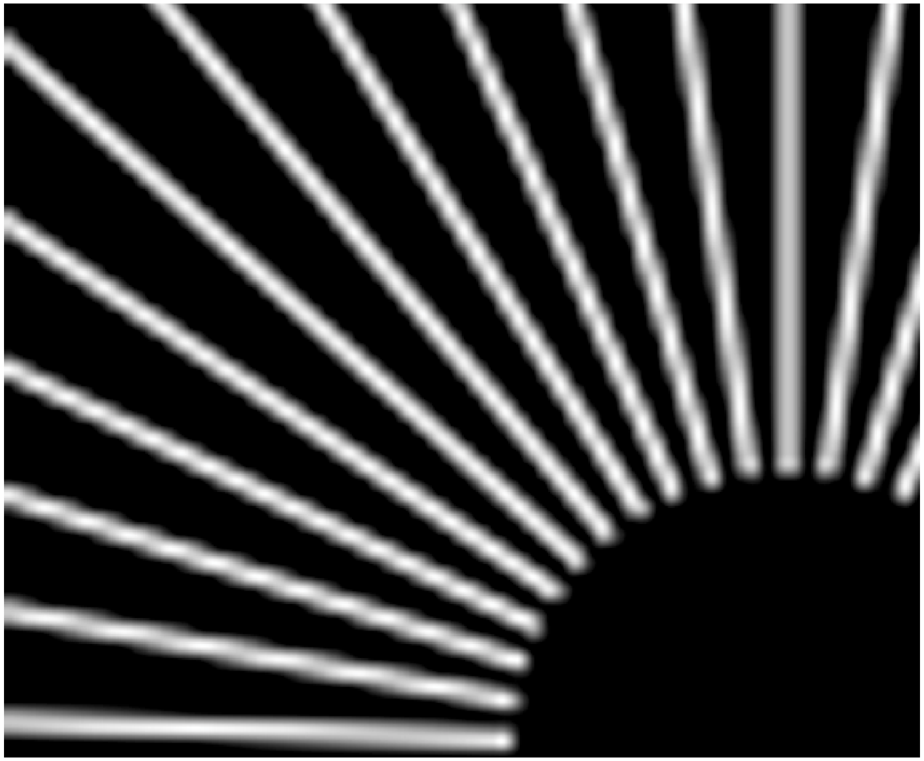


Image Comparisons: Black & White Stairstepping

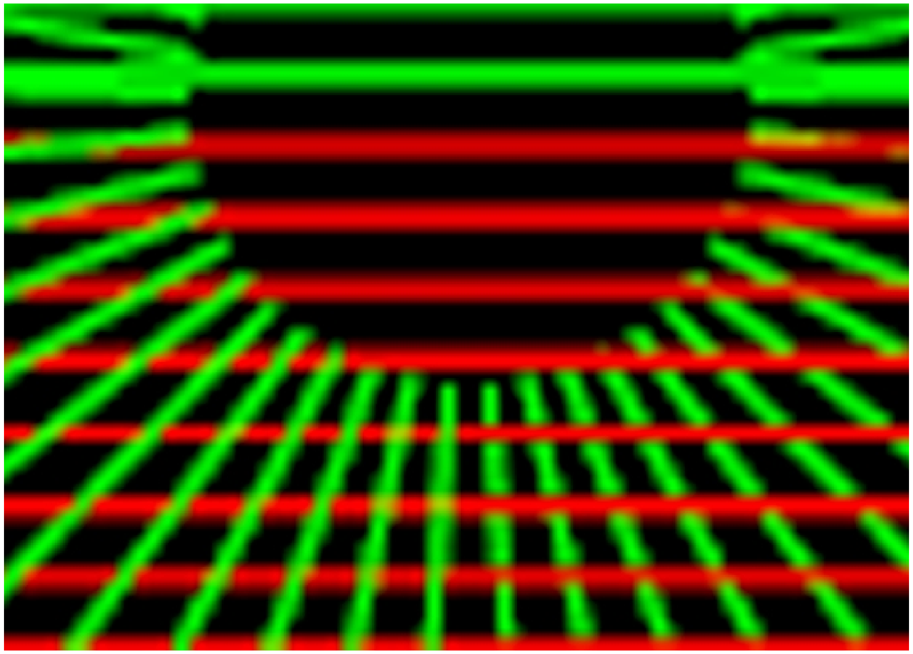


4x4 supersampling

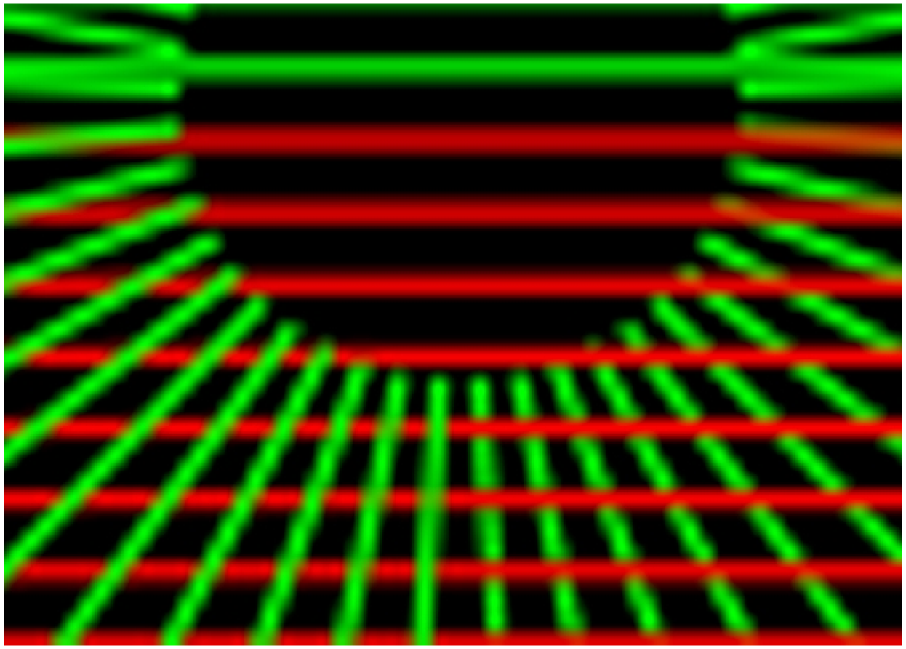


Prefiltering

Image Comparisons: Color Intersections



4x4 supersampling



Prefiltering

Conclusions

- Prefiltering advantages:
  - Fewer aliasing artifacts
  - 1/8 supersample memory storage and bandwidth
- Supersampling advantages:
  - Maintains near/far information for different colored lines
- Maybe someday…
  - Supersampling with >½ pixel radius
  - Supersampling with weighted samples
  - Check out Norm's Z^3 paper last year for cheap(er) supersampling