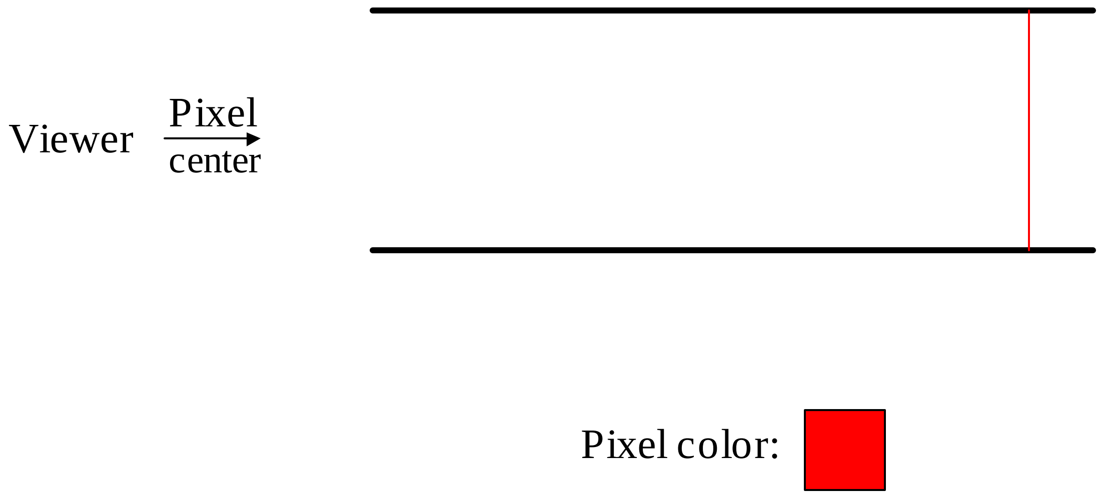
Z^3: An Economical Hardware Technique for High-Quality Antialiasing and Order-Independent Transparency

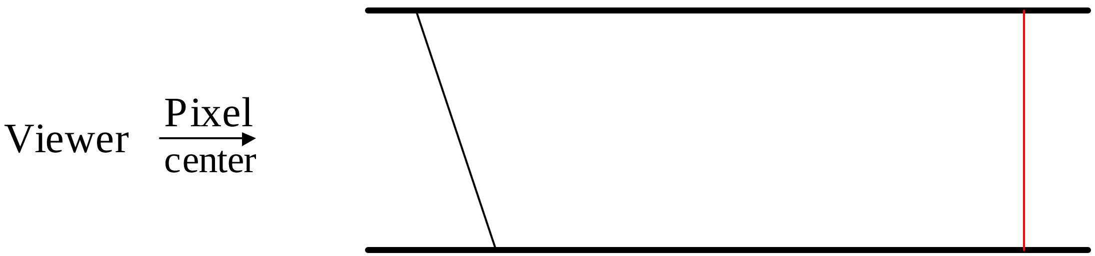
Norm Jouppi - Compaq WRL

Chun-Fa Chang - UNC Chapel Hill

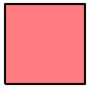
Order-Dependent Transparency



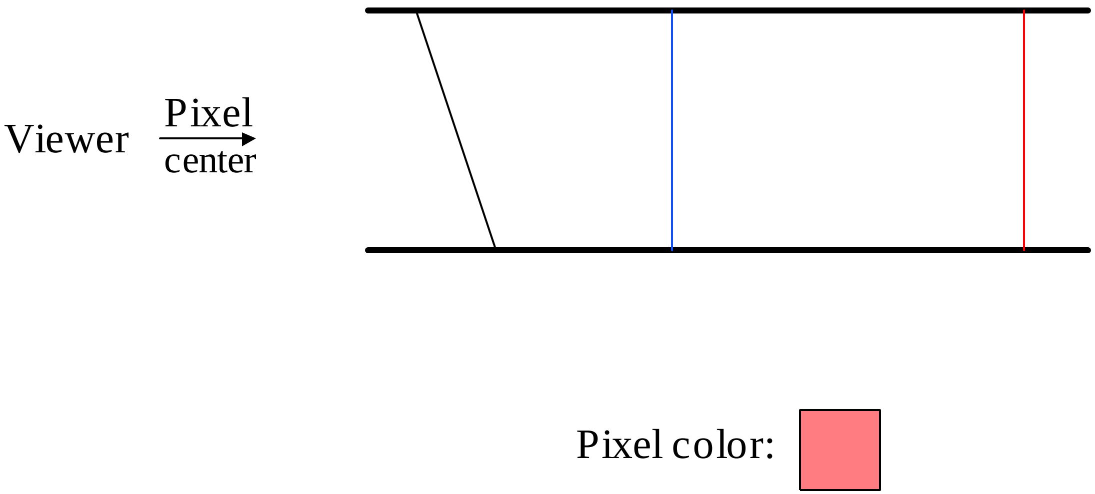
Order-Dependent Transparency



Pixel color:



Order-Dependent Transparency



Order-Independent (O-I) Transparency

- Render transparent & opaque objects in any order
  - “Don’t want to sort primitives”
  - Can’t sort subpixels
- Useful with textures (e.g., trees) and compositing
- David Kirk, NVIDIA, 1998 Eurographics/ SIGGRAPH keynote: Order-independent transparency is an “unsolved problem and opportunity for high-quality high-performance 3D graphics on a PC”

A-buffer Methods for O-I Transparency

- Developed by Carpenter as a software technique and enhanced by others
- Object-based algorithm:
  - Keep a list of each visible sub-pixel fragment
  - Compute final pixel from fragment list
- Implemented by some in hardware
  - Dynamic storage allocation in hardware!
- Doesn’t correctly antialias interpenetrating objects

How Can Aliasing Be Eliminated?

- Five main hardware methods:
  - Higher resolution monitors (impractical, expensive)
  - Blending (only works for lines)
  - Accumulation buffer (slow)
  - A-buffer methods
  - Supersampling

Supersampling

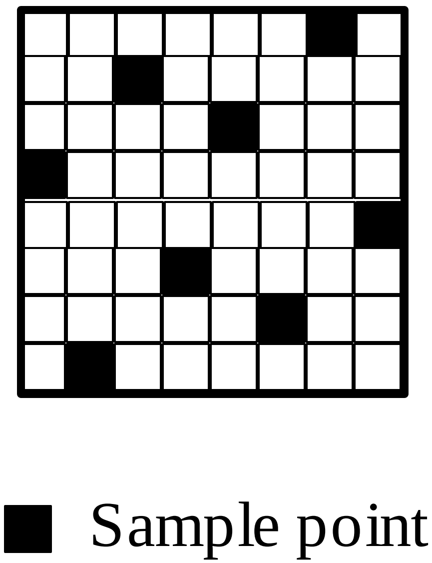
- Transparent to the user:
  - Render the image at higher resolution
  - Filter down to screen resolution
  - Requires more memory and time
  - Need at least 4X resolution in x and y to look significantly better
- Correctly handles interpenetrating opaque objects

Sparse Supersampling

- Have larger sample array, but sparsely populated
  - For various *n*, consider *n* samples on *n*x*n* grid
- Can give more intensity steps with fewer samples
  - Better images
  - Less time and storage

8x8 Sparse Supersampling

- One sample per row and column
- Near horizontal and vertical edges look better
  - 9 intensity steps with 8 samples
- Some other angles aren't as good
  - Diagonals already look better due to screen and eyes
- Looks almost as good as full 8x8
- Used in SGI's Infinite Reality



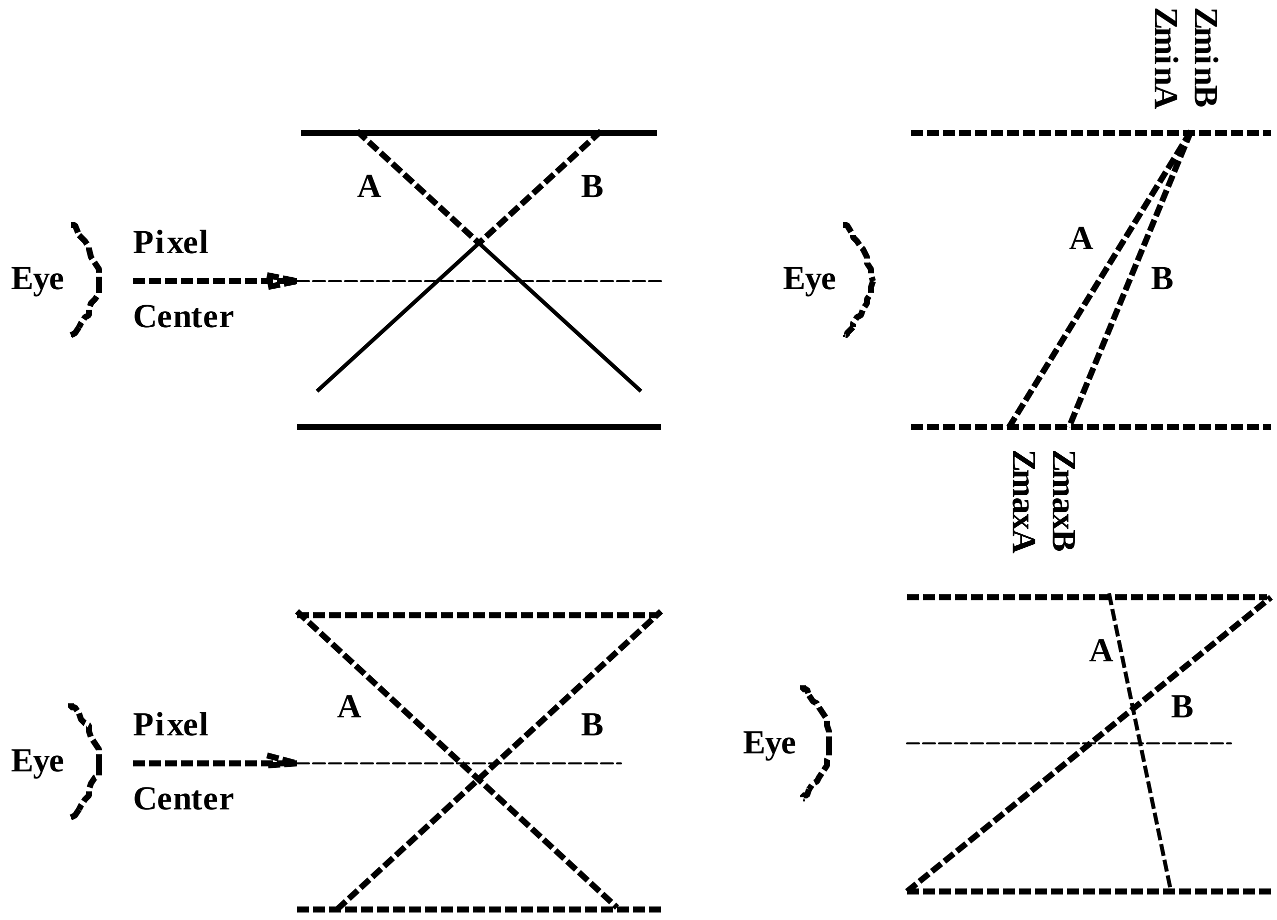
Problems with Sparse Supersampling

- Doesn’t support order-independent transparency
  - Sorting doesn’t fix interpenetrating transparency
- Uses too much memory capacity and bandwidth
- Wastes resources: Much of the sample point data is similar to data for other sample points
  - Often only a few objects are visible within a pixel
  - Use a single color for an entire fragment
  - Use a more compact Z representation

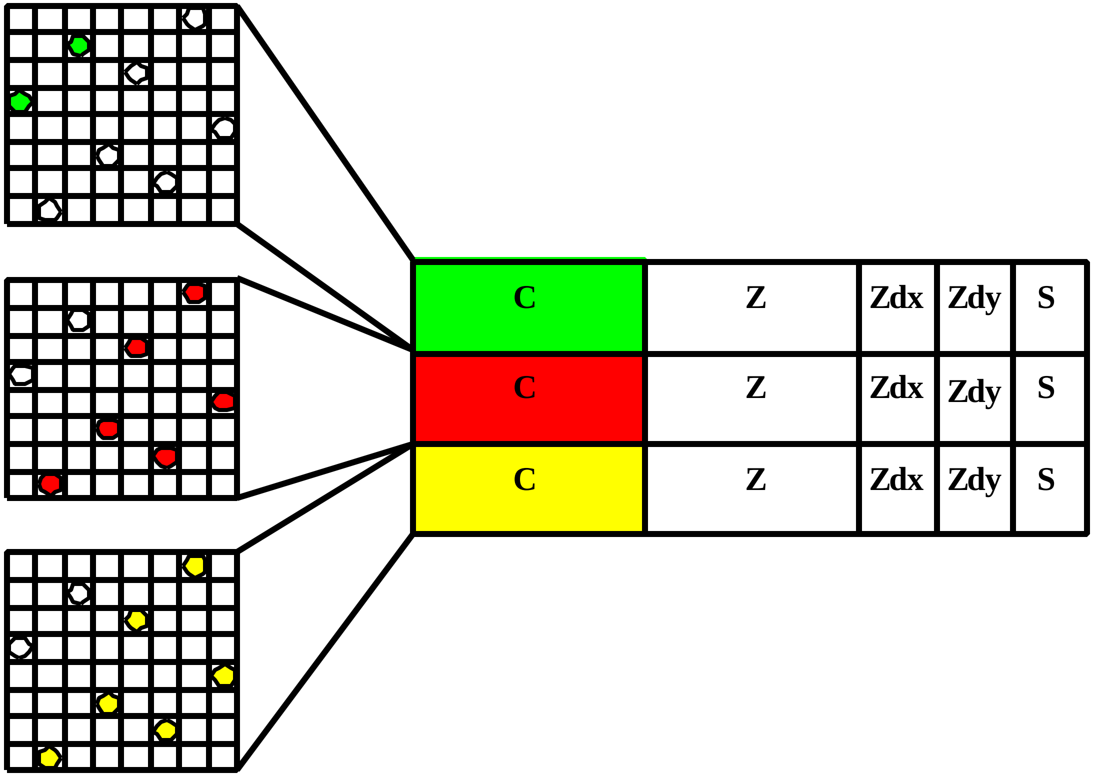
Compact Z Representations

- 4 options:
  - Single Z at pixel center
  - Zmin and Zmax
  - Centroid adjusted Z
  - Z, Zdx, and Zdy
- WARNING:
  Correct subpixel visibility calculations are more important than “correct” antialiasing of subpixels

Visibility Errors



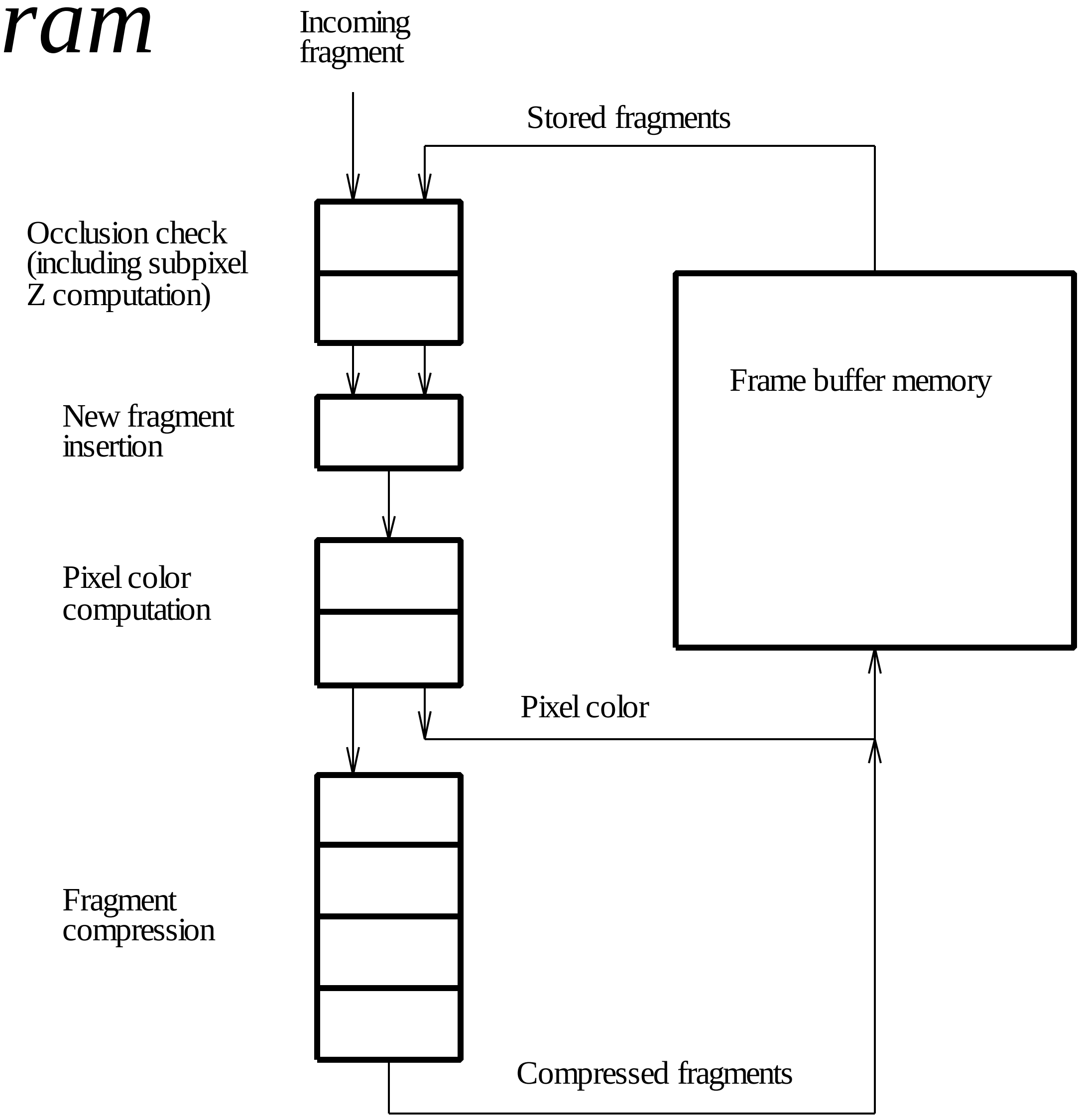
The Z3 Data Structure



Z^3 Algorithm

- 4 stages for processing a new fragment:
  - Occlusion check
  - If pass, insert fragment in Z order
  - Compute final pixel color
  - If too many fragments to fit, merge two fragments

Z^3 Block Diagram



Occlusion Check Stage

- Read in existing fragments sorted by center Z
- Expand Z gradients and Z values into per-sample Z values
- Standard occlusion test per sample
- Totally occluded fragments are deleted

Fragment Insertion Stage

- If not occluded, insert new fragment in proper place in sorted fragment list

Final Color Computation

- Only if new fragment is not fully occluded
- Compute per-sample ordering of fragment pairs
  - Produce a "swap vector" between fragments
- Accumulate fragments for each sample
- Average each sample into the final pixel color

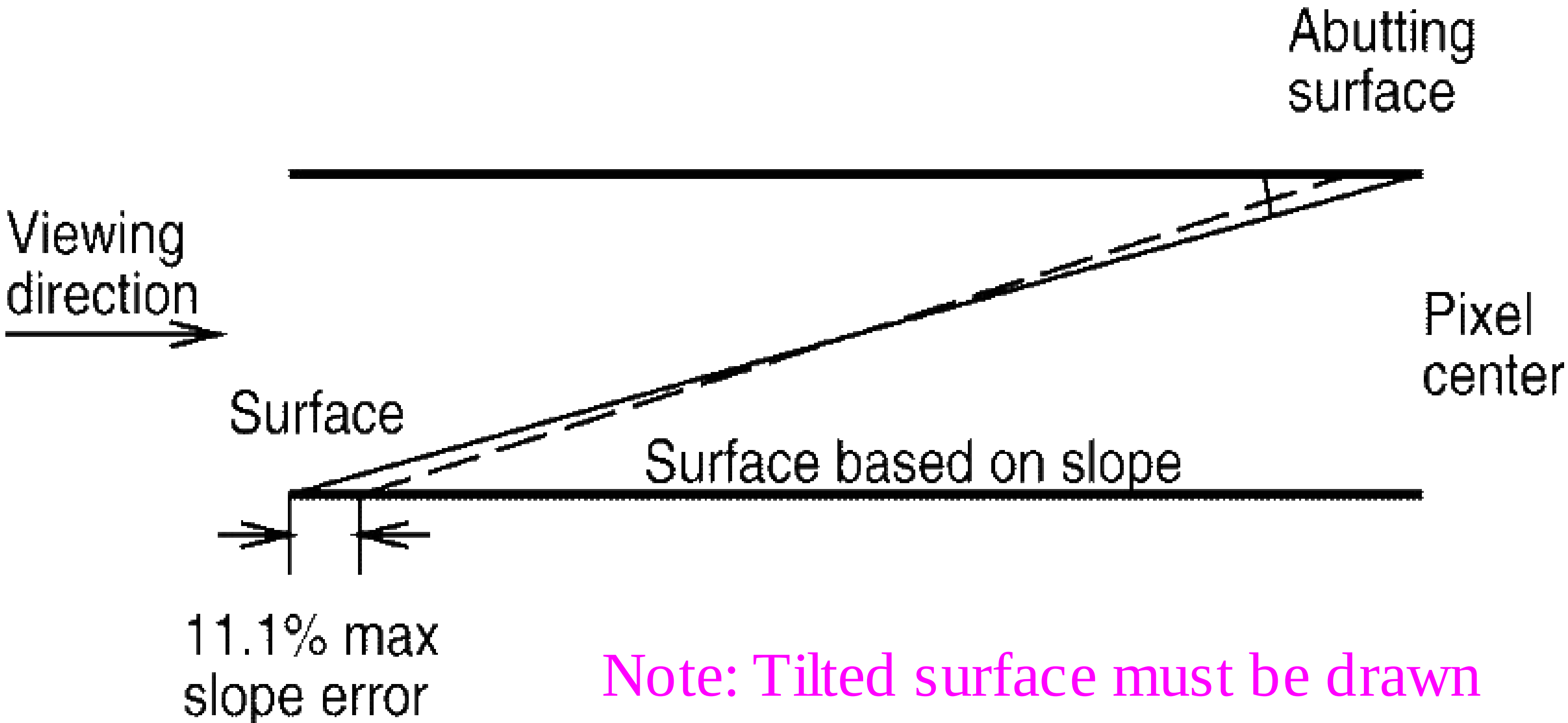
Fragment Merging

- Only if we have too many fragments
- Compute Z distance between pairs of fragments
- Merge closest pair of fragments
- Merges fragments from the same surface first, etc.

Limited Precision Z-slopes

- Compact 8-bit format:
  - 1-bit sign
  - 5-bit exponent
    - Can represent 2^{31} to 0
    - Covers entire range of 24-bit Z value
  - 3-bit mantissa (one bit is hidden)
- With rounding max error is 1 part in 9
  - .1001 is rounded up to .101
  - .1000111… is rounded down to .100

Maximum Potential Errors Due to Compact Slope Format

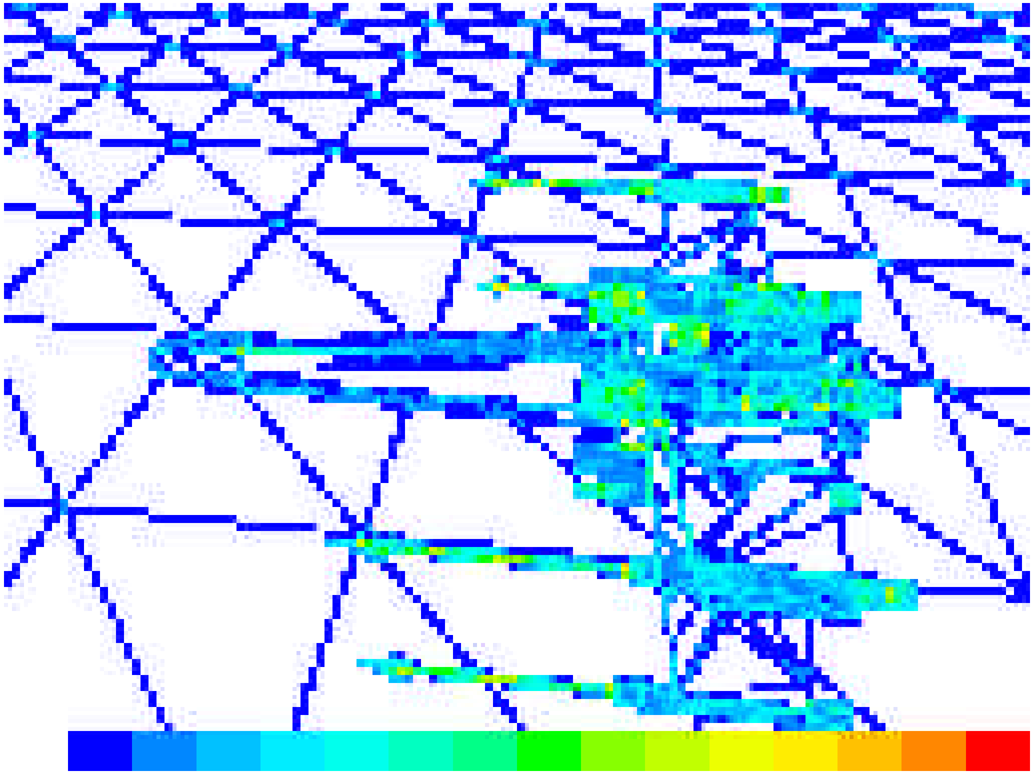


Note: Tilted surface must be drawn first to produce error

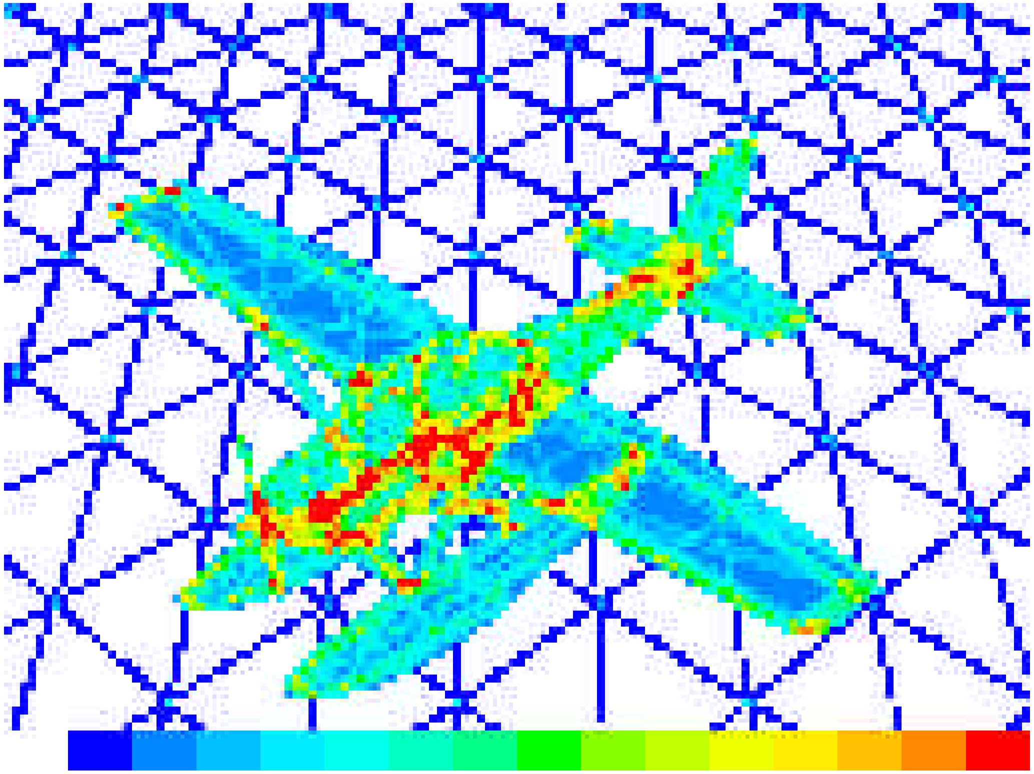
Reduced Precision Slopes: The Bottom Line

- May misplace edges by a sample point
  - Pixel could be 15/16 instead of 14/16
- Traditional 4X sparse supersampling is worse
  - Pixel limited to 3/4 or 4/4 instead of 14/16

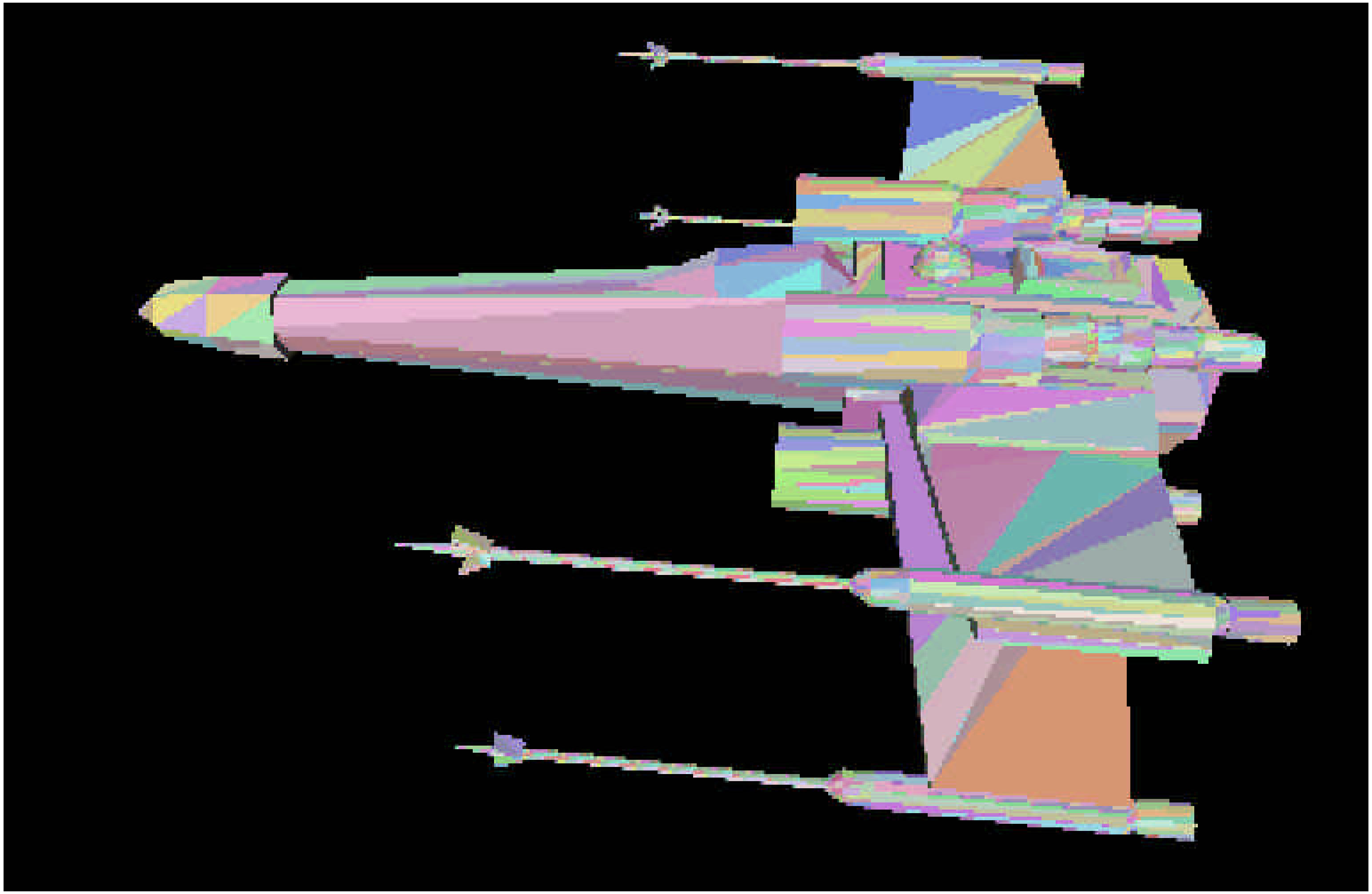
Pixel Complexity of Opaque Test Case



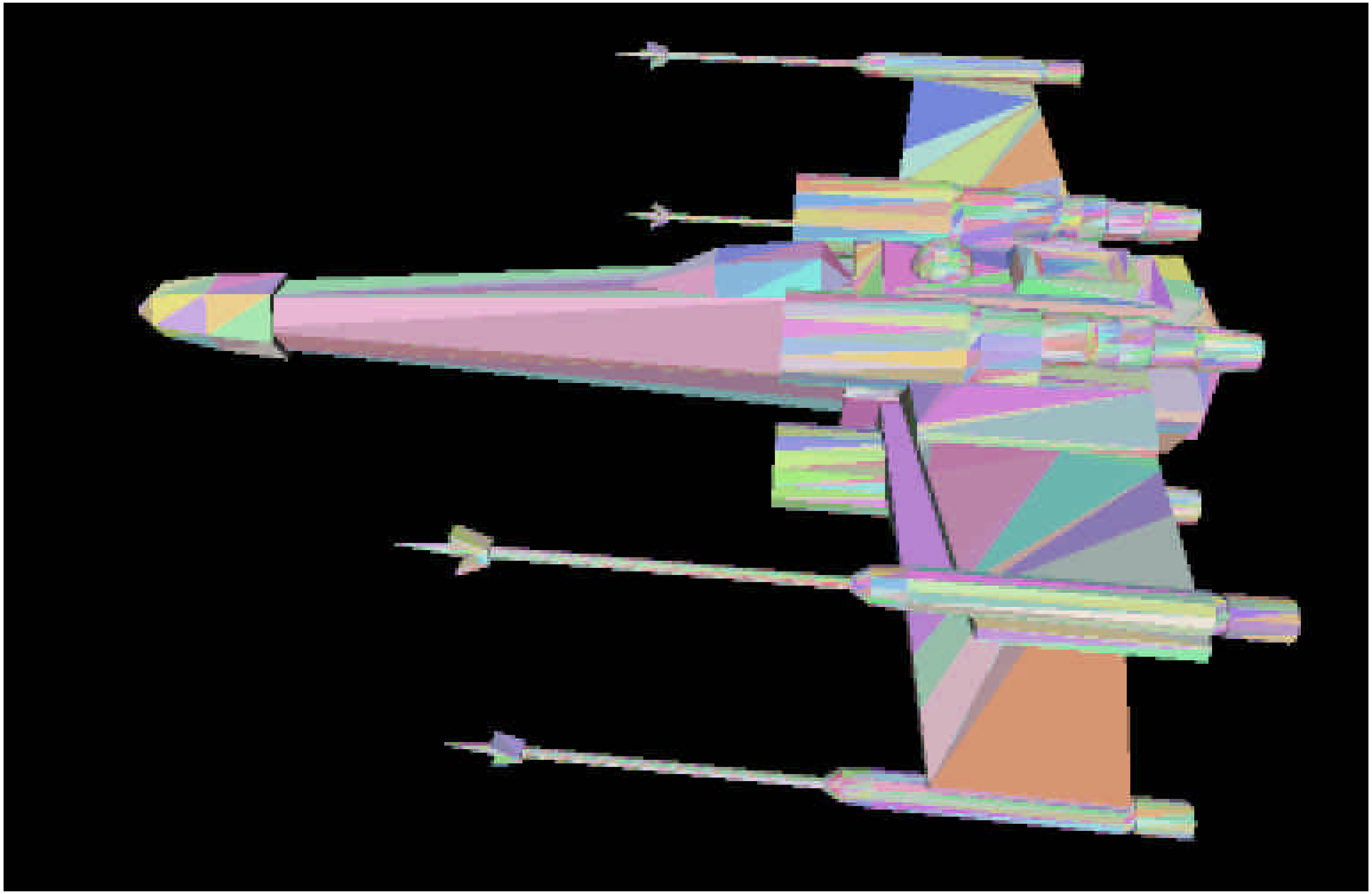
Complexity of Transparent Test Case



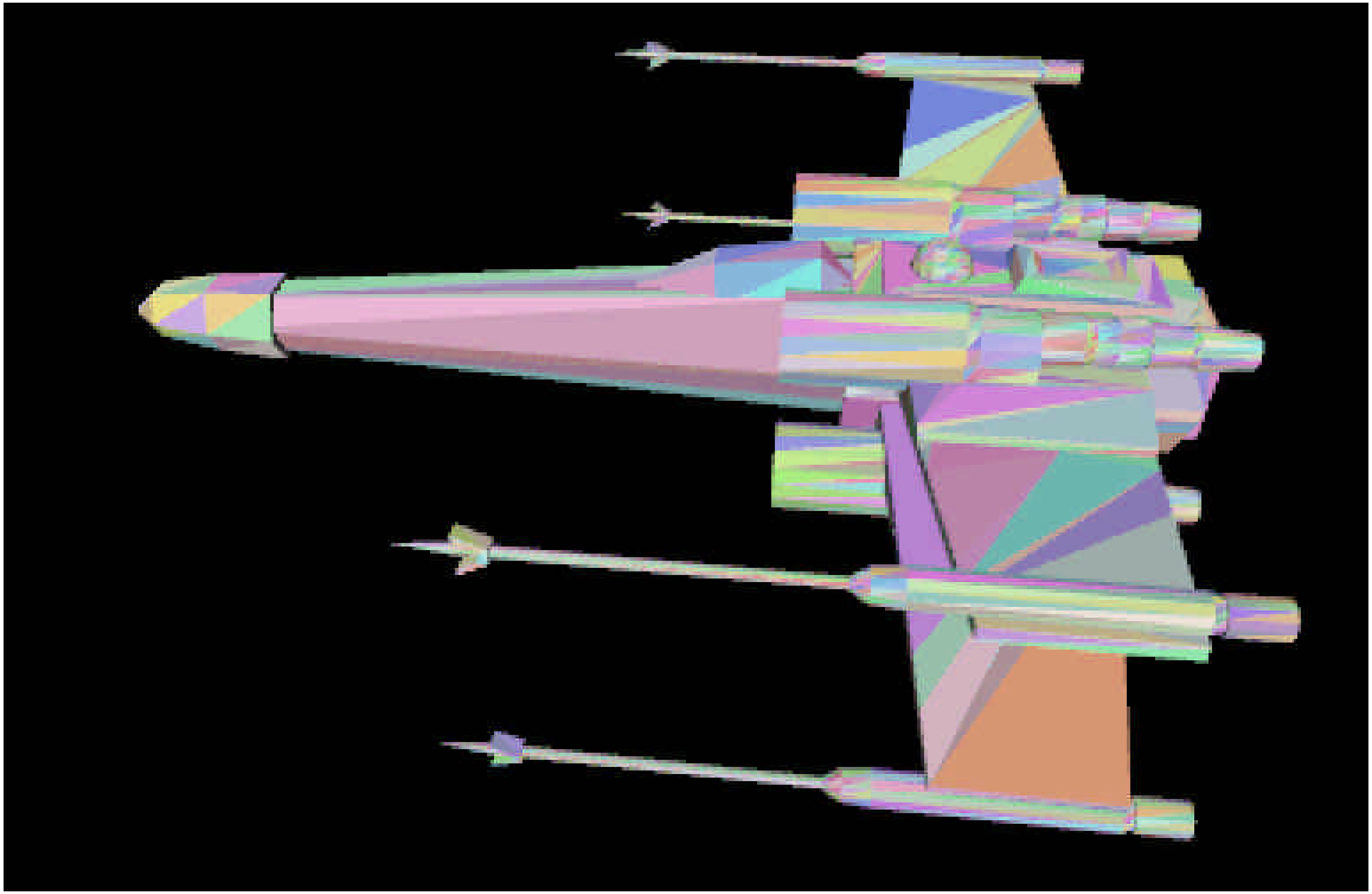
Original Aliased Image



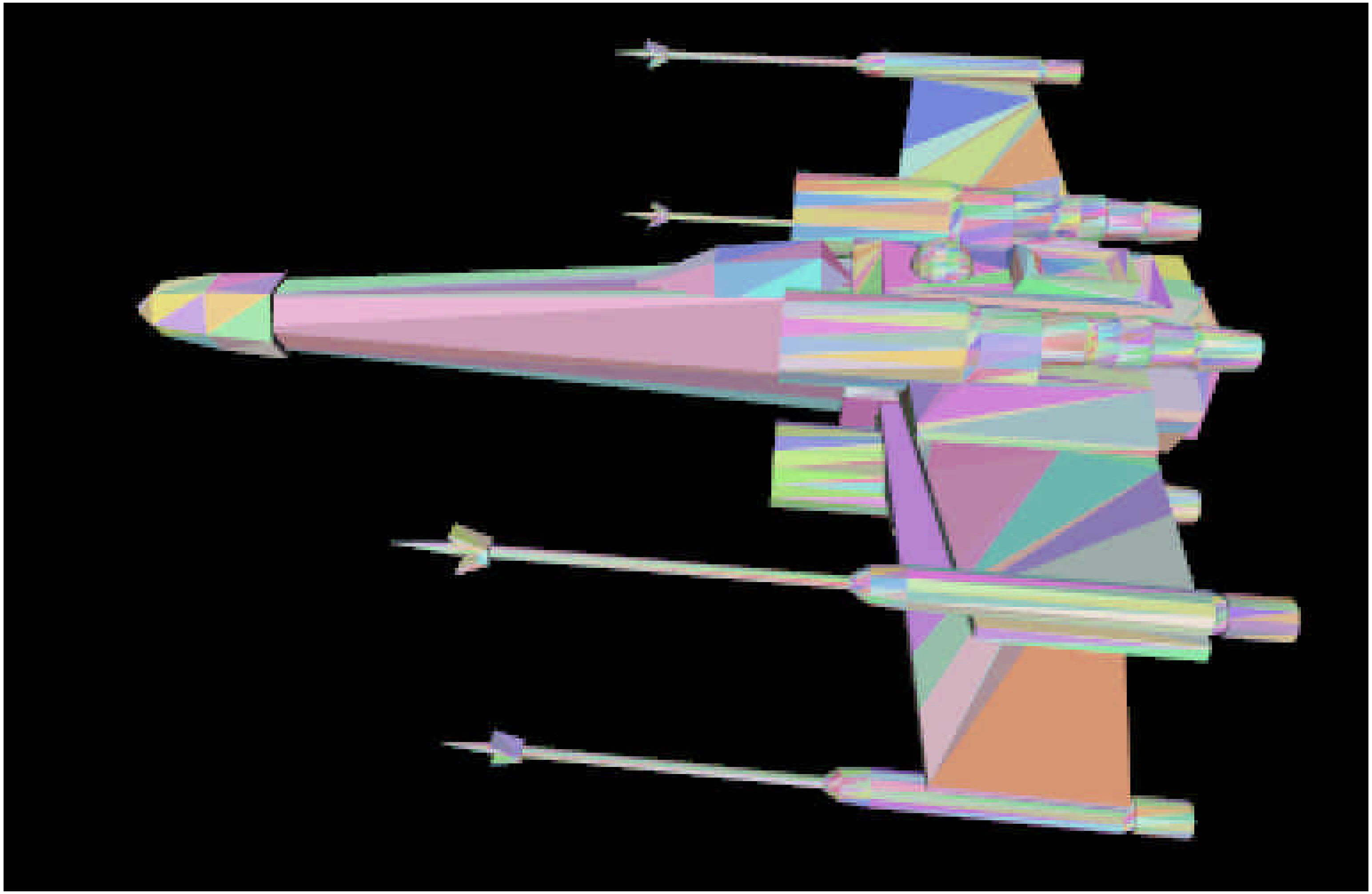
2x2 "Antialiasing"



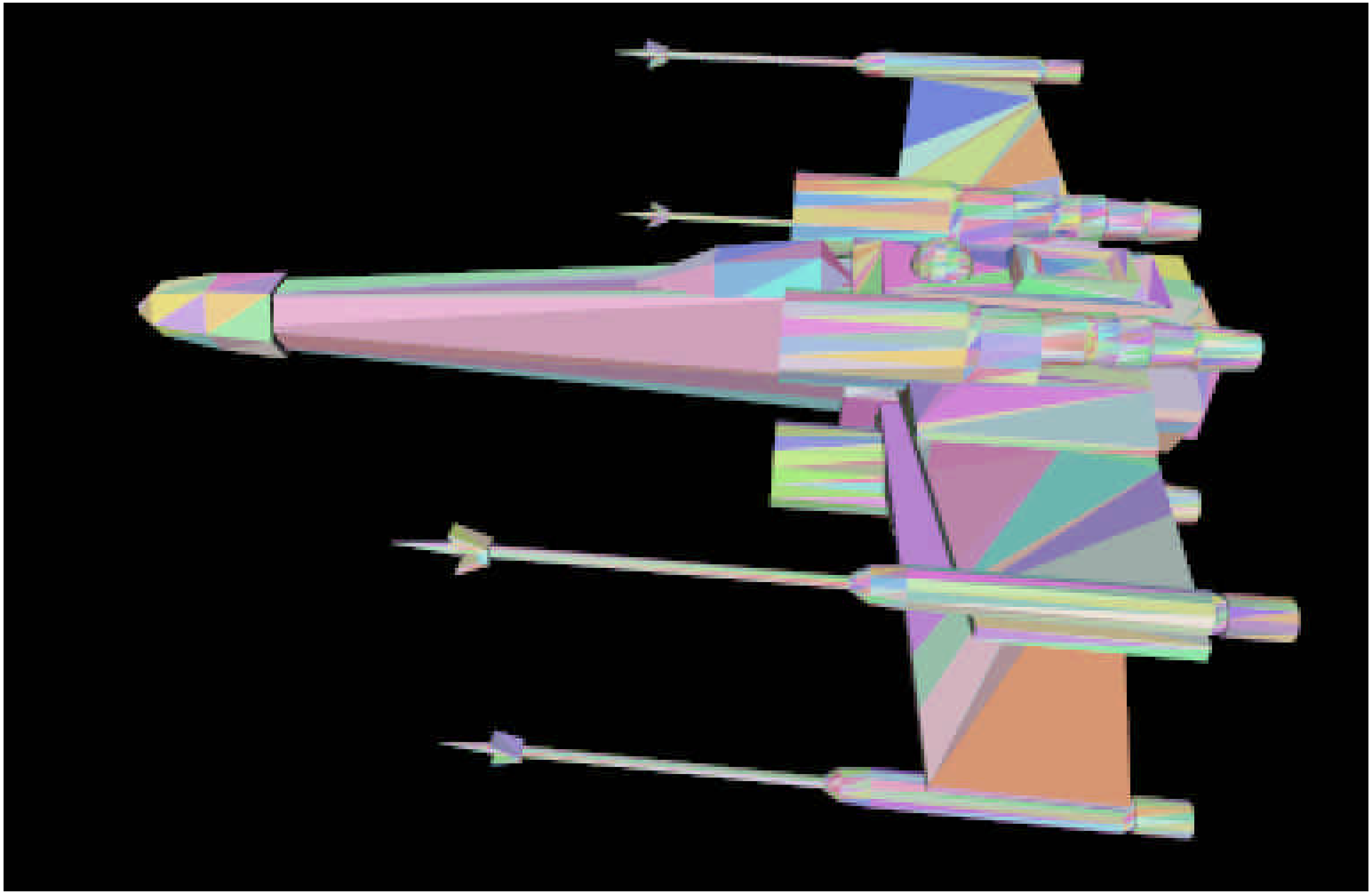
Low-end SGI Infinite Reality



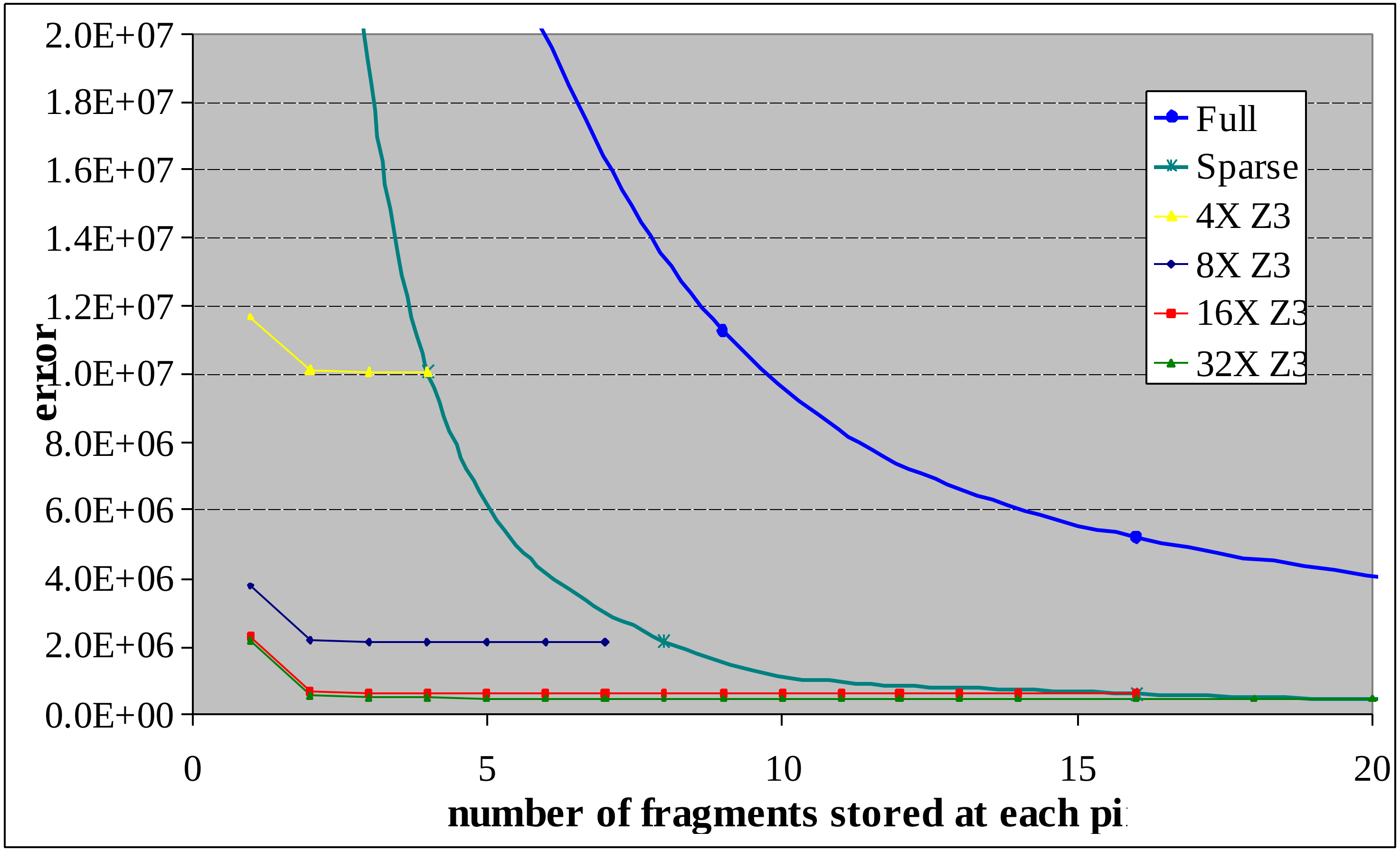
High-end Infinite Reality



16x16 Sparse Z3



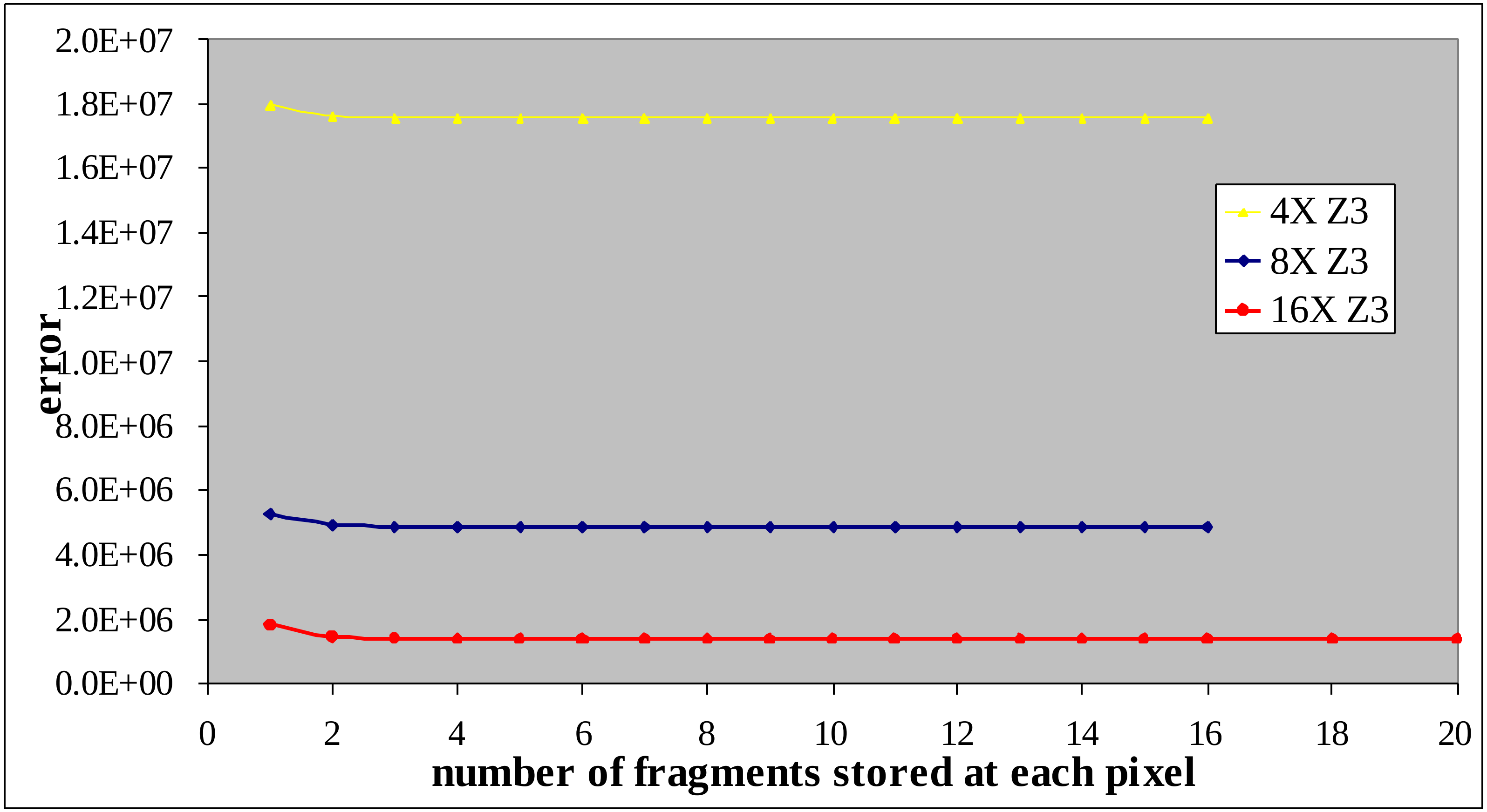
Sum of Squares of Per-Pixel Errors for Opaque Test Case vs. # of Fragments



Video

Also available at
http://www.research.digital.com/wrl/people/jouppi/Z3.html

Errors for Transparent Test Case vs. # of Fragments



Additional Memory Requirements

Screen size	2-fragment pixels	4-fragment pixels
1024x768	14MB	28MB
1280x1024	24MB	48MB
1600x1200	35MB	70MB
1920x1200	41MB	82MB

Note: Memory is currently about 75¢ a MB

Memory gets 4X cheaper every 3 years

Conclusions

- Z^3 provides high-quality antialiasing *and* order-independent transparency at small additional cost
- Easy to implement due to fixed per-pixel storage
- Large numbers of sample points (e.g., 16) feasible
- Correctly antialiases interpentrating surfaces, even if they are transparent (unlike A-buffer)
- Z^3's smaller memory requirements mean higher performance for a given memory bandwidth