Linear-time CSG Rendering of Intersected Convex Objects

Nigel Stewart, Geoff Leach and Sabu John

RMIT School of Computer Science and Information Technology

WSCG 2002 - Pilzen, Czech Republic
February 2002

Graphics Hardware Trends

- Faster polygon rasterisation
- Higher bandwidth
- Larger amounts of memory
- Pipeline Parallelisation & Graphics Clustering

Objective: Hardware-based algorithms for CSG rendering.
Context: Dynamic and interactive applications.

Overview

- Graphics Hardware Trends
- Z-Buffer
- Z-Buffer Intersection
- SCS CSG Algorithm
- Performance
- Conclusion & Future Work
- Live Demo
- MPEG Movies

Z-Buffer

- Hardware-based solution to visible-surface problem.
- Implied volumetric union.
- Store z for each pixel in graphics hardware.
- Configurable z-test as part of pipeline.

1. Initialise the z-buffer to \(Z_{\text{far}}\)
2. Only draw pixels closer than z-buffer.
3. Update z-buffer when z-test passes.
**Z-Buffer Intersection**

Algorithm for rendering the intersection of convex objects.

1. Initialise the z-buffer to Znear
2. Draw front-facing surfaces with z-greater test
3. Count back-facing surfaces behind z-buffer
4. Reset pixels where stencil \( l = n \)
5. Draw front-facing surfaces with z-equal test

**SCS-Intersection Illustration**

![Image of intersection illustration]

**SCS-Intersection Results**

Algorithm for rendering CSG trees of convex objects.

1. Convert CSG tree to sum-of-products
2. For each product:
   1. Resolve intersections in z-buffer
   2. Resolve subtractions in z-buffer
3. Merge z-buffer with accumulated result
4. Draw visible surfaces with z-equal test

**SCS CSG Algorithm**

![Image of CSG tree and algorithm steps]
**Future Work**
- Improve performance for convex subtraction.
- Real-world applications: 5-axis NC verification.
- Models for deeper frame buffers in OpenGL 2.0

**Performance**
- Convex intersection resolved in $O(n)$ time.
- Convex subtraction still requires $O(n^2)$ time.
- Speedup depends on relative number of intersections and subtractions.
- Z-buffer copying still a major bottleneck for multiple CSG products.

**Conclusion**
- Convex intersection rendered in linear time on standard OpenGL graphics hardware.
- SCS-Intersect utilised by SCS algorithm for general CSG rendering.
- Interactive frame rates on consumer-grade graphics hardware.