An Improved Z-Buffer CSG Rendering Algorithm

Nigel Stewart, Geoff Leach and Sabu John
RMIT University, Australia

1998 EG/SIGGRAPH Workshop on Graphics Hardware

Goldfeather Methodology
Constructive Solid Geometry (CSG)
Z-Buffer Surface Parity
Tree Normalisation
Z-Buffer Surface Clipping
Optimisation - Geometric Pruning, Caching

Solid Modelling
Analytic Surface Intersection
Boundary Representation (B-REP)
Complex Mathematics
Complex Algorithms
Robustness issues

Image Space
Simple Implementation
Interactivity, Flexibility
Slow

Constructive Solid Geometry (CSG)
Volumetric Objects at Leaf Nodes
Volumetric Operations at Parent Nodes
Boolean operations - Union, Intersection, Difference
Comparative Analysis

Goldfeather: \( n(an+b) \)
Clip layers: \( k(2an+b) \)

- \( n \) is number of primitives
- \( k \) is depth complexity
- \( a \) is cost of rastenisation
- \( b \) is cost of z-buffer copy
- since \( b \) dominates performance, speedup factor of \( n/k \)

Depth Complexity

- \( k(n) \) dependent on domain
- \( k(n) \) observed for an NC verification application
- \( k(n) \) observed in rare (degenerate?) situations
- Topic of research in own right

Goldfeather Optimisations

- Geometric Pruning
- Collapse trivial cases such as A.A
- Use bounding boxes to avoid clipping
- Caching
- Pre-compute invariant subtrees via b-rep

A New Optimisation

- Take advantage of depth complexity
- Clip convex layers rather than primitives
- Merge z-buffer
- Clip surface
- Draw kth surface
Hardware Implementation Issues

Advantage of deeper frame buffers
Advantage of faster z-buffer copying
Further algorithmic improvements

Conclusion

Modified Goldfeather algorithm
O(\(\text{n}^2\)) performance, rather than O(\(\text{n}^3\))
Performs better in many useful cases