Fast Backface Culling Using Normal Masks

Hansong Zhang

Kenneth E. Hoff III

Department of Computer Science

University of North Carolina at Chapel Hill

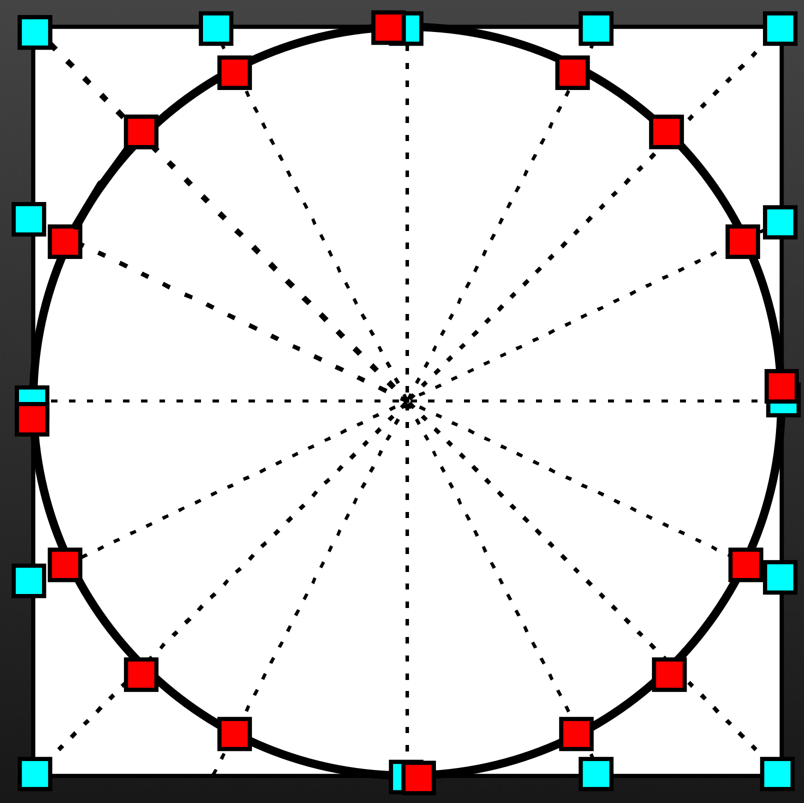
Overview

- Preprocessing: the normal of each polygon is encoded in *two bytes*: **`(a, b)`**
- For each frame, a byte array **`BackfaceMask`** is computed in constant time.
- For each polygon, the backface test is *a single logical operation*: **`BackfaceMask[a] AND b`**

Normal Masks

- Subdivision of the Normal Space - Normal Clusters

Project a uniform subdivision of the faces of a bounding cube on to the unit sphere



Normal Masks

- Representing Normal Clusters
  - Each cluster has a bit in a *normal mask*
  - The normal mask (conceptually) has as many bits as the number of normal clusters
- Encoding the Normals
  - A normal is approximated by a unique cluster

Normal Masks

– A normal can be represented by a normal mask with only one bit on

– A normal mask with only one bit on can be stored in two bytes (a, b)

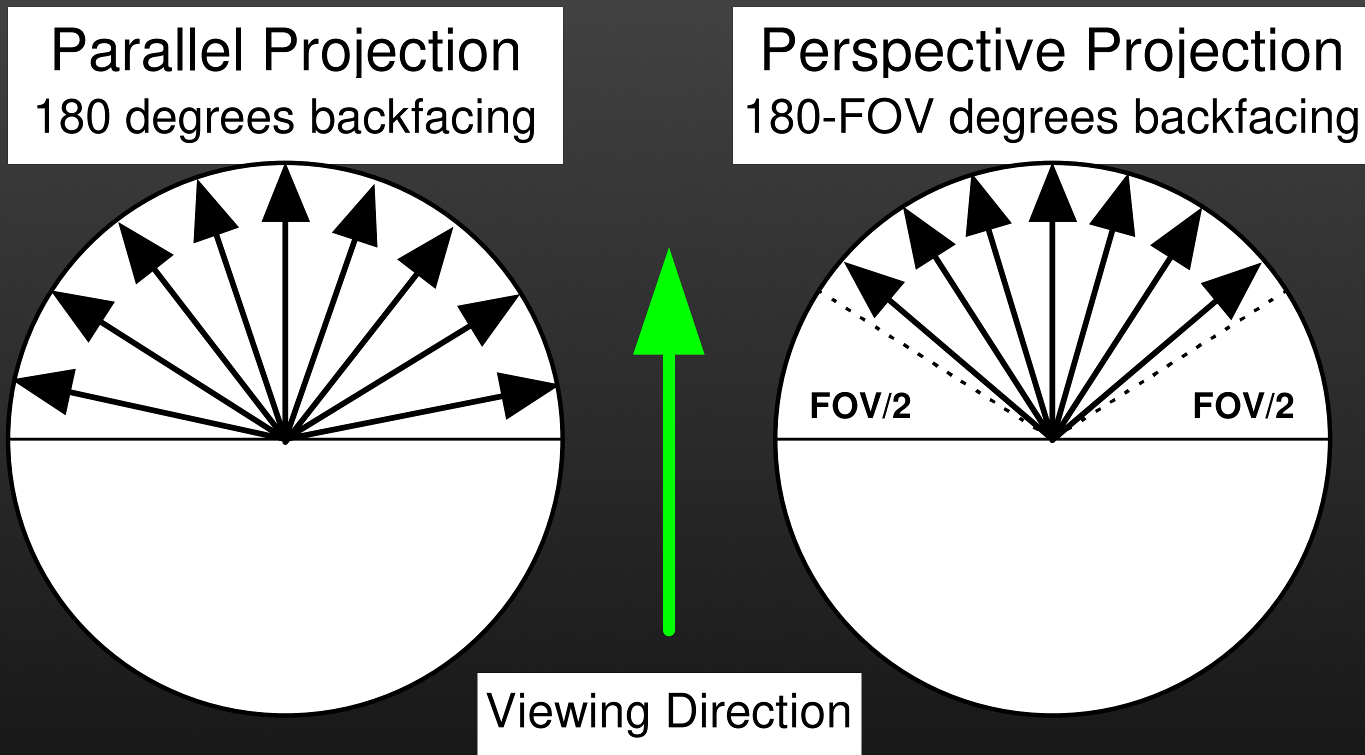
  » a = the byte offset of the non-zero byte

  » b = the non-zero byte

– So each polygon gets two bytes as its encoded, approximate normal

Backface Culling with Normal Masks

- Conservative backface culling



Backface Culling with Normal Masks

- Detecting backfacing normal clusters:
  - A normal cluster is backfacing if all normals in it are backfacing
  - In practice, test only the 4 corner normals
- Recording backfacing normal clusters:
  - Use a **`BackfaceMask`** in which the bits for the backfacing clusters are 1

`...10010111110000101...`

Backface Culling with Normal Masks

- For a polygon with normal code **`(a, b)`**,

<u>**`BackfaceMask[a] AND b`**</u>

$\neq 0$

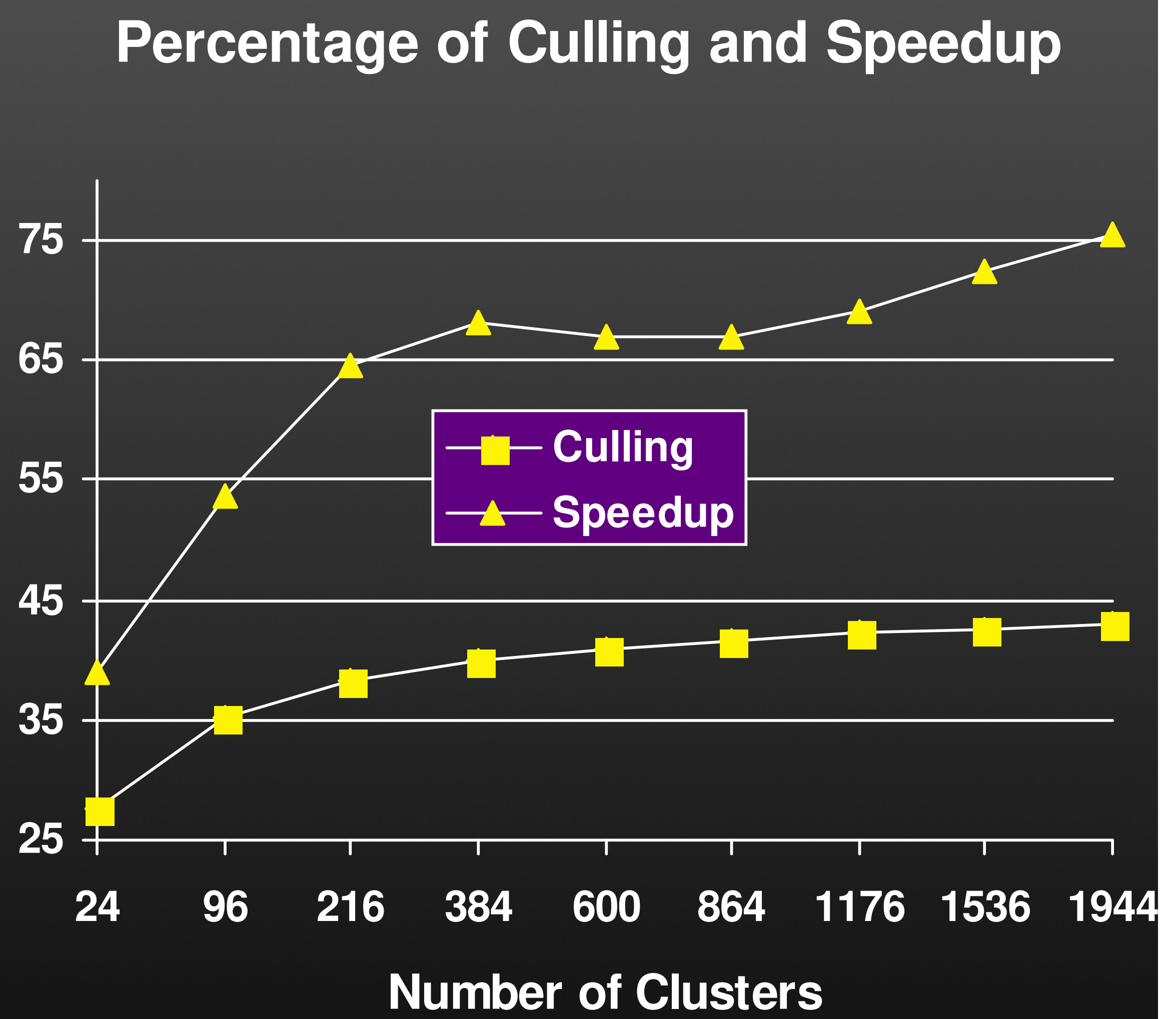
Normal is in a back-facing cluster $\Rightarrow$ *polygon is backfacing*.

$= 0$

Polygon considered front-facing.

Implementation

- Speedup
- Number of Clusters
- Field of View



Related Work

- Normal clusters
  - L.A. Shirman and S.S. Abi-Ezzi, The cone of normals technique for fast processing of curved patches
- Hierarchical Back-face Culling
  - S. Kumar, D. Manocha, B. Garrett, and M. Lin. Hierarchical Back-face Culling.

Conclusion

- Major feature: simple and fast
  - Space: 2 extra bytes per polygon
  - Time: one logical AND per polygon
- Cheap even when applied per-polygon: does not require a special organization of the model
  - Handles unstructured models
  - Easy to integrate into existing applications

Acknowledgements

Sarah H. Hoff

ARPA Contract DABT63-93-C-0048